User's Manual

16

The revision list can be viewed directly by clicking the title page.
The revision list summarizes the locations of revisions and additions.
Details should always be checked by referring to the relevant text.

# H8/36109 Group

User's Manual: Hardware

Renesas 16-Bit Single-Chip Microcomputer
H8 Family / H8/300H Tiny Series

H8/36109F HD64F36109
HD64F36109G

Renesas Electronics
www.renesas.com

Rev.2.00 Sep 2011

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

# General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions may occur due to the false recognition of the pin state as an input signal. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

# Configuration of This Manual

This manual comprises the following items:

1. General Precautions in the Handling of MPU/MCU Products
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

   The configuration of the functional description of each module differs according to the module. However, the generic style includes the following items:

   i) Feature
   ii) Input/Output Pin
   iii) Register Description
   iv) Operation
   v) Usage Note

When designing an application system that includes this LSI, take notes into account. Each section includes notes in relation to the descriptions given, and usage notes are given, as required, as the final part of each section.

7. List of Registers
8. Electrical Characteristics
9. Appendix
10. Main Revisions for this Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier versions. This does not include all of the revised contents. For details, see the actual locations in this manual.

11. Index

# Preface

The H8/36109 Group are single-chip microcomputers made up of the high-speed H8/300H CPU employing Renesas Electronics original architecture as their cores, and the peripheral functions required to configure a system. The H8/300H CPU has an instruction set that is compatible with the H8/300 CPU.

Target Users: This manual was written for users who will be using the H8/36109 Group in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8/36109 Group to the target users.
Refer to the H8/300H Series Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions, and electrical characteristics.
- In order to understand the details of the CPU's functions
  Read the H8/300H Series Software Manual.
- In order to understand the details of a register when its name is known
  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 22, List of Registers.

| | | |
|---|---|---|
| Example: | Register name: | The following notation is used for cases when the same or a similar function, e.g. serial communication interface, is implemented on more than one channel:<br>XXX_N (XXX is the register name and N is the channel number) |
| | Bit order: | The MSB is on the left and the LSB is on the right. |
| | Number notation: | Binary is B'xxxx, hexadecimal is H'xxxx, and decimal is xxxx. |
| | Signal notation: | An overbar is added to a low-active signal: $\overline{\text{xxxx}}$ |

Notes:

When using an on-chip emulator (E7, E8, or E8a) for H8/36109 program development and debugging, the following restrictions must be noted.

1. The $\overline{\text{NMI}}$ pin is reserved for the E7, E8, or E8a, and cannot be used.
2. Pins P85, P86, and P87 cannot be used. In order to use these pins, additional hardware must be provided on the user board.
3. Area H'01F000 to H'01FFFF is used by the E7, E8, or E8a, and is not available to the user.
4. Area H'F780 to H'FB7F must on no account be accessed.
5. When the E7, E8, or E8a is used, address breaks can be set as either available to the user or for use by the E7, E8, or E8a. If address breaks are set as being used by the E7, E8, or E8a, the address break control registers must not be accessed.
6. When the E7, E8, or E8a is used, $\overline{\text{NMI}}$ is an input/output pin (open-drain in output mode), P85 and P87 are input pins, and P86 is an output pin.
7. Use channel 1 of the SCI3 (P21/RXD, P22/TXD) in on-board programming mode by boot mode.

Related Manuals: The latest versions of all related manuals are available from our web site. Please ensure you have the latest versions of all documents you require. http://www.renesas.com/

H8/36109 Group manuals:

| Document Title | Document No. |
|---|---|
| H8/36109 Group Hardware Manual | This manual |
| H8/300H Series Software Manual | REJ09B0213 |

User's manuals for development tools:

| Document Title | Document No. |
|---|---|
| H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual | REJ10J2039 |
| H8S, H8/300 Series Simulator/Debugger User's Manual | REJ10B0211 |
| High-Performance Embedded Workshop, User's Manual | R20UT0372EJ0100 |

Application notes:

| Document Title | Document No. |
|---|---|
| H8S, H8/300 Series C/C++ Compiler Package Application Note | REJ05B0464 |
| Single Power Supply F-ZTAT™ On-Board Programming | REJ05B0520 |

All trademarks and registered trademarks are the property of their respective owners.

# Contents


Section 1 Overview ..... 1
1.1 Features ..... 1
1.2 Internal Block Diagram ..... 3
1.3 Pin Assignment ..... 4
1.4 Pin Functions ..... 6

Section 2 CPU ..... 11
2.1 Address Space and Memory Map ..... 12
2.2 Register Configuration ..... 13
2.2.1 General Registers ..... 14
2.2.2 Program Counter (PC) ..... 15
2.2.3 Condition-Code Register (CCR) ..... 15
2.3 Data Formats ..... 17
2.3.1 General Register Data Formats ..... 17
2.3.2 Memory Data Formats ..... 19
2.4 Instruction Set ..... 20
2.4.1 List of Instructions Classified by Function ..... 20
2.4.2 Basic Instruction Formats ..... 30
2.5 Addressing Modes and Effective Address Calculation ..... 31
2.5.1 Addressing Modes ..... 31
2.5.2 Effective Address Calculation ..... 34
2.6 Basic Bus Cycle ..... 36
2.6.1 Access to On-Chip Memory (RAM, ROM) ..... 36
2.6.2 On-Chip Peripheral Modules ..... 37
2.7 CPU States ..... 38
2.8 Usage Notes ..... 39
2.8.1 Notes on Data Access to Empty Areas ..... 39
2.8.2 EEPMOV Instruction ..... 39
2.8.3 Bit Manipulation Instruction ..... 39

Section 3 Exception Handling ..... 45
3.1 Exception Sources and Vector Address ..... 45
3.2 Register Descriptions ..... 48
3.2.1 Interrupt Edge Select Register 1 (IEGR1) ..... 48
3.2.2 Interrupt Edge Select Register 2 (IEGR2) ..... 49
3.2.3 Interrupt Enable Register 1 (IENR1) ..... 50

3.2.4 Interrupt Enable Register 2 (IENR2) ..... 51
3.2.5 Interrupt Flag Register 1 (IRR1) ..... 51
3.2.6 Interrupt Flag Register 2 (IRR2) ..... 53
3.2.7 Wakeup Interrupt Flag Register (IWPR) ..... 53
3.2.8 Interrupt Control Registers A to D (ICRA to ICRD) ..... 55
3.3 Reset Exception Handling ..... 56
3.4 Interrupt Exception Handling ..... 56
3.4.1 External Interrupts ..... 56
3.4.2 Internal Interrupts ..... 58
3.4.3 Interrupt Handling Sequence ..... 58
3.4.4 Interrupt Response Time ..... 62
3.5 Usage Notes ..... 63
3.5.1 Interrupts after Reset ..... 63
3.5.2 Notes on Stack Area Use ..... 63
3.5.3 Notes on Rewriting Port Mode Registers ..... 63

Section 4 Address Break ..... 65
4.1 Register Descriptions ..... 66
4.1.1 Address Break Control Register (ABRKCR) ..... 66
4.1.2 Address Break Status Register (ABRKSR) ..... 68
4.1.3 Break Address Registers E, H, L (BARE, BARH, BARL) ..... 68
4.1.4 Break Data Registers H, L (BDRH, BDRL) ..... 68
4.2 Operation ..... 69

Section 5 Clock Pulse Generators ..... 71
5.1 Features ..... 72
5.2 Register Descriptions ..... 72
5.2.1 RC Control Register (RCCR) ..... 73
5.2.2 RC Trimming Data Protect Register (RCTRMDPR) ..... 74
5.2.3 RC Trimming Data Register (RCTRMDR) ..... 75
5.2.4 Clock Control/Status Register (CKCSR) ..... 76
5.3 System Clock Oscillator ..... 77
5.3.1 State Transition of System Clock ..... 77
5.3.2 Clock Control Operation ..... 78
5.3.3 Clock Change Timing ..... 80
5.4 Trimming of On-Chip Oscillator Frequency ..... 82
5.5 External Oscillator ..... 84
5.5.1 Connecting Crystal Resonator ..... 84
5.5.2 Connecting Ceramic Resonator ..... 85
5.5.3 External Clock Input Method ..... 85

5.6 Subclock Generator ... 86
5.6.1 Connecting 32.768-kHz Crystal Resonator ... 86
5.6.2 Pin Connection when not Using Subclock ... 87
5.7 Prescaler ... 87
5.7.1 Prescaler S ... 87
5.7.2 Prescaler W ... 87
5.8 Usage Notes ... 88
5.8.1 Note on Resonators ... 88
5.8.2 Notes on Board Design ... 88

Section 6 Power-Down Modes ... 89
6.1 Register Descriptions ... 90
6.1.1 System Control Register 1 (SYSCR1) ... 90
6.1.2 System Control Register 2 (SYSCR2) ... 93
6.1.3 System Control Register 3 (SYSCR3) ... 94
6.1.4 Module Standby Control Register 1 (MSTCR1) ... 94
6.1.5 Module Standby Control Register 2 (MSTCR2) ... 95
6.1.6 Module Standby Control Register 4 (MSTCR4) ... 96
6.2 Mode Transitions and States of LSI ... 97
6.2.1 Sleep Mode ... 100
6.2.2 Standby Mode ... 100
6.2.3 Subsleep Mode ... 101
6.2.4 Subactive Mode ... 101
6.3 Operating Frequency in Active Mode ... 101
6.4 Direct Transition ... 102
6.4.1 Direct Transition from Active Mode to Subactive Mode ... 102
6.4.2 Direct Transition from Subactive Mode to Active Mode ... 103
6.5 Module Standby Function ... 103

Section 7 ROM ... 105
7.1 Block Configuration ... 106
7.2 Register Descriptions ... 107
7.2.1 Flash Memory Control Register 1 (FLMCR1) ... 107
7.2.2 Flash Memory Control Register 2 (FLMCR2) ... 108
7.2.3 Erase Block Register 1 (EBR1) ... 109
7.2.4 Flash Memory Power Control Register (FLPWCR) ... 110
7.2.5 Flash Memory Enable Register (FENR) ... 110
7.3 On-Board Programming Modes ... 111
7.3.1 Boot Mode ... 111
7.3.2 Programming/Erasing in User Programming Mode ... 114

7.4 Flash Memory Programming/Erasing ..... 115
7.4.1 Programming/Program-Verify ..... 115
7.4.2 Erasure/Erase-Verify ..... 118
7.4.3 Interrupt Handling when Programming/Erasing Flash Memory ..... 118
7.5 Programming/Erasing Protection ..... 120
7.5.1 Hardware Protection ..... 120
7.5.2 Software Protection ..... 120
7.5.3 Error Protection ..... 120
7.6 Programmer Mode ..... 121
7.7 Power-Down States for Flash Memory ..... 121

Section 8 RAM ..... 123

Section 9 I/O Ports ..... 125
9.1 Port 1 ..... 125
9.1.1 Port Mode Register 1 (PMR1) ..... 126
9.1.2 Port Control Register 1 (PCR1) ..... 127
9.1.3 Port Data Register 1 (PDR1) ..... 127
9.1.4 Port Pull-Up Control Register 1 (PUCR1) ..... 128
9.1.5 Pin Functions ..... 128
9.2 Port 2 ..... 131
9.2.1 Port Control Register 2 (PCR2) ..... 131
9.2.2 Port Data Register 2 (PDR2) ..... 132
9.2.3 Port Mode Register 3 (PMR3) ..... 132
9.2.4 Pin Functions ..... 133
9.3 Port 3 ..... 135
9.3.1 Port Control Register 3 (PCR3) ..... 135
9.3.2 Port Data Register 3 (PDR3) ..... 136
9.3.3 Pin Functions ..... 136
9.4 Port 5 ..... 138
9.4.1 Port Mode Register 5 (PMR5) ..... 139
9.4.2 Port Control Register 5 (PCR5) ..... 140
9.4.3 Port Data Register 5 (PDR5) ..... 140
9.4.4 Port Pull-Up Control Register 5 (PUCR5) ..... 141
9.4.5 Pin Functions ..... 141
9.5 Port 7 ..... 144
9.5.1 Port Control Register 7 (PCR7) ..... 144
9.5.2 Port Data Register 7 (PDR7) ..... 145
9.5.3 Pin Functions ..... 145
9.6 Port 8 ..... 147

9.6.1 Port Control Register 8 (PCR8) ..... 147
9.6.2 Port Data Register 8 (PDR8) ..... 148
9.6.3 Pin Functions ..... 148
9.7 Port C ..... 149
9.7.1 Port Control Register C (PCRC) ..... 149
9.7.2 Port Data Register C (PDRC) ..... 150
9.7.3 Pin Functions ..... 150
9.8 Port D ..... 151
9.8.1 Port Control Register D (PCRD) ..... 152
9.8.2 Port Data Register D (PDRD) ..... 152
9.8.3 Pin Functions ..... 153
9.9 Port E ..... 160
9.9.1 Port Control Register E (PCRE) ..... 161
9.9.2 Port Data Register E (PDRE) ..... 161
9.9.3 Pin Functions ..... 162
9.10 Port F ..... 170
9.10.1 Port Data Register F (PDRF) ..... 170
9.10.2 Port Mode Register F (PMRF) ..... 171
9.10.3 Pin Functions ..... 171
9.11 Port G ..... 173
9.11.1 Port Control Register G (PCRG) ..... 174
9.11.2 Port Data Register G (PDRG) ..... 174
9.11.3 Port Mode Register G (PMRG) ..... 175
9.11.4 Pin Functions ..... 176
9.12 Port H ..... 179
9.12.1 Port Control Register H (PCRH) ..... 179
9.12.2 Port Data Register H (PDRH) ..... 180
9.12.3 Pin Functions ..... 180
9.13 Port J ..... 184
9.13.1 Port Control Register J (PCRJ) ..... 184
9.13.2 Port Data Register J (PDRJ) ..... 185
9.13.3 Pin Functions ..... 185

Section 10 Realtime Clock (RTC) ..... 187
10.1 Features ..... 187
10.2 Input/Output Pin ..... 188
10.3 Register Descriptions ..... 188
10.3.1 Second Data Register/Free Running Counter Data Register (RSECDR) ..... 188
10.3.2 Minute Data Register (RMINDR) ..... 189
10.3.3 Hour Data Register (RHRDR) ..... 190

10.3.4 Day-of-Week Data Register (RWKDR) ..... 191
10.3.5 RTC Control Register 1 (RTCCR1) ..... 192
10.3.6 RTC Control Register 2 (RTCCR2) ..... 193
10.3.7 Clock Source Select Register (RTCCSR) ..... 194
10.4 Operation ..... 195
10.4.1 Initial Settings of Registers after Power-On ..... 195
10.4.2 Initial Setting Procedure ..... 195
10.4.3 Data Reading Procedure ..... 196
10.5 Interrupt Sources ..... 197

Section 11 Timer B1 ..... 199
11.1 Features ..... 199
11.2 Input/Output Pin ..... 199
11.3 Register Descriptions ..... 200
11.3.1 Timer Mode Register B1 (TMB1) ..... 200
11.3.2 Timer Counter B1 (TCB1) ..... 201
11.3.3 Timer Load Register B1 (TLB1) ..... 201
11.4 Operation ..... 201
11.4.1 Interval Timer Operation ..... 201
11.4.2 Auto-Reload Timer Operation ..... 202
11.4.3 Event Counter Operation ..... 202
11.5 Timer B1 Operating Modes ..... 202

Section 12 Timer V ..... 203
12.1 Features ..... 203
12.2 Input/Output Pins ..... 204
12.3 Register Descriptions ..... 205
12.3.1 Timer Counter V (TCNTV) ..... 205
12.3.2 Time Constant Registers A, B (TCORA, TCORB) ..... 205
12.3.3 Timer Control Register V0 (TCRV0) ..... 206
12.3.4 Timer Control/Status Register V (TCSRV) ..... 207
12.3.5 Timer Control Register V1 (TCRV1) ..... 209
12.4 Operation ..... 210
12.4.1 Timer V Operation ..... 210
12.5 Timer V Application Examples ..... 213
12.5.1 Pulse Output with Arbitrary Duty Cycle ..... 213
12.5.2 Pulse Output with Arbitrary Pulse Width and Delay from TRGV Input ..... 214
12.6 Usage Notes ..... 215

Section 13 Timer RC ..... 217
13.1 Features ..... 217
13.2 Input/Output Pins ..... 220
13.3 Register Descriptions ..... 221
13.3.1 Timer RC Mode Register (TRCMR) ..... 222
13.3.2 Timer RC Control Register 1 (TRCCR1) ..... 224
13.3.3 Timer RC Control Register 2 (TRCCR2) ..... 225
13.3.4 Timer RC Interrupt Enable Register (TRCIER) ..... 226
13.3.5 Timer RC Status Register (TRCSR) ..... 227
13.3.6 Timer RC I/O Control Register 0 (TRCIOR0) ..... 229
13.3.7 Timer RC I/O Control Register 1 (TRCIOR1) ..... 231
13.3.8 Timer RC Output Enable Register (TRCOER) ..... 233
13.3.9 Timer RC Digital Filtering Function Select Register (TRCDF) ..... 234
13.3.10 Timer RC Counter (TRCCNT) ..... 235
13.3.11 General Registers A, B, C, and D (GRA, GRB, GRC, and GRD) ..... 235
13.4 Operation ..... 236
13.4.1 Timer Mode Operation ..... 238
13.4.2 PWM Mode Operation ..... 243
13.4.3 PWM2 Mode Operation ..... 247
13.4.4 Digital Filtering Function for Input Capture Inputs ..... 253
13.5 Operation Timing ..... 254
13.5.1 TRCCNT Counting Timing ..... 254
13.5.2 Output Compare Output Timing ..... 255
13.5.3 Input Capture Timing ..... 256
13.5.4 Timing of Counter Clearing by Compare Match ..... 256
13.5.5 Buffer Operation Timing ..... 257
13.5.6 Timing of IMFA to IMFD Flag Setting at Compare Match ..... 258
13.5.7 Timing of IMFA to IMFD Setting at Input Capture ..... 259
13.5.8 Timing of Status Flag Clearing ..... 260
13.6 Usage Notes ..... 261

Section 14 Timer RD ..... 265
14.1 Features ..... 265
14.2 Input/Output Pins ..... 272
14.3 Register Descriptions ..... 273
14.3.1 Timer RD Start Register (TRDSTR) ..... 274
14.3.2 Timer RD Mode Register (TRDMDR) ..... 278
14.3.3 Timer RD PWM Mode Register (TRDPMR) ..... 279
14.3.4 Timer RD Function Control Register (TRDFCR) ..... 280

14.3.5 Timer RD Output Master Enable Register 1 (TRDOER1) ..... 282
14.3.6 Timer RD Output Master Enable Register 2 (TRDOER2) ..... 284
14.3.7 Timer RD Output Control Register (TRDOCR) ..... 284
14.3.8 Timer RD Counter (TRDCNT) ..... 286
14.3.9 General Registers A, B, C, and D (GRA, GRB, GRC, and GRD) ..... 286
14.3.10 Timer RD Control Register (TRDCR) ..... 287
14.3.11 Timer RD I/O Control Registers (TRDIORA and TRDIORC) ..... 289
14.3.12 Timer RD Status Register (TRDSR) ..... 293
14.3.13 Timer RD Interrupt Enable Register (TRDIER) ..... 296
14.3.14 PWM Mode Output Level Control Register (POCR) ..... 297
14.3.15 Timer RD Digital Filtering Function Select Register (TRDDF) ..... 298
14.3.16 Interface with CPU ..... 299
14.4 Operation ..... 300
14.4.1 Counter Operation ..... 306
14.4.2 Waveform Output by Compare Match ..... 309
14.4.3 Input Capture Function ..... 312
14.4.4 Synchronous Operation ..... 315
14.4.5 PWM Mode ..... 316
14.4.6 Reset Synchronous PWM Mode ..... 322
14.4.7 Complementary PWM Mode ..... 326
14.4.8 PWM3 Mode Operation ..... 332
14.4.9 Buffer Operation ..... 336
14.4.10 Timer RD Output Timing ..... 344
14.4.11 Digital Filtering Function for Input Capture Inputs ..... 346
14.4.12 Function of Changing Output Pins for GR ..... 347
14.5 Interrupt Sources ..... 349
14.5.1 Status Flag Set Timing ..... 349
14.5.2 Status Flag Clearing Timing ..... 351
14.6 Usage Notes ..... 351

Section 15 Watchdog Timer ..... 359
15.1 Features ..... 359
15.2 Register Descriptions ..... 360
15.2.1 Timer Control/Status Register WD (TCSRWD) ..... 360
15.2.2 Timer Counter WD (TCWD) ..... 362
15.2.3 Timer Mode Register WD (TMWD) ..... 362
15.3 Operation ..... 363

Section 16 14-Bit PWM ..... 365
16.1 Features ..... 365

16.2 Input/Output Pin ..... 365
16.3 Register Descriptions ..... 366
16.3.1 PWM Control Register (PWCR) ..... 366
16.3.2 PWM Data Registers U, L (PWDRU, PWDRL) ..... 367
16.4 Operation ..... 367

Section 17 Serial Communication Interface 3 (SCI3) ..... 369
17.1 Features ..... 369
17.2 Input/Output Pins ..... 372
17.3 Register Descriptions ..... 373
17.3.1 Receive Shift Register (RSR) ..... 373
17.3.2 Receive Data Register (RDR) ..... 373
17.3.3 Transmit Shift Register (TSR) ..... 373
17.3.4 Transmit Data Register (TDR) ..... 374
17.3.5 Serial Mode Register (SMR) ..... 374
17.3.6 Serial Control Register 3 (SCR3) ..... 375
17.3.7 Serial Status Register (SSR) ..... 377
17.3.8 Bit Rate Register (BRR) ..... 379
17.4 Operation in Asynchronous Mode ..... 383
17.4.1 Clock ..... 383
17.4.2 SCI3 Initialization ..... 384
17.4.3 Data Transmission ..... 385
17.4.4 Serial Data Reception ..... 387
17.5 Operation in Clock Synchronous Mode ..... 391
17.5.1 Clock ..... 391
17.5.2 SCI3 Initialization ..... 391
17.5.3 Serial Data Transmission ..... 392
17.5.4 Serial Data Reception (Clock Synchronous Mode) ..... 394
17.5.5 Simultaneous Serial Data Transmission and Reception ..... 396
17.6 Multiprocessor Communication Function ..... 397
17.6.1 Multiprocessor Serial Data Transmission ..... 399
17.6.2 Multiprocessor Serial Data Reception ..... 400
17.7 Interrupt Requests ..... 403
17.8 Usage Notes ..... 404
17.8.1 Break Detection and Processing ..... 404
17.8.2 Mark State and Break Sending ..... 404
17.8.3 Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only) ..... 404
17.8.4 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode ..... 405

Section 18 I²C Bus Interface 2 (IIC2) ..... 407
18.1 Features ..... 407
18.2 Input/Output Pins ..... 409
18.3 Register Descriptions ..... 410
18.3.1 I²C Bus Control Register 1 (ICCR1) ..... 410
18.3.2 I²C Bus Control Register 2 (ICCR2) ..... 413
18.3.3 I²C Bus Mode Register (ICMR) ..... 415
18.3.4 I²C Bus Interrupt Enable Register (ICIER) ..... 417
18.3.5 I²C Bus Status Register (ICSR) ..... 419
18.3.6 Slave Address Register (SAR) ..... 421
18.3.7 I²C Bus Transmit Data Register (ICDRT) ..... 422
18.3.8 I²C Bus Receive Data Register (ICDRR) ..... 422
18.3.9 I²C Bus Shift Register (ICDRS) ..... 422
18.4 Operation ..... 423
18.4.1 I²C Bus Format ..... 423
18.4.2 Master Transmit Operation ..... 424
18.4.3 Master Receive Operation ..... 426
18.4.4 Slave Transmit Operation ..... 428
18.4.5 Slave Receive Operation ..... 431
18.4.6 Clocked Synchronous Serial Format ..... 432
18.4.7 Noise Canceller ..... 435
18.4.8 Example of Use ..... 435
18.5 Interrupts ..... 440
18.6 Bit Synchronous Circuit ..... 441
18.7 Usage Notes ..... 442
18.7.1 Restriction on Setting of Transfer Rate in Use of Multi-Master ..... 442
18.7.2 Restriction on Use of Bit Manipulation Instructions to Set MST and TRS when Multi-Master is Used ..... 442
18.7.3 Usage Note on Master Receive Mode ..... 442
18.7.4 Note Regarding Master Receive Mode of I²C-Bus Interface Mode ..... 443
18.7.5 Accessing ICE and IICRST During I²C Bus Operation ..... 443
18.7.6 Note on Issuance of Stop Condition in Master Transmit Mode ..... 444

Section 19 A/D Converter ..... 445
19.1 Features ..... 445
19.2 Input/Output Pins ..... 447
19.3 Register Descriptions ..... 448
19.3.1 A/D Data Registers A to D (ADDRA to ADDRD) ..... 448
19.3.2 A/D Control/Status Register (ADCSR) ..... 449

19.3.3 A/D Control Register (ADCR) ..... 451
19.4 Operation ..... 452
19.4.1 Single Mode ..... 452
19.4.2 Scan Mode ..... 452
19.4.3 Input Sampling and A/D Conversion Time ..... 453
19.4.4 External Trigger Input Timing ..... 454
19.5 A/D Conversion Accuracy Definitions ..... 455
19.6 Usage Notes ..... 458
19.6.1 Permissible Signal Source Impedance ..... 458
19.6.2 Influences on Absolute Accuracy ..... 458
19.6.3 Notes on Analog Pins ..... 459

Section 20 Band-Gap Regulator, Power-On Reset (Optional), and Low-Voltage Detection Circuits (Optional) ..... 461
20.1 Features ..... 462
20.2 Register Descriptions ..... 464
20.2.1 Low-Voltage-Detection Control Register (LVDCR) ..... 464
20.2.2 Low-Voltage-Detection Status Register (LVDSR) ..... 466
20.3 Operation ..... 467
20.3.1 Power-On Reset Circuit ..... 467
20.3.2 Low-Voltage Detection Circuit ..... 468

Section 21 Power Supply Circuit ..... 471
21.1 When Using Internal Power Supply Step-Down Circuit ..... 471
21.2 When Not Using Internal Power Supply Step-Down Circuit ..... 472

Section 22 List of Registers ..... 473
22.1 Register Addresses (Address Order) ..... 474
22.2 Register Bits ..... 483
22.3 Register States in Each Operating Mode ..... 491

Section 23 Electrical Characteristics ..... 499
23.1 Absolute Maximum Ratings ..... 499
23.2 Electrical Characteristics ..... 500
23.2.1 Power Supply Voltage and Operating Ranges ..... 500
23.2.2 DC Characteristics ..... 502
23.2.3 AC Characteristics ..... 510
23.2.4 A/D Converter Characteristics ..... 515
23.2.5 Watchdog Timer Characteristics ..... 516
23.2.6 Flash Memory Characteristics ..... 517

23.2.7 Power-Supply-Voltage Detection Circuit Characteristics (Optional) ..... 519
23.2.8 Power-On Reset Circuit Characteristics (Optional) ..... 520
23.3 Operation Timing ..... 520
23.4 Output Load Condition ..... 522

Appendix ..... 523
A. Instruction Set ..... 523
A.1 Instruction List ..... 523
A.2 Operation Code Map ..... 538
A.3 Number of Execution States ..... 541
A.4 Combinations of Instructions and Addressing Modes ..... 552
B. I/O Port Block Diagrams ..... 553
B.1 I/O Port Block Diagrams ..... 553
B.2 Port States in Each Operating Mode ..... 571
C. Product Code Lineup ..... 572
D. Package Dimensions ..... 572

Main Revisions for this Edition ..... 575

Index ..... 579

# Figures

**Section 1 Overview**
Figure 1.1 Internal Block Diagram ..... 3
Figure 1.2 Pin Assignments (FP-100A) ..... 4
Figure 1.3 Pin Assignments (FP-100U) ..... 5

**Section 2 CPU**
Figure 2.1 Memory Map ..... 12
Figure 2.2 CPU Registers ..... 13
Figure 2.3 Usage of General Registers ..... 14
Figure 2.4 Relationship between Stack Pointer and Stack Area ..... 15
Figure 2.5 General Register Data Formats (1) ..... 17
Figure 2.5 General Register Data Formats (2) ..... 18
Figure 2.6 Memory Data Formats ..... 19
Figure 2.7 Instruction Formats ..... 30
Figure 2.8 Branch Address Specification in Memory Indirect Mode ..... 33
Figure 2.9 On-Chip Memory Access Cycle ..... 36
Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access) ..... 37
Figure 2.11 CPU Operation States ..... 38
Figure 2.12 State Transitions ..... 39
Figure 2.13 Example of Timer Configuration with Two Registers Allocated to Same Address ..... 40

**Section 3 Exception Handling**
Figure 3.1 Reset Sequence ..... 57
Figure 3.2 Interrupt Acceptance Flowchart ..... 59
Figure 3.3 Stack Status after Exception Handling ..... 60
Figure 3.4 Interrupt Sequence ..... 61
Figure 3.5 Port Mode Register Setting and Interrupt Request Flag Clearing Procedure ..... 63

**Section 4 Address Break**
Figure 4.1 Block Diagram of Address Break ..... 65
Figure 4.2 Address Break Interrupt Operation Example (1) ..... 69
Figure 4.2 Address Break Interrupt Operation Example (2) ..... 70

**Section 5 Clock Pulse Generators**
Figure 5.1 Block Diagram of Clock Pulse Generators ..... 71
Figure 5.2 State Transition of System Clock ..... 77
Figure 5.3 Flowchart of Clock Switching (From On-Chip Oscillator Clock to External Clock) ..... 78

Figure 5.4 Flowchart of Clock Switching
(From External Clock to On-Chip Oscillator Clock) ..... 79
Figure 5.5 Timing Chart of Switching from On-Chip Oscillator Clock to External Clock ..... 80
Figure 5.6 Timing Chart to Switch from External Clock to On-Chip Oscillator Clock ..... 81
Figure 5.7 Example of Trimming Flow for On-Chip Oscillator Frequency ..... 82
Figure 5.8 Timing Chart of Trimming of On-Chip Oscillator Frequency ..... 83
Figure 5.9 Block Diagram of External Oscillator ..... 84
Figure 5.10 Example of Connection to Crystal Resonator ..... 84
Figure 5.11 Equivalent Circuit of Crystal Resonator ..... 84
Figure 5.12 Example of Connection to Ceramic Resonator ..... 85
Figure 5.13 Example of External Clock Input ..... 85
Figure 5.14 Block Diagram of Subclock Generator ..... 86
Figure 5.15 Typical Connection to 32.768-kHz Crystal Resonator ..... 86
Figure 5.16 Equivalent Circuit of 32.768-kHz Crystal Resonator ..... 86
Figure 5.17 Pin Connection when not Using Subclock ..... 87
Figure 5.18 Example of Incorrect Board Design ..... 88

**Section 6 Power-Down Modes**

Figure 6.1 Mode Transition Diagram ..... 97

**Section 7 ROM**

Figure 7.1 Block Configuration of Flash Memory ..... 106
Figure 7.2 Programming/Erasing Flowchart Example in User Programming Mode ..... 114
Figure 7.3 Programming/Program-Verify Flowchart ..... 116
Figure 7.4 Erasure/Erase-Verify Flowchart ..... 119

**Section 9 I/O Ports**

Figure 9.1 Port 1 Pin Configuration ..... 125
Figure 9.2 Port 2 Pin Configuration ..... 131
Figure 9.3 Port 3 Pin Configuration ..... 135
Figure 9.4 Port 5 Pin Configuration ..... 138
Figure 9.5 Port 7 Pin Configuration ..... 144
Figure 9.6 Port 8 Pin Configuration ..... 147
Figure 9.7 Port C Pin Configuration ..... 149
Figure 9.8 Port D Pin Configuration ..... 151
Figure 9.9 Port E Pin Configuration ..... 160
Figure 9.10 Port F Pin Configuration ..... 170
Figure 9.11 Port G Pin Configuration ..... 173
Figure 9.12 Port H Pin Configuration ..... 179
Figure 9.13 Port J Pin Configuration ..... 184

**Section 10 Realtime Clock (RTC)**

Figure 10.1 Block Diagram of RTC ..... 187
Figure 10.2 Definition of Time Expression ..... 192
Figure 10.3 Initial Setting Procedure ..... 195
Figure 10.4 Example: Reading of Inaccurate Time Data ..... 196

**Section 11 Timer B1**

Figure 11.1 Block Diagram of Timer B1 ..... 199

**Section 12 Timer V**

Figure 12.1 Block Diagram of Timer V ..... 204
Figure 12.2 Increment Timing with Internal Clock ..... 211
Figure 12.3 Increment Timing with External Clock ..... 211
Figure 12.4 OVF Set Timing ..... 211
Figure 12.5 CMFA and CMFB Set Timing ..... 212
Figure 12.6 TMOV Output Timing ..... 212
Figure 12.7 Clear Timing by Compare Match ..... 212
Figure 12.8 Clear Timing by TMRIV Input ..... 212
Figure 12.9 Pulse Output Example ..... 213
Figure 12.10 Example of Pulse Output Synchronized to TRGV Input ..... 214
Figure 12.11 Contention between TCNTV Write and Clear ..... 215
Figure 12.12 Contention between TCORA Write and Compare Match ..... 216
Figure 12.13 Internal Clock Switching and TCNTV Operation ..... 216

**Section 13 Timer RC**

Figure 13.1 Timer RC Block Diagram ..... 219
Figure 13.2 Free-Running Counter Operation ..... 238
Figure 13.3 Periodic Counter Operation ..... 239
Figure 13.4 0 and 1 Output Example (TOA = 0, TOB = 1) ..... 239
Figure 13.5 Toggle Output Example (TOA = 0, TOB = 1) ..... 240
Figure 13.6 Toggle Output Example (TOA = 0, TOB = 1) ..... 240
Figure 13.7 Input Capture Operating Example ..... 241
Figure 13.8 Buffer Operation Example (Input Capture) ..... 242
Figure 13.9 PWM Mode Example (1) ..... 243
Figure 13.10 PWM Mode Example (2) ..... 244
Figure 13.11 Buffer Operation Example (Output Compare) ..... 244
Figure 13.12 PWM Mode Example (TOB, TOC, and TOD = 0: Initial Output Set to 0) ..... 245
Figure 13.13 PWM Mode Example (TOB, TOC, and TOD = 1: Initial Output Set to 1) ..... 246
Figure 13.14 Block Diagram in PWM2 Mode ..... 248
Figure 13.15 GRD and GRB Buffer Operating Timing in PWM2 Mode (1) ..... 248
Figure 13.16 GRD and GRB Buffer Operating Timing in PWM2 Mode (2) ..... 249

Figure 13.17 Example (1) of TRGC Synchronous Operation in PWM2 Mode ..... 250
Figure 13.18 Example (2) of TRGC Synchronous Operation in PWM2 Mode ..... 250
Figure 13.19 Example of Stopping Operation of the Counter in PWM2 Mode ..... 251
Figure 13.20 Example (1) of Output Operation of One-Shot Pulse Waveform in PWM2 Mode ..... 251
Figure 13.21 Example (2) of Output Operation of One-Shot Pulse Waveform in PWM2 Mode ..... 252
Figure 13.22 Block Diagram of Digital Filter ..... 253
Figure 13.23 Count Timing for Internal Clock Source ..... 254
Figure 13.24 Count Timing for External Clock Source ..... 254
Figure 13.25 Output Compare Output Timing ..... 255
Figure 13.26 Input Capture Input Signal Timing ..... 256
Figure 13.27 Timing of Counter Clearing by Compare Match ..... 256
Figure 13.28 Buffer Operation Timing (Compare Match) ..... 257
Figure 13.29 Buffer Operation Timing (Input Capture) ..... 257
Figure 13.30 Timing of IMFA to IMFD Flag Setting at Compare Match ..... 258
Figure 13.31 Timing of IMFA to IMFD Flag Setting at Input Capture ..... 259
Figure 13.32 Timing of Status Flag Clearing by CPU ..... 260
Figure 13.33 Contention between TRCCNT Write and Clear ..... 261
Figure 13.34 Internal Clock Switching and TRCCNT Operation ..... 262
Figure 13.35 When Compare Match and Bit Manipulation Instruction to TRCCR1 Occur at the Same Timing ..... 263

**Section 14 Timer RD**

Figure 14.1 Timer RD Block Diagram ..... 269
Figure 14.2 Timer RD (Channel 0) Block Diagram ..... 270
Figure 14.3 Timer RD (Channel 1) Block Diagram ..... 271
Figure 14.4 Example (1) of Stopping Operation of the Counter (in PWM3 Mode) ..... 276
Figure 14.5 Example (2) of Stopping Operation of the Counter (in PWM3 Mode) ..... 276
Figure 14.6 Example of Starting and Stopping Operations of Counters (in PWM3 Mode) ..... 277
Figure 14.7 Example of Outputs in Reset Synchronous PWM Mode and Complementary PWM Mode ..... 282
Figure 14.8 Accessing Operation of 16-Bit Register (between CPU and TRDCNT (16 bits)) ..... 299
Figure 14.9 Accessing Operation of 8-Bit Register (between CPU and TRDSTR (8 bits)) ..... 299
Figure 14.10 Example of Counter Operation Setting Procedure ..... 306
Figure 14.11 Free-Running Counter Operation ..... 307
Figure 14.12 Periodic Counter Operation ..... 308
Figure 14.13 Count Timing at Internal Clock Operation ..... 308
Figure 14.14 Count Timing at External Clock Operation (Both Edges Detected) ..... 309
Figure 14.15 Example of Setting Procedure for Waveform Output by Compare Match ..... 309

Figure 14.16 Example of 0 Output/1 Output Operation ..... 310
Figure 14.17 Example of Toggle Output Operation ..... 311
Figure 14.18 Output Compare Timing ..... 311
Figure 14.19 Example of Input Capture Operation Setting Procedure ..... 312
Figure 14.20 Example of Input Capture Operation ..... 313
Figure 14.21 Input Capture Signal Timing ..... 314
Figure 14.22 Example of Synchronous Operation Setting Procedure ..... 315
Figure 14.23 Example of Synchronous Operation ..... 316
Figure 14.24 Example of PWM Mode Setting Procedure ..... 317
Figure 14.25 Example of PWM Mode Operation (1) ..... 318
Figure 14.26 Example of PWM Mode Operation (2) ..... 319
Figure 14.27 Example of PWM Mode Operation (3) ..... 320
Figure 14.28 Example of PWM Mode Operation (4) ..... 321
Figure 14.29 Example of Reset Synchronous PWM Mode Setting Procedure ..... 323
Figure 14.30 Example of Reset Synchronous PWM Mode Operation (OLS0 = OLS1 = 1) ..... 324
Figure 14.31 Example of Reset Synchronous PWM Mode Operation (OLS0 = OLS1 = 0) ..... 325
Figure 14.32 Example of Complementary PWM Mode Setting Procedure ..... 327
Figure 14.33 Canceling Procedure of Complementary PWM Mode ..... 327
Figure 14.34 Example of Complementary PWM Mode Operation (1) ..... 328
Figure 14.35 Example of Complementary PWM Mode Operation (2) ..... 329
Figure 14.36 Timing of Overshooting ..... 330
Figure 14.37 Timing of Undershooting ..... 330
Figure 14.38 Block Diagram in PWM3 Mode ..... 333
Figure 14.39 Flowchart of Setting in PWM3 Mode ..... 334
Figure 14.40 Example of Non-Overlap Pulses ..... 335
Figure 14.41 Compare Match Buffer Operation ..... 336
Figure 14.42 Input Capture Buffer Operation ..... 337
Figure 14.43 Example of Buffer Operation Setting Procedure ..... 338
Figure 14.44 Example of Buffer Operation (1)
(Buffer Operation for Output Compare Register) ..... 339
Figure 14.45 Example of Compare Match Timing for Buffer Operation ..... 340
Figure 14.46 Example of Buffer Operation (2)
(Buffer Operation for Input Capture Register) ..... 341
Figure 14.47 Input Capture Timing of Buffer Operation ..... 341
Figure 14.48 Buffer Operation (3)
(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1) ..... 342
Figure 14.49 Buffer Operation (4)
(Buffer Operation in Complementary PWM Mode CMD1 =1, CMD0 = 0) ..... 343
Figure 14.50 Example of Output Disable Timing of Timer RD by Writing to TRDOER1 ..... 344
Figure 14.51 Example of Output Disable Timing of Timer RD by External Trigger ..... 344

Figure 14.52 Example of Output Inverse Timing of Timer RD by Writing to TRDFCR ..... 345
Figure 14.53 Example of Output Inverse Timing of Timer RD by Writing to POCR ..... 345
Figure 14.54 Block Diagram of Digital Filter ..... 346
Figure 14.55 Block Diagram of Output Pins for GR ..... 347
Figure 14.56 Example of Non-Overlapped Pulses Output on Pins FTIOA0 and FTIOB0 (TRDCNT_0 Used) ..... 348
Figure 14.57 Example of Non-Overlapped Pulses Output on Pins FTIOA1 and FTIOB1 (TRDCNT_1 Used) ..... 348
Figure 14.58 IMF Flag Set Timing when Compare Match Occurs ..... 349
Figure 14.59 IMF Flag Set Timing at Input Capture ..... 350
Figure 14.60 OVF Flag Set Timing ..... 350
Figure 14.61 Status Flag Clearing Timing ..... 351
Figure 14.62 Conflict between TRDCNT Write and Clear Operations ..... 352
Figure 14.63 Conflict between TRDCNT Write and Increment Operations ..... 352
Figure 14.64 Conflict between GR Write and Compare Match ..... 353
Figure 14.65 Conflict between TRDCNT Write and Overflow ..... 354
Figure 14.66 Conflict between GR Read and Input Capture ..... 355
Figure 14.67 Conflict between Count Clearing and Increment Operations by Input Capture ..... 355
Figure 14.68 Conflict between GR Write and Input Capture ..... 356
Figure 14.69 When Compare Match and Bit Manipulation Instruction to TRDOCR Occur at the Same Timing ..... 358

**Section 15 Watchdog Timer**

Figure 15.1 Block Diagram of Watchdog Timer ..... 359
Figure 15.2 Watchdog Timer Operation Example ..... 363

**Section 16 14-Bit PWM**

Figure 16.1 Block Diagram of 14-Bit PWM ..... 365
Figure 16.2 Waveform Output by 14-Bit PWM ..... 368

**Section 17 Serial Communication Interface 3 (SCI3)**

Figure 17.1 Block Diagram of SCI3 ..... 372
Figure 17.2 Data Format in Asynchronous Communication ..... 383
Figure 17.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode)(Example with 8-Bit Data, Parity, Two Stop Bits) ..... 383
Figure 17.4 Sample SCI3 Initialization Flowchart ..... 384
Figure 17.5 Example of SCI3 Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit) ..... 385
Figure 17.6 Sample Serial Transmission Data Flowchart (Asynchronous Mode) ..... 386
Figure 17.7 Example of SCI3 Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit) ..... 387
Figure 17.8 Sample Serial Reception Data Flowchart (Asynchronous Mode) (1) ..... 389

Figure 17.8 Sample Serial Reception Data Flowchart (Asynchronous Mode) (2) ..... 390
Figure 17.9 Data Format in Clock Synchronous Communication ..... 391
Figure 17.10 Example of SCI3 Transmission in Clock Synchronous Mode ..... 392
Figure 17.11 Sample Serial Transmission Flowchart (Clock Synchronous Mode) ..... 393
Figure 17.12 Example of SCI3 Reception in Clock Synchronous Mode ..... 394
Figure 17.13 Sample Serial Reception Flowchart (Clock Synchronous Mode) ..... 395
Figure 17.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations (Clock Synchronous Mode) ..... 396
Figure 17.15 Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A) ..... 398
Figure 17.16 Sample Multiprocessor Serial Transmission Flowchart ..... 399
Figure 17.17 Sample Multiprocessor Serial Reception Flowchart (1) ..... 400
Figure 17.17 Sample Multiprocessor Serial Reception Flowchart (2) ..... 401
Figure 17.18 Example of SCI3 Reception Using Multiprocessor Format (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit) ..... 402
Figure 17.19 Receive Data Sampling Timing in Asynchronous Mode ..... 405

**Section 18 $I^2C$ Bus Interface 2 (IIC2)**

Figure 18.1 Block Diagram of $I^2C$ Bus Interface 2 ..... 408
Figure 18.2 External Circuit Connections of I/O Pins ..... 409
Figure 18.3 $I^2C$ Bus Formats ..... 423
Figure 18.4 $I^2C$ Bus Timing ..... 423
Figure 18.5 Master Transmit Mode Operation Timing (1) ..... 425
Figure 18.6 Master Transmit Mode Operation Timing (2) ..... 425
Figure 18.7 Master Receive Mode Operation Timing (1) ..... 427
Figure 18.8 Master Receive Mode Operation Timing (2) ..... 427
Figure 18.9 Slave Transmit Mode Operation Timing (1) ..... 429
Figure 18.10 Slave Transmit Mode Operation Timing (2) ..... 430
Figure 18.11 Slave Receive Mode Operation Timing (1) ..... 431
Figure 18.12 Slave Receive Mode Operation Timing (2) ..... 432
Figure 18.13 Clocked Synchronous Serial Transfer Format ..... 432
Figure 18.14 Transmit Mode Operation Timing ..... 433
Figure 18.15 Receive Mode Operation Timing ..... 434
Figure 18.16 Block Diagram of Noise Canceller ..... 435
Figure 18.17 Sample Flowchart for Master Transmit Mode ..... 436
Figure 18.18 Sample Flowchart for Master Receive Mode ..... 437
Figure 18.19 Sample Flowchart for Slave Transmit Mode ..... 438
Figure 18.20 Sample Flowchart for Slave Receive Mode ..... 439
Figure 18.21 Timing of Bit Synchronous Circuit ..... 441

**Section 19 A/D Converter**
Figure 19.1 Block Diagram of A/D Converter ..... 446
Figure 19.2 A/D Conversion Timing ..... 453
Figure 19.3 External Trigger Input Timing ..... 454
Figure 19.4 A/D Conversion Accuracy Definitions (1) ..... 456
Figure 19.4 A/D Conversion Accuracy Definitions (2) ..... 457
Figure 19.5 Analog Input Circuit Example ..... 458

**Section 20 Band-Gap Regulator, Power-On Reset (Optional), and Low-Voltage Detection Circuits (Optional)**
Figure 20.1 Block Diagram around BGR ..... 462
Figure 20.2 Block Diagram of Power-On Reset Circuit and Low-Voltage Detection Circuit ..... 463
Figure 20.3 Operational Timing of Power-On Reset Circuit ..... 467
Figure 20.4 Operational Timing of LVDR Circuit ..... 468
Figure 20.5 Operational Timing of LVDI Circuit ..... 469

**Section 21 Power Supply Circuit**
Figure 21.1 Power Supply Connection when Internal Step-Down Circuit is Used ..... 471
Figure 21.2 Power Supply Connection when Internal Step-Down Circuit is Not Used ..... 472

**Section 23 Electrical Characteristics**
Figure 23.1 System Clock Input Timing ..... 520
Figure 23.2 $\overline{\text{RES}}$ Low Width Timing ..... 520
Figure 23.3 Input Timing ..... 521
Figure 23.4 $I^2C$ Bus Interface Input/Output Timing ..... 521
Figure 23.5 SCK3 Input Clock Timing ..... 521
Figure 23.6 SCI Input/Output Timing in Clocked Synchronous Mode ..... 522
Figure 23.7 Output Load Circuit ..... 522

**Appendix**
Figure B.1 Port 1 Block Diagram (P17) ..... 553
Figure B.2 Port 1 Block Diagram (P16, P14) ..... 554
Figure B.3 Port 1 Block Diagram (P15) ..... 554
Figure B.4 Port 1 Block Diagram (P12) ..... 555
Figure B.5 Port 1 Block Diagram (P11) ..... 555
Figure B.6 Port 1 Block Diagram (P10) ..... 556
Figure B.7 Port 2 Block Diagram (P27, P26, P25, P24, P23) ..... 556
Figure B.8 Port 2 Block Diagram (P22) ..... 557
Figure B.9 Port 2 Block Diagram (P21) ..... 557
Figure B.10 Port 2 Block Diagram (P20) ..... 558
Figure B.11 Port 3 Block Diagram (P37, P36, P35, P34, P33, P32, P31, P30) ..... 558
Figure B.12 Port5 Block Diagram (P57, P56) ..... 559

Figure B.13 Port 5 Block Diagram (P55, P54, P53, P52, P51, P50) ..... 559
Figure B.14 Port 7 Block Diagram (P77) ..... 560
Figure B.15 Port 7 Block Diagram (P76) ..... 560
Figure B.16 Port 7 Block Diagram (P75) ..... 561
Figure B.17 Port 7 Block Diagram (P74) ..... 561
Figure B.18 Port 7 Block Diagram (P72) ..... 562
Figure B.19 Port 7 Block Diagram (P71) ..... 562
Figure B.20 Port 7 Block Diagram (P70) ..... 563
Figure B.21 Port 8 Block Diagram (P87, P86, P85) ..... 563
Figure B.22 Port C Block Diagram (PC3, PC2, PC1, PC0) ..... 564
Figure B.23 Port D Block Diagram (PD7, PD6, PD5, PD4, PD3, PD2, PD1, PD0) ..... 564
Figure B.24 Port E Block Diagram (PE7, PE6, PE5, PE4, PE3, PE2, PE1, PE0) ..... 565
Figure B.25 Port F Block Diagram (PF7, PF6, PF5, PF4, PF3, PF2, PF1, PF0) ..... 565
Figure B.26 Port G Block Diagram (PG7, PG6, PG5) ..... 566
Figure B.27 Port G Block Diagram (PG4, PG3, PG2, PG1, PG0) ..... 566
Figure B.28 Port H Block Diagram (PH7, PH6, PH5, PH4) ..... 567
Figure B.29 Port H Block Diagram (PH3) ..... 567
Figure B.30 Port H Block Diagram (PH2) ..... 568
Figure B.31 Port H Block Diagram (PH1) ..... 568
Figure B.32 Port H Block Diagram (PH0) ..... 569
Figure B.33 Port J Block Diagram (PJ1) ..... 569
Figure B.34 Port J Block Diagram (PJ0) ..... 570
Figure D.1 FP-100A Package Dimensions ..... 573
Figure D.2 FP-100U Package Dimensions ..... 574

# Tables

**Section 1 Overview**
Table 1.1 Pin Functions ..... 6

**Section 2 CPU**
Table 2.1 Operation Notation ..... 20
Table 2.2 Data Transfer Instructions ..... 21
Table 2.3 Arithmetic Operations Instructions (1) ..... 22
Table 2.3 Arithmetic Operations Instructions (2) ..... 23
Table 2.4 Logic Operations Instructions ..... 24
Table 2.5 Shift Instructions ..... 24
Table 2.6 Bit Manipulation Instructions (1) ..... 25
Table 2.6 Bit Manipulation Instructions (2) ..... 26
Table 2.7 Branch Instructions ..... 27
Table 2.8 System Control Instructions ..... 28
Table 2.9 Block Data Transfer Instructions ..... 29
Table 2.10 Addressing Modes ..... 31
Table 2.11 Absolute Address Access Ranges ..... 32
Table 2.12 Effective Address Calculation (1) ..... 34
Table 2.12 Effective Address Calculation (2) ..... 35

**Section 3 Exception Handling**
Table 3.1 Exception Sources and Vector Address ..... 46
Table 3.2 Interrupt request and ICR ..... 55
Table 3.3 Interrupt Wait States ..... 62

**Section 4 Address Break**
Table 4.1 Access and Data Bus Used ..... 67

**Section 5 Clock Pulse Generators**
Table 5.1 Crystal Resonator Parameters ..... 85

**Section 6 Power-Down Modes**
Table 6.1 Operating Frequency and Waiting Time ..... 92
Table 6.2 Transition Mode after SLEEP Instruction Execution and Transition Mode due to Interrupt ..... 98
Table 6.3 Internal State in Each Operating Mode ..... 99

**Section 7 ROM**
Table 7.1 Setting Programming Modes ..... 111
Table 7.2 Boot Mode Operation ..... 113

Table 7.3 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible ..... 114
Table 7.4 Reprogramming Data Computation Table ..... 117
Table 7.5 Additional-Program Data Computation Table ..... 117
Table 7.6 Programming Time ..... 117
Table 7.7 Flash Memory Operating States ..... 122

**Section 10 Realtime Clock (RTC)**
Table 10.1 Pin Configuration ..... 188
Table 10.2 Interrupt Sources ..... 197

**Section 11 Timer B1**
Table 11.1 Pin Configuration ..... 199
Table 11.2 Timer B1 Operating Modes ..... 202

**Section 12 Timer V**
Table 12.1 Pin Configuration ..... 204
Table 12.2 Clock Signals to Input to TCNTV and Counting Conditions ..... 207

**Section 13 Timer RC**
Table 13.1 Timer RC Functions ..... 218
Table 13.2 Pin Configuration ..... 220
Table 13.3 Pin Configuration in PWM2 Mode and GR Registers ..... 247

**Section 14 Timer RD**
Table 14.1 Timer RD Functions ..... 267
Table 14.2 Channel Configuration of Timer RD ..... 268
Table 14.3 Pin Configuration ..... 272
Table 14.4 Initial Output Level of FTIOB0 Pin ..... 317
Table 14.5 Output Pins in Reset Synchronous PWM Mode ..... 322
Table 14.6 Register Settings in Reset Synchronous PWM Mode ..... 322
Table 14.7 Output Pins in Complementary PWM Mode ..... 326
Table 14.8 Register Settings in Complementary PWM Mode ..... 326
Table 14.9 Pin Configuration in PWM3 Mode and GR Registers ..... 333
Table 14.10 Register Combinations in Buffer Operation ..... 336

**Section 16 14-Bit PWM**
Table 16.1 Pin Configuration ..... 365

**Section 17 Serial Communication Interface 3 (SCI3)**
Table 17.1 Channel Configuration ..... 370
Table 17.2 Pin Configuration ..... 372
Table 17.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1) ..... 380
Table 17.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2) ..... 381

Table 17.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode) ..... 382
Table 17.5 Examples of BRR Settings for Various Bit Rates (Clock Synchronous Mode) .... 382
Table 17.6 SSR Status Flags and Receive Data Handling ..... 388
Table 17.7 SCI3 Interrupt Requests ..... 403

**Section 18 $I^2C$ Bus Interface 2 (IIC2)**
Table 18.1 Pin Configuration ..... 409
Table 18.2 Transfer Rate ..... 412
Table 18.3 Interrupt Requests ..... 440
Table 18.4 Time for Monitoring SCL ..... 441

**Section 19 A/D Converter**
Table 19.1 Pin Configuration ..... 447
Table 19.2 Analog Input Channels and Corresponding ADDR Registers ..... 448
Table 19.3 A/D Conversion Time (Single Mode) ..... 454

**Section 20 Band-Gap Regulator, Power-On Reset (Optional), and Low-Voltage Detection Circuits (Optional)**
Table 20.1 LVDCR Settings and Select Functions ..... 465

**Section 23 Electrical Characteristics**
Table 23.1 Absolute Maximum Ratings ..... 499
Table 23.2 DC Characteristics (1) ..... 502
Table 23.2 DC Characteristics (2) ..... 509
Table 23.3 AC Characteristics ..... 510
Table 23.4 $I^2C$ Bus Interface Timing ..... 513
Table 23.5 Serial Communication Interface (SCI) Timing ..... 514
Table 23.6 A/D Converter Characteristics ..... 515
Table 23.7 Watchdog Timer Characteristics ..... 516
Table 23.8 Flash Memory Characteristics ..... 517
Table 23.9 Power-Supply-Voltage Detection Circuit Characteristics ..... 519
Table 23.10 Power-On Reset Circuit Characteristics ..... 520

**Appendix**
Table A.1 Instruction Set ..... 525
Table A.2 Operation Code Map (1) ..... 538
Table A.2 Operation Code Map (2) ..... 539
Table A.2 Operation Code Map (3) ..... 540
Table A.3 Number of Cycles in Each Instruction ..... 542
Table A.4 Number of Cycles in Each Instruction ..... 543
Table A.5 Combinations of Instructions and Addressing Modes ..... 552

# Section 1 Overview

## 1.1 Features

- High-speed H8/300H central processing unit with an internal 16-bit architecture
  Upward-compatible with H8/300 CPU on an object level
  Sixteen 16-bit general registers
  62 basic instructions
- Various peripheral functions
  RTC (can be used as a free running counter)
  Timer B1 (8-bit timer)
  Timer V (8-bit timer)
  Timer RC (16-bit timer)
  Timer RD (16-bit timer)
  14-bit PWM
  Watchdog timer
  SCI3 (Asynchronous or clock synchronous serial communication interface)
  $I^2C$ bus interface 2 (conforms to the $I^2C$ bus interface format that is advocated by Philips Electronics)
  10-bit A/D converter
  POR/LVD (Power-on reset and low-voltage detection circuit) (optional)
- On-chip memory

| | | Model | | | | |
|---|---|---|---|---|---|---|
| **Product Classification** | | **Standard Version** | **On-Chip Power-On Reset and Low-Voltage Detection Circuit Version** | **ROM** | **RAM** | **Remark** |
| Flash memory version (F-ZTAT™ version) | H8/36109F | HD64F36109 | HD64F36109G | 128 kbytes | 5 kbytes | |

Note: F-ZTAT™ is a trademark of Renesas Electronics Corp.

- General I/O ports
  - I/O pins: 79 I/O pins, including 20 large current ports ($I_{OL}$ = 20 mA @$V_{OL}$ = 1.5 V)
  - Input-only pins: 8 input pins (also used for analog input)
- Supports various power-down modes
- Compact package

| Package | Code | Body Size | Pin Pitch |
|---|---|---|---|
| QFP-100 | FP-100A | 20.0 × 14.0 mm | 0.65 mm |
| LQFP-100 | FP-100U | 14.0 × 14.0 mm | 0.5 mm |

## 1.2 Internal Block Diagram

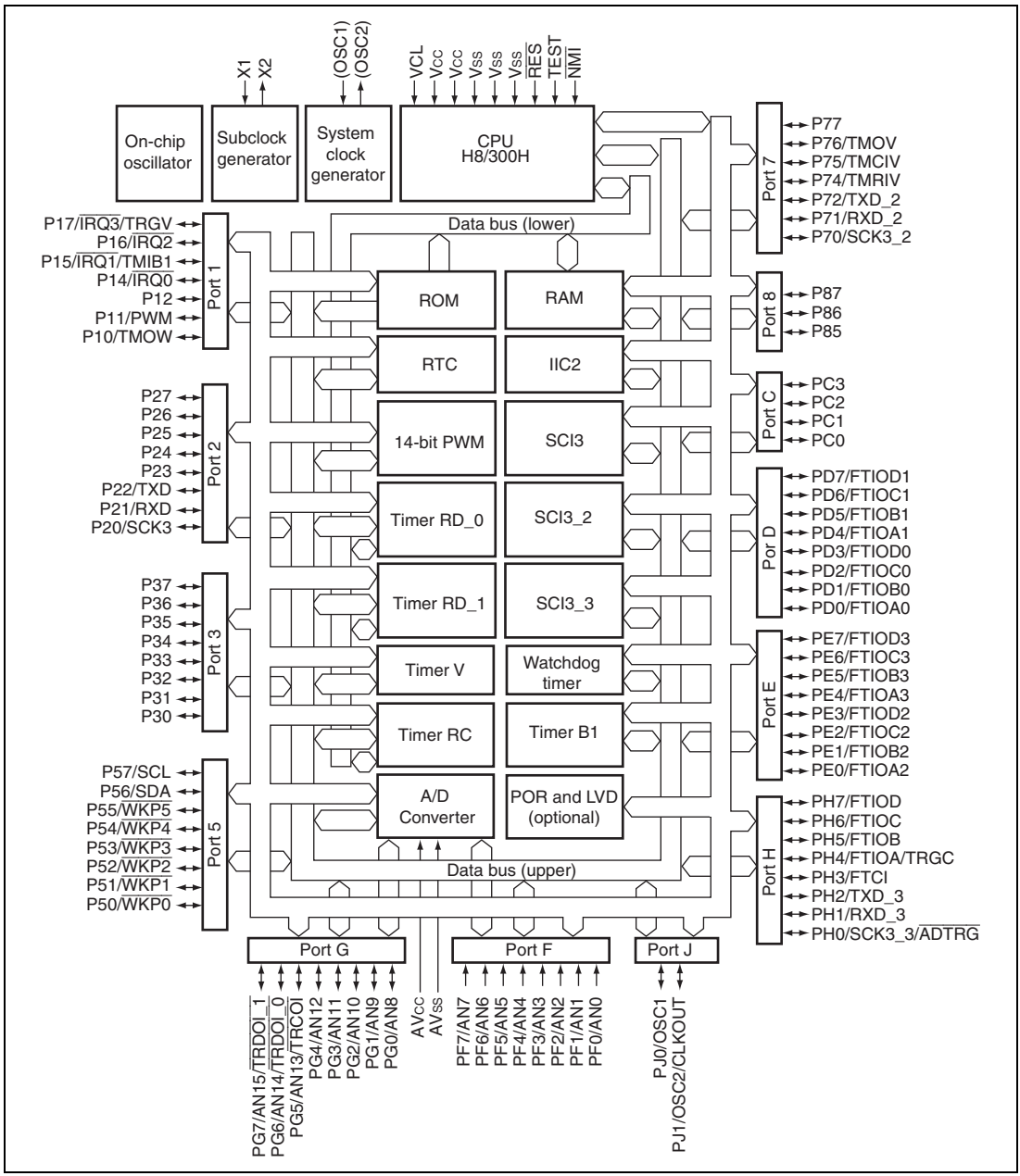



**Figure 1.1 Internal Block Diagram**

## 1.3 Pin Assignment

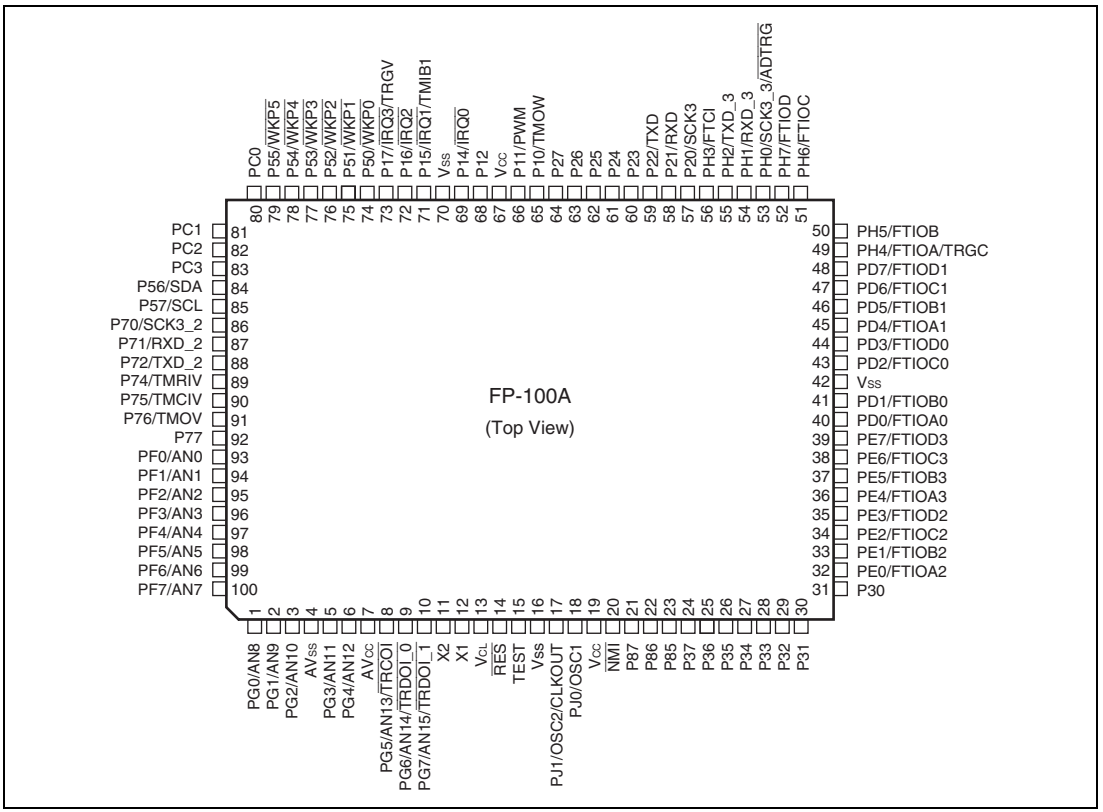



**Figure 1.2 Pin Assignments (FP-100A)**

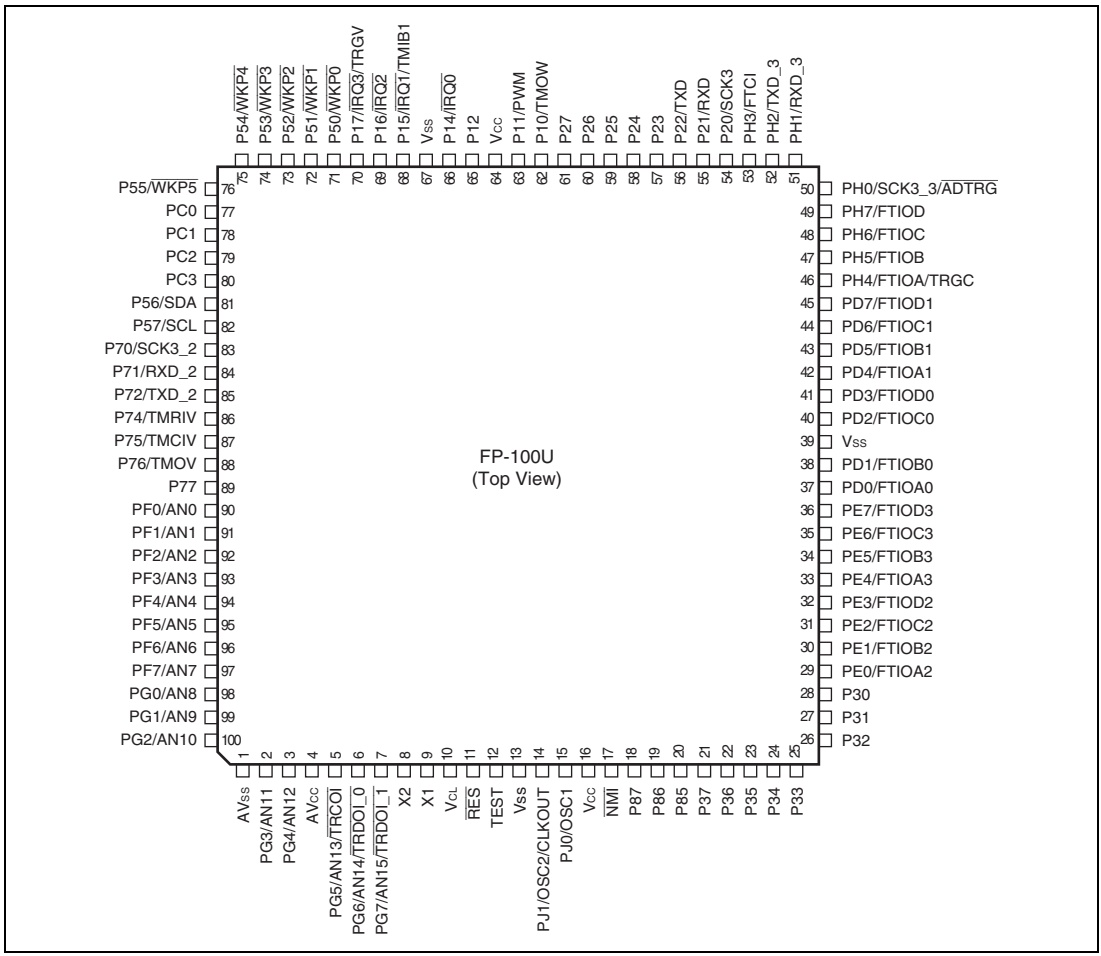



**Figure 1.3 Pin Assignments (FP-100U)**

## 1.4 Pin Functions

**Table 1.1 Pin Functions**

| | | Pin No. | | | |
|---|---|---|---|---|---|
| **Type** | **Symbol** | **FP-100A** | **FP-100U** | **I/O** | **Functions** |
| Power supply pins | Vcc | 19, 67 | 16, 64 | Input | Power supply pin. Connect this pin to the system power supply. |
| | Vss | 16, 42, 70 | 13, 39, 67 | Input | Ground pin. Ensure to connect all pins to the system power supply (0 V). |
| | AVcc | 7 | 4 | Input | Analog power supply pin for the A/D converter. When the A/D converter is not used, connect this pin to the system power supply. |
| | AVss | 4 | 1 | Input | Analog ground pin for the A/D converter. Connect this pin to the system power supply (0 V). |
| | $V_{CL}$ | 13 | 10 | Input | Internal step-down power supply pin. Connect a capacitor of around 0.1 μF between this pin and the Vss pin for stabilization. |
| Clock pins | OSC1 | 18 | 15 | Input | These pins connect with crystal or ceramic resonator for the system clock, or can be used to input an external clock. When using the on-chip oscillator, the system clock can be output from OSC2 pin. See section 5, Clock Pulse Generators, for a typical connection. |
| | OSC2/ CLKOUT | 17 | 14 | Output | |
| | X1 | 12 | 9 | Input | These pins connect with a 32.768 kHz crystal resonator for the subclock. See section 5, Clock Pulse Generators, for a typical connection. |
| | X2 | 11 | 8 | Output | |
| System control | $\overline{RES}$ | 14 | 11 | Input | Reset pin. The pull-up resistor (typ.150 kΩ) is incorporated. When driven low, the chip is reset. |
| | TEST | 15 | 12 | Input | Test pin. Connect this pin to Vss. |

| Type | Symbol | Pin No. FP-100A | Pin No. FP-100U | I/O | Functions |
|---|---|---|---|---|---|
| External interrupt pins | $\overline{\text{NMI}}$ | 20 | 17 | Input | Non-maskable interrupt request input pin. Be sure to pull-up by a pull-up resistor. |
| | $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$ | 69, 71 to 73 | 66, 68, 69, 70 | Input | External interrupt request input pins. Can select the rising or falling edge. |
| | $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$ | 74 to 79 | 71 to 76 | Input | External interrupt request input pins. Can select the rising or falling edge. |
| RTC | TMOW | 65 | 62 | Output | This is an output pin for divided clocks. |
| Timer B1 | TMIB1 | 71 | 68 | Input | External event input pin. |
| Timer V | TMOV | 91 | 88 | Output | This is an output pin for waveforms generated by the output compare function. |
| | TMCIV | 90 | 87 | Input | External event input pin. |
| | TMRIV | 89 | 86 | Input | Counter reset input pin. |
| | TRGV | 73 | 70 | Input | Count start trigger input pin. |
| Timer RC | FTCI | 56 | 53 | Input | External event input pin. |
| | FTIOA to FTIOD | 49 to 52 | 46 to 49 | I/O | Output compare output/input capture input/PWM output pin. |
| | TRGC | 49 | 46 | Input | External trigger input pin. |
| | $\overline{\text{TRCOI}}$ | 8 | 5 | Input | Input pin for the timer output enable/disable signal. |
| Timer RD_0 | FTIOA0 | 40 | 37 | I/O | Output compare output/input capture input/external clock input pin. |
| | FTIOB0 | 41 | 38 | I/O | Output compare output/input capture input/PWM output pin. |
| | FTIOC0 | 43 | 40 | I/O | Output compare output/input capture input/PWM synchronous output pin (at a reset or in complementary PWM mode). |
| | FTIOD0 | 44 | 41 | I/O | Output compare output/input capture input/PWM output pin. |
| | FTIOA1 | 45 | 42 | I/O | Output compare output/input capture input/PWM output pin (at a reset or in complementary PWM mode). |
| | FTIOB1 to FTIOD1 | 46 to 48 | 43 to 45 | I/O | Output compare output/input capture input/PWM output pin. |
| | $\overline{\text{TRDOI_0}}$ | 9 | 6 | Input | Input pin for the timer output enable/disable signal. |

| Type | Symbol | Pin No. FP-100A | Pin No. FP-100U | I/O | Functions |
|---|---|---|---|---|---|
| Timer RD_1 | FTIOA2 | 32 | 29 | I/O | Output compare output/input capture input/external clock input pin |
| | FTIOB2 | 33 | 30 | I/O | Output compare output/input capture input/PWM output pin |
| | FTIOC2 | 34 | 31 | I/O | Output compare output/input capture input/PWM synchronous output pin (at a reset or in complementary PWM mode) |
| | FTIOD2 | 35 | 32 | I/O | Output compare output/input capture input/PWM output pin |
| | FTIOA3 | 36 | 33 | I/O | Output compare output/input capture input/PWM output pin (at a reset or in complementary PWM mode) |
| | FTIOB3 to FTIOD3 | 37 to 39 | 34 to 36 | I/O | Output compare output/input capture input/PWM output pin |
| | $\overline{\text{TRDOI_1}}$ | 10 | 7 | Input | Input pin for the timer output enable/disable signal. |
| $I^2C$ bus interface 2 (IIC2) | SDA | 84 | 81 | I/O | IIC data I/O pin. Can directly drive a bus by NMOS open-drain output. When using this pin, external pull-up resistor is required. |
| | SCL | 85 | 82 | I/O | IIC clock I/O pin. Can directly drive a bus by NMOS open-drain output. When using this pin, external pull-up resistor is required. |
| Serial communication interface 3 (SCI3) | TXD, TXD_2, TXD_3 | 59, 88, 55 | 56, 85, 52 | Output | Transmit data output pin |
| | RXD, RXD_2, RXD_3 | 58, 87, 54 | 55, 84, 51 | Input | Receive data input pin |
| | SCK3, SCK3_2, SCK3_3 | 57, 86, 53 | 54, 83, 50 | I/O | Clock I/O pin |

| Type | Symbol | Pin No. FP-100A | Pin No. FP-100U | I/O | Functions |
|---|---|---|---|---|---|
| 14-bit PWM | PWM | 66 | 63 | Output | 14-bit PWM square wave output pin |
| A/D converter | AN15 to AN0 | 10 to 8, 6, 5, 3 to 1, 100 to 93 | 7 to 5, 3, 2, 100 to 90 | Input | Analog input pin |
| | $\overline{\text{ADTRG}}$ | 53 | 50 | Input | Conversion start trigger input pin |
| I/O ports | PF7 to PF0 | 100 to 93 | 97 to 90 | Input | 8-bit input port |
| | P17 to P14, P12 to P10 | 73 to 71, 69, 68, 66, 65 | 70 to 68, 66, 65, 63, 62 | I/O | 7-bit I/O port |
| | P27 to P20 | 64 to 57 | 61 to 54 | I/O | 8-bit I/O port |
| | P37 to P30 | 24 to 31 | 21 to 28 | I/O | 8-bit I/O port |
| | P57 to P50 | 85, 84, 79 to 74 | 82, 81, 76 to 71 | I/O | 8-bit I/O port |
| | P77 to P74, P72 to P70 | 92 to 89, 88 to 86 | 89 to 86, 85 to 83 | I/O | 7-bit I/O port |
| | P87 to P85 | 21 to 23 | 18 to 20 | I/O | 3-bit I/O port |
| | PC3 to PC0 | 83 to 80 | 80 to 77 | I/O | 4-bit I/O port |
| | PD7 to PD0 | 48 to 43, 41, 40 | 45 to 40, 38, 37 | I/O | 8-bit I/O port |
| | PE7 to PE0 | 39 to 32 | 36 to 29 | I/O | 8-bit I/O port |
| | PG7 to PG0 | 10 to 8, 6, 5, 3 to 1 | 7 to 5, 3, 2, 100 to 98 | I/O | 8-bit I/O port |
| | PH7 to PH0 | 52 to 49, 56 to 53 | 49 to 46, 53 to 50 | I/O | 8-bit I/O port |
| | PJ1, PJ0 | 17, 18 | 14, 15 | I/O | 2-bit I/O port |

# Section 2 CPU

This LSI has an H8/300H CPU with an internal 32-bit architecture that is upward-compatible with the H8/300CPU, and supports only advanced mode, which has a 16-Mbyte address space.

- Upward-compatible with H8/300 CPUs
  Can execute H8/300 CPUs object programs
  Additional eight 16-bit extended registers
  32-bit transfer and arithmetic and logic instructions are added
  Signed multiply and divide instructions are added.
- General-register architecture
  Sixteen 16-bit general registers also usable as sixteen 8-bit registers and eight 16-bit registers, or eight 32-bit registers
- 62 basic instructions
  8/16/32-bit data transfer and arithmetic and logic instructions
  Multiply and divide instructions
  Powerful bit-manipulation instructions
- Eight addressing modes
  Register direct [Rn]
  Register indirect [@ERn]
  Register indirect with displacement [@(d:16,ERn) or @(d:24,ERn)]
  Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  Absolute address [@aa:8, @aa:16, @aa:24]
  Immediate [#xx:8, #xx:16, or #xx:32]
  Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  Memory indirect [@@aa:8]
- 16-Mbyte address space
- High-speed operation
  All frequently-used instructions execute in two to four states
  8/16/32-bit register-register add/subtract: 2 state
  8 × 8-bit register-register multiply: 14 states
  16 ÷ 8-bit register-register divide: 14 states
  16 × 16-bit register-register multiply: 22 states
  32 ÷ 16-bit register-register divide: 22 states

- Power-down state
  Transition to power-down state by SLEEP instruction

## 2.1 Address Space and Memory Map

The address space of this LSI is 16 Mbytes, which includes the program area and data area.

Figure 2.1 shows the memory map.

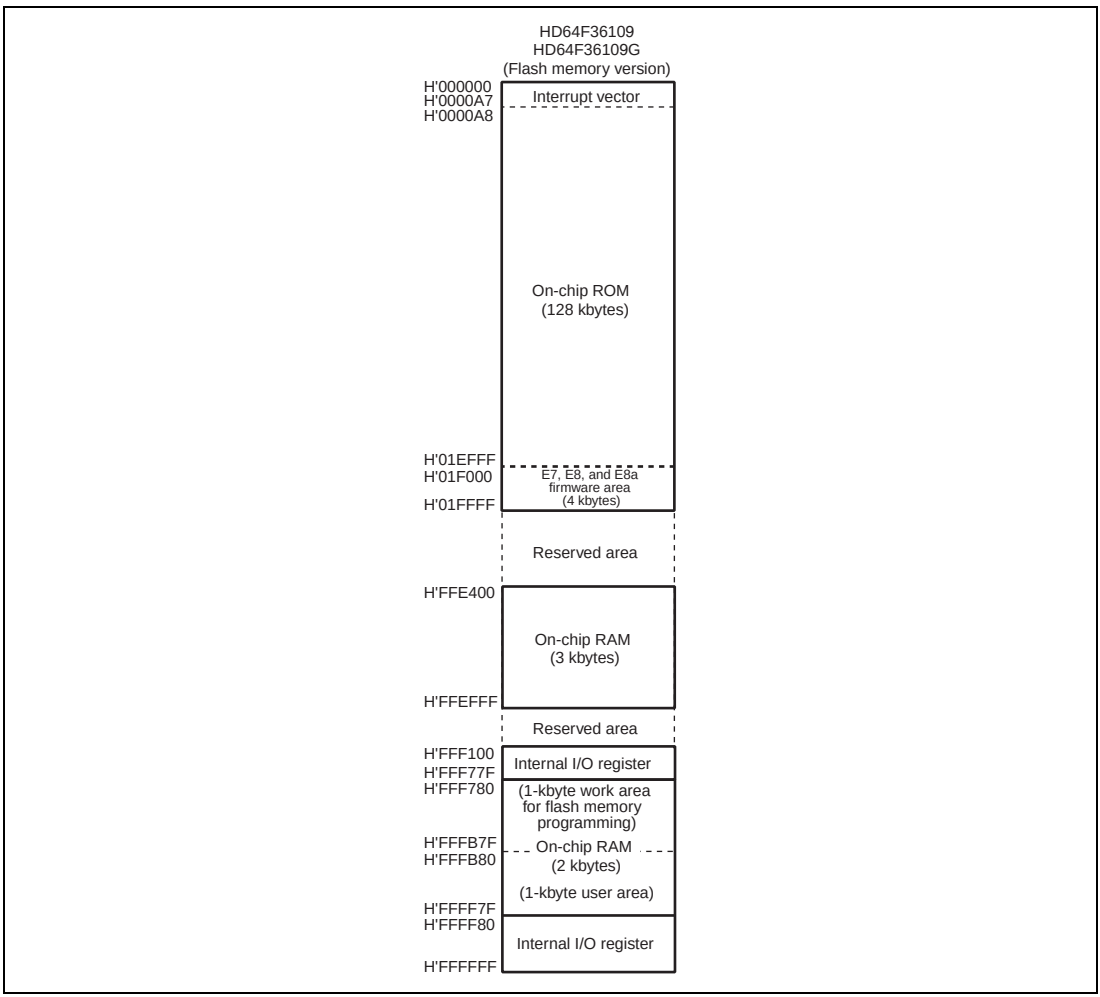



**Figure 2.1 Memory Map**

## 2.2 Register Configuration

The H8/300H CPU has the internal registers shown in figure 2.2. There are two types of registers; general registers and control registers. The control registers are a 24-bit program counter (PC), and an 8-bit condition-code register (CCR).

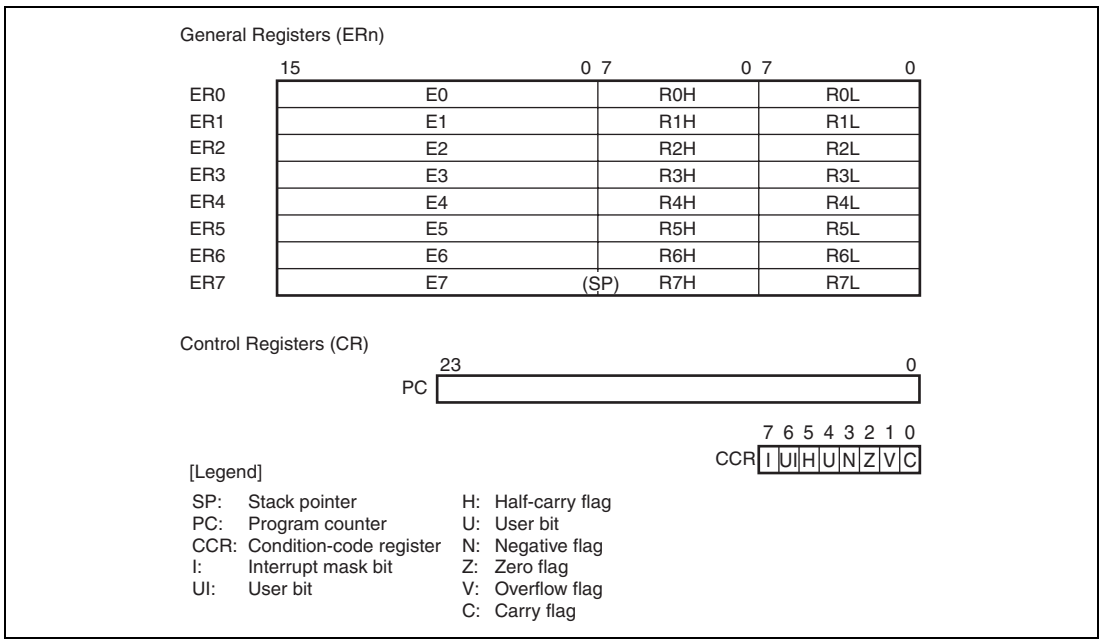



**Figure 2.2 CPU Registers**

### 2.2.1 General Registers

The H8/300H CPU has eight 32-bit general registers. These general registers are all functionally identical and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 2.3 illustrates the usage of the general registers. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

The ER registers divide into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum of sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

The R registers divide into 8-bit registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum of sixteen 8-bit registers.

The usage of each register can be selected independently.

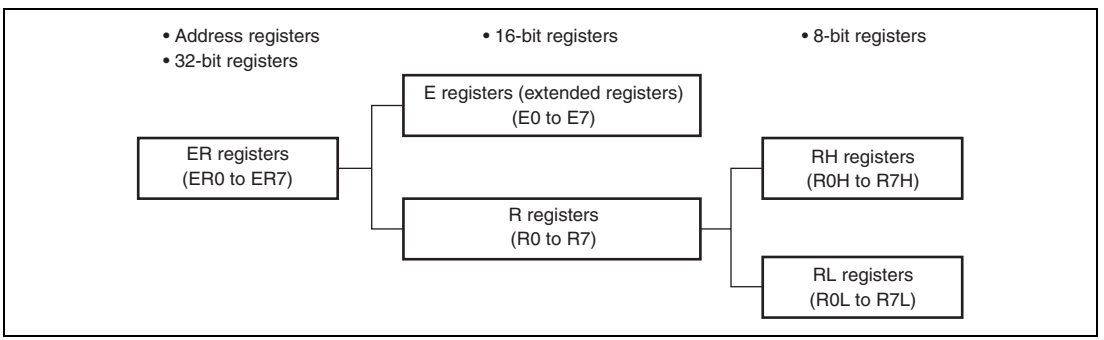



**Figure 2.3 Usage of General Registers**

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.4 shows the relationship between the stack pointer and the stack area.

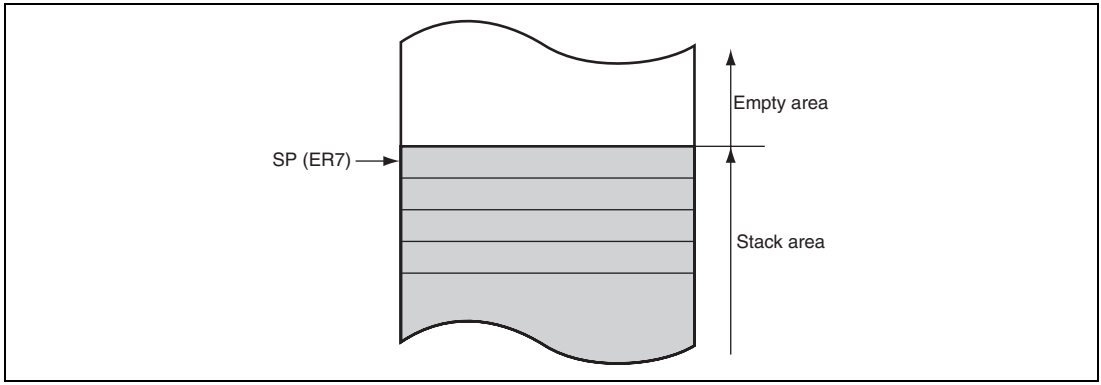



**Figure 2.4 Relationship between Stack Pointer and Stack Area**

### 2.2.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (When an instruction is fetched, the least significant PC bit is regarded as 0). The PC is initialized when the start address is loaded by the vector address generated during reset exception-handling sequence.

### 2.2.3 Condition-Code Register (CCR)

This 8-bit register contains internal CPU status information, including an interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags. The I bit is initialized to 1 by reset exception-handling sequence, but other bits are not initialized.

Some instructions leave flag bits unchanged. Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branching conditions for conditional branch (Bcc) instructions.

For the action of each instruction on the flag bits, see appendix A.1, Instruction List.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | I | 1 | R/W | Interrupt Mask Bit<br>Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence. |
| 6 | UI | Undefined | R/W | User Bit<br>Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. |
| 5 | H | Undefined | R/W | Half-Carry Flag<br>When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise. |
| 4 | U | Undefined | R/W | User Bit<br>Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. |
| 3 | N | Undefined | R/W | Negative Flag<br>Stores the value of the most significant bit of data as a sign bit. |
| 2 | Z | Undefined | R/W | Zero Flag<br>Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data. |
| 1 | V | Undefined | R/W | Overflow Flag<br>Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times. |
| 0 | C | Undefined | R/W | Carry Flag<br>Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by:<br>• Add instructions, to indicate a carry<br>• Subtract instructions, to indicate a borrow<br>• Shift and rotate instructions, to indicate a carry<br>The carry flag is also used as a bit accumulator by bit manipulation instructions. |

## 2.3 Data Formats

The H8/300H CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

### 2.3.1 General Register Data Formats

Figure 2.5 shows the data formats in general registers.

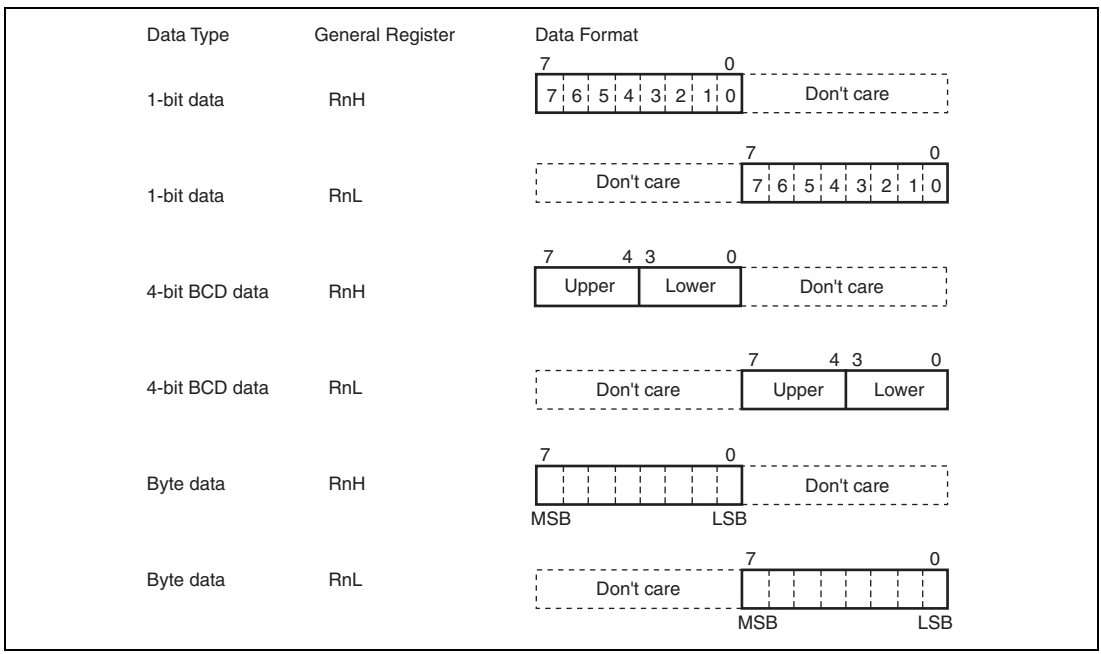



**Figure 2.5 General Register Data Formats (1)**

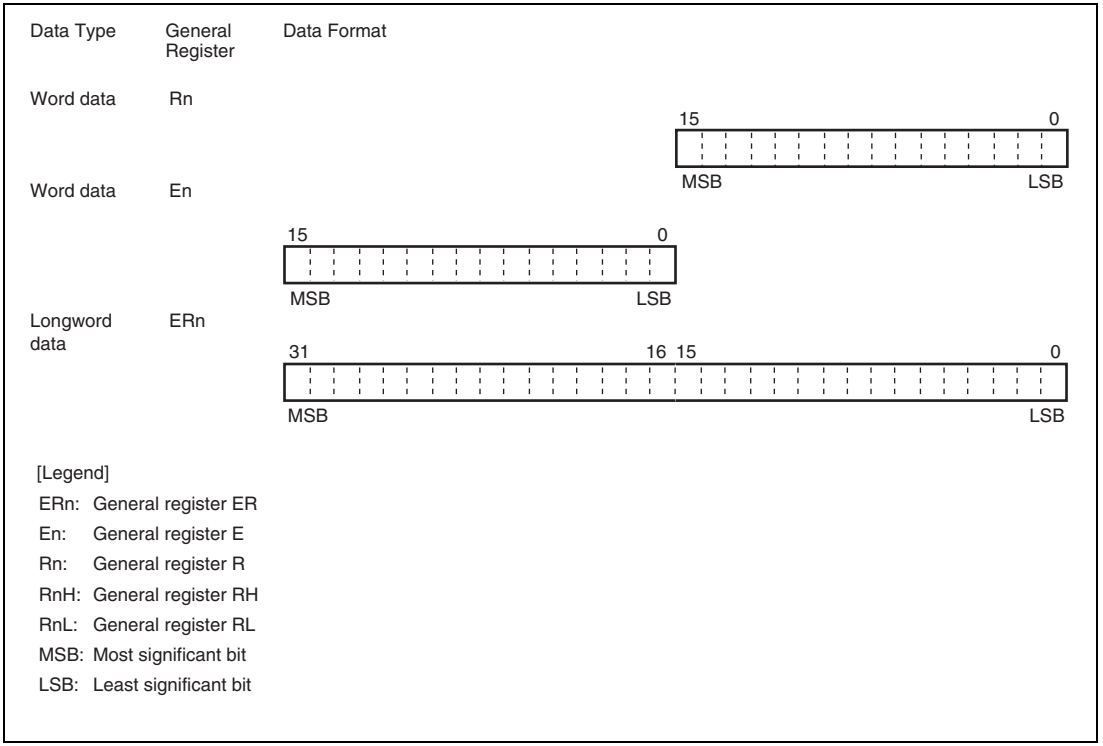



**Figure 2.5 General Register Data Formats (2)**

### 2.3.2 Memory Data Formats

Figure 2.6 shows the data formats in memory. The H8/300H CPU can access word data and longword data in memory, however word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, an address error does not occur, however the least significant bit of the address is regarded as 0, so access begins the preceding address. This also applies to instruction fetches.

When ER7 (SP) is used as an address register to access the stack area, the operand size should be word or longword.

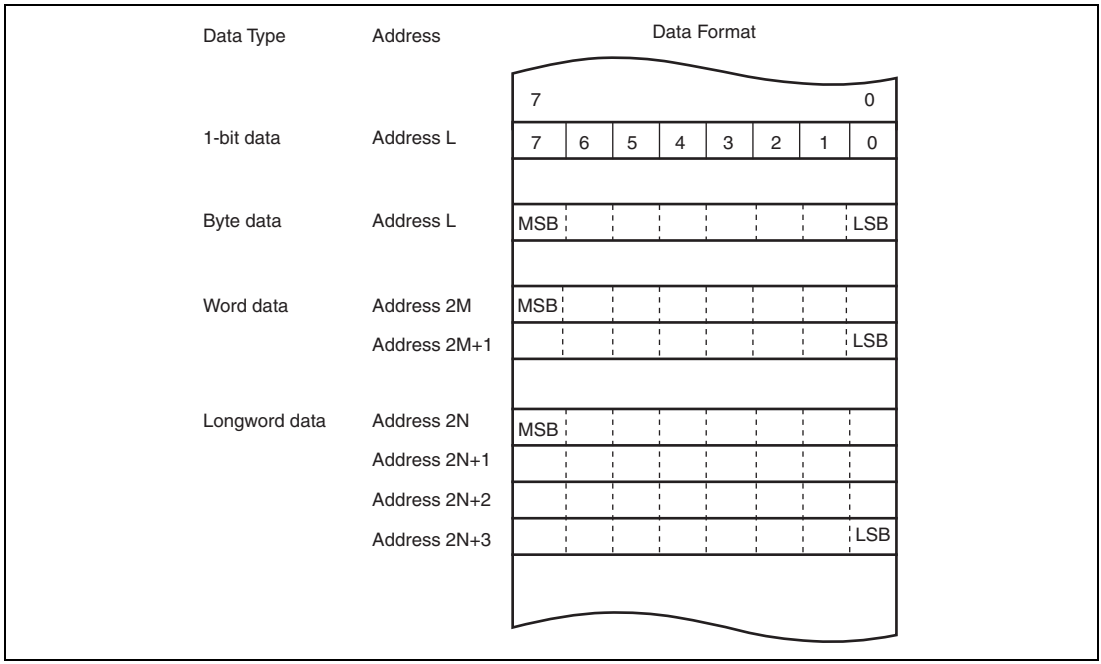



**Figure 2.6 Memory Data Formats**

## 2.4 Instruction Set

### 2.4.1 List of Instructions Classified by Function

The H8/300H CPU has 62 instructions. Tables 2.2 to 2.9 summarize the instructions in each functional category. The notation used in tables 2.2 to 2.9 is defined below.

**Table 2.1 Operation Notation**

| Symbol | Description |
|---|---|
| Rd | General register (destination)* |
| Rs | General register (source)* |
| Rn | General register* |
| ERn | General register (32-bit register or address register) |
| (EAd) | Destination operand |
| (EAs) | Source operand |
| CCR | Condition-code register |
| N | N (negative) flag in CCR |
| Z | Z (zero) flag in CCR |
| V | V (overflow) flag in CCR |
| C | C (carry) flag in CCR |
| PC | Program counter |
| SP | Stack pointer |
| #IMM | Immediate data |
| disp | Displacement |
| + | Addition |
| – | Subtraction |
| × | Multiplication |
| ÷ | Division |
| ∧ | Logical AND |
| ∨ | Logical OR |
| ⊕ | Logical XOR |
| → | Move |
| ~ | NOT (logical complement) |
| :3/:8/:16/:24 | 3-, 8-, 16-, or 24-bit length |

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers/address register (ER0 to ER7).

**Table 2.2 Data Transfer Instructions**

| Instruction | Size* | Function |
|---|---|---|
| MOV | B/W/L | (EAs) → Rd, Rs → (EAd)<br>Moves data between two general registers or between a general register and memory, or moves immediate data to a general register. |
| MOVFPE | B | (EAs) → Rd<br>Cannot be used in this LSI. |
| MOVTPE | B | Rs → (EAs)<br>Cannot be used in this LSI. |
| POP | W/L | @SP+ → Rn<br>Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn. |
| PUSH | W/L | Rn → @–SP<br>Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn, @–SP. |

[Legend]

B: Byte

W: Word

L: Longword

Note: * Refers to the operand size.

**Table 2.3 Arithmetic Operations Instructions (1)**

| Instruction | Size* | Function |
|---|---|---|
| ADD<br>SUB | B/W/L | Rd ± Rs → Rd, Rd ± #IMM → Rd<br>Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register (immediate byte data cannot be subtracted from byte data in a general register. Use the SUBX or ADD instruction.) |
| ADDX<br>SUBX | B | Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd<br>Performs addition or subtraction with carry on byte data in two general registers, or on immediate data and data in a general register. |
| INC<br>DEC | B/W/L | Rd ± 1 → Rd, Rd ± 2 → Rd<br>Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.) |
| ADDS<br>SUBS | L | Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd<br>Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register. |
| DAA<br>DAS | B | Rd (decimal adjust) → Rd<br>Decimal-adjusts an addition or subtraction result in a general register by referring to the CCR to produce 4-bit BCD data. |
| MULXU | B/W | Rd × Rs → Rd<br>Performs unsigned multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits. |
| MULXS | B/W | Rd × Rs → Rd<br>Performs signed multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits. |
| DIVXU | B/W | Rd ÷ Rs → Rd<br>Performs unsigned division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder. |

[Legend]

B: Byte

W: Word

L: Longword

Note: * Refers to the operand size.

**Table 2.3 Arithmetic Operations Instructions (2)**

| Instruction | Size* | Function |
|---|---|---|
| DIVXS | B/W | Rd ÷ Rs → Rd<br>Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder. |
| CMP | B/W/L | Rd – Rs, Rd – #IMM<br>Compares data in a general register with data in another general register or with immediate data, and sets CCR bits according to the result. |
| NEG | B/W/L | 0 – Rd → Rd<br>Takes the two's complement (arithmetic complement) of data in a general register. |
| EXTU | W/L | Rd (zero extension) → Rd<br>Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left. |
| EXTS | W/L | Rd (sign extension) → Rd<br>Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit. |

[Legend]

B: Byte

W: Word

L: Longword

Note: * Refers to the operand size.

**Table 2.4 Logic Operations Instructions**

| Instruction | Size* | Function |
|---|---|---|
| AND | B/W/L | Rd ∧ Rs → Rd, Rd ∧ #IMM → Rd<br>Performs a logical AND operation on a general register and another general register or immediate data. |
| OR | B/W/L | Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd<br>Performs a logical OR operation on a general register and another general register or immediate data. |
| XOR | B/W/L | Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd<br>Performs a logical exclusive OR operation on a general register and another general register or immediate data. |
| NOT | B/W/L | ~ (Rd) → (Rd)<br>Takes the one's complement (logical complement) of general register contents. |

[Legend]

B: Byte

W: Word

L: Longword

Note: * Refers to the operand size.

**Table 2.5 Shift Instructions**

| Instruction | Size* | Function |
|---|---|---|
| SHAL<br>SHAR | B/W/L | Rd (shift) → Rd<br>Performs an arithmetic shift on general register contents. |
| SHLL<br>SHLR | B/W/L | Rd (shift) → Rd<br>Performs a logical shift on general register contents. |
| ROTL<br>ROTR | B/W/L | Rd (rotate) → Rd<br>Rotates general register contents. |
| ROTXL<br>ROTXR | B/W/L | Rd (rotate) → Rd<br>Rotates general register contents through the carry flag. |

[Legend]

B: Byte

W: Word

L: Longword

Note: * Refers to the operand size.

**Table 2.6 Bit Manipulation Instructions (1)**

| Instruction | Size* | Function |
|---|---|---|
| BSET | B | 1 → (<bit-No.> of <EAd>)<br>Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register. |
| BCLR | B | 0 → (<bit-No.> of <EAd>)<br>Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register. |
| BNOT | B | ~ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>)<br>Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower three bits of a general register. |
| BTST | B | ~ (<bit-No.> of <EAd>) → Z<br>Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register. |
| BAND | B | C ∧ (<bit-No.> of <EAd>) → C<br>ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag. |
| BIAND | B | C ∧ ~ (<bit-No.> of <EAd>) → C<br>ANDs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. |
| BOR | B | C ∨ (<bit-No.> of <EAd>) → C<br>ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag. |
| BIOR | B | C ∨ [~ (<bit-No.> of <EAd>)] → C<br>ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data. |

[Legend]

B: Byte

Note: * Refers to the operand size.

**Table 2.6 Bit Manipulation Instructions (2)**

| Instruction | Size* | Function |
|---|---|---|
| BXOR | B | C ⊕ (<bit-No.> of <EAd>) → C<br>XORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag. |
| BIXOR | B | C ⊕ [~ (<bit-No.> of <EAd>)] → C<br>XORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag.<br>The bit number is specified by 3-bit immediate data. |
| BLD | B | (<bit-No.> of <EAd>) → C<br>Transfers a specified bit in a general register or memory operand to the carry flag. |
| BILD | B | ~ (<bit-No.> of <EAd>) → C<br>Transfers the inverse of a specified bit in a general register or memory operand to the carry flag.<br>The bit number is specified by 3-bit immediate data. |
| BST | B | C → (<bit-No.> of <EAd>)<br>Transfers the carry flag value to a specified bit in a general register or memory operand. |
| BIST | B | C → ~ (<bit-No.> of <EAd>)<br>Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand.<br>The bit number is specified by 3-bit immediate data. |

[Legend]

B: Byte

Note: * Refers to the operand size.

**Table 2.7 Branch Instructions**

| Instruction | Size | Function | | |
|---|---|---|---|---|
| Bcc* | — | Branches to a specified address if a specified condition is true. The branching conditions are listed below. | | |
| | | **Mnemonic** | **Description** | **Condition** |
| | | BRA(BT) | Always (true) | Always |
| | | BRN(BF) | Never (false) | Never |
| | | BHI | High | $C \vee Z = 0$ |
| | | BLS | Low or same | $C \vee Z = 1$ |
| | | BCC(BHS) | Carry clear (high or same) | $C = 0$ |
| | | BCS(BLO) | Carry set (low) | $C = 1$ |
| | | BNE | Not equal | $Z = 0$ |
| | | BEQ | Equal | $Z = 1$ |
| | | BVC | Overflow clear | $V = 0$ |
| | | BVS | Overflow set | $V = 1$ |
| | | BPL | Plus | $N = 0$ |
| | | BMI | Minus | $N = 1$ |
| | | BGE | Greater or equal | $N \oplus V = 0$ |
| | | BLT | Less than | $N \oplus V = 1$ |
| | | BGT | Greater than | $Z \vee (N \oplus V) = 0$ |
| | | BLE | Less or equal | $Z \vee (N \oplus V) = 1$ |
| JMP | — | Branches unconditionally to a specified address. | | |
| BSR | — | Branches to a subroutine at a specified address. | | |
| JSR | — | Branches to a subroutine at a specified address. | | |
| RTS | — | Returns from a subroutine | | |

Note: * Bcc is the general name for conditional branch instructions.

**Table 2.8 System Control Instructions**

| Instruction | Size* | Function |
|---|---|---|
| TRAPA | — | Starts trap-instruction exception handling. |
| RTE | — | Returns from an exception-handling routine. |
| SLEEP | — | Causes a transition to a power-down state. |
| LDC | B/W | (EAs) → CCR<br>Moves the source operand contents to the CCR. The CCR size is one byte, but in transfer from memory, data is read by word access. |
| STC | B/W | CCR → (EAd)<br>Transfers the CCR contents to a destination location. The condition code register size is one byte, but in transfer to memory, data is written by word access. |
| ANDC | B | CCR ∧ #IMM → CCR<br>Logically ANDs the CCR with immediate data. |
| ORC | B | CCR ∨ #IMM → CCR<br>Logically ORs the CCR with immediate data. |
| XORC | B | CCR ⊕ #IMM → CCR<br>Logically XORs the CCR with immediate data. |
| NOP | — | PC + 2 → PC<br>Only increments the program counter. |

[Legend]

B: Byte

W: Word

Note: * Refers to the operand size.

**Table 2.9 Block Data Transfer Instructions**

| Instruction | Size | Function |
|---|---|---|
| EEPMOV.B | — | if R4L ≠ 0 then<br>Repeat @ER5+ → @ER6+,<br>R4L–1 → R4L<br>Until R4L = 0<br>else next; |
| EEPMOV.W | — | if R4 ≠ 0 then<br>Repeat @ER5+ → @ER6+,<br>R4–1 → R4<br>Until R4 = 0<br>else next;<br>Transfers a data block. Starting from the address set in ER5, transfers data for the number of bytes set in R4L or R4 to the address location set in ER6.<br>Execution of the next instruction begins as soon as the transfer is completed. |

### 2.4.2 Basic Instruction Formats

H8/300H CPU instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op), a register field (r), an effective address extension (EA), and a condition field (cc).

Figure 2.7 shows examples of instruction formats.

**(1) Operation Field**

Indicates the function of the instruction, the addressing mode, and the operation to be carried out on the operand. The operation field always includes the first four bits of the instruction. Some instructions have two operation fields.

**(2) Register Field**

Specifies a general register. Address registers are specified by 3 bits, and data registers by 3 bits or 4 bits. Some instructions have two register fields. Some have no register field.

**(3) Effective Address Extension**

8, 16, or 32 bits specifying immediate data, an absolute address, or a displacement. A 24-bit address or displacement is treated as a 32-bit data in which the upper 8 bits are 0 (H'00).

**(4) Condition Field**

Specifies the branching condition of Bcc instructions.

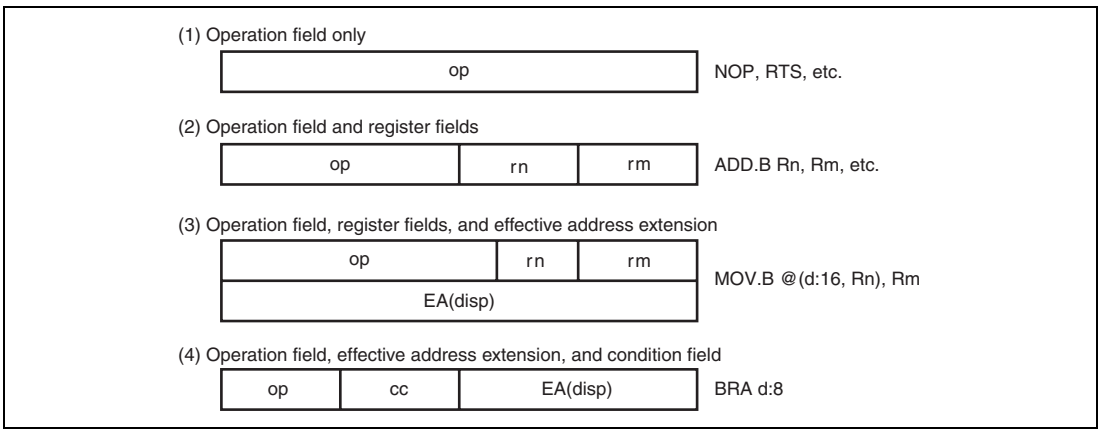



**Figure 2.7 Instruction Formats**

## 2.5 Addressing Modes and Effective Address Calculation

### 2.5.1 Addressing Modes

The H8/300H CPU supports the eight addressing modes listed in table 2.10. Each instruction uses a subset of these addressing modes. Addressing modes that can be used differ depending on the instruction. For details, refer to appendix A.4, Combinations of Instructions and Addressing Modes.

Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit-manipulation instructions use register direct, register indirect, or the absolute addressing mode (@aa:8) to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

**Table 2.10 Addressing Modes**

| No. | Addressing Mode | Symbol |
|---|---|---|
| 1 | Register direct | Rn |
| 2 | Register indirect | @ERn |
| 3 | Register indirect with displacement | @(d:16,ERn)/@(d:24,ERn) |
| 4 | Register indirect with post-increment<br>Register indirect with pre-decrement | @ERn+<br>@–ERn |
| 5 | Absolute address | @aa:8/@aa:16/@aa:24 |
| 6 | Immediate | #xx:8/#xx:16/#xx:32 |
| 7 | Program-counter relative | @(d:8,PC)/@(d:16,PC) |
| 8 | Memory indirect | @@aa:8 |

#### (1) Register Direct—Rn

The register field of the instruction specifies an 8-, 16-, or 32-bit general register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

#### (2) Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn), the lower 24 bits of which contain the address of the operand on memory.

### (3) Register Indirect with Displacement—@(d:16, ERn) or @(d:24, ERn)

A 16-bit or 24-bit displacement contained in the instruction is added to an address register (ERn) specified by the register field of the instruction, and the lower 24 bits of the sum the address of a memory operand. A 16-bit displacement is sign-extended when added.

### (4) Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn

- Register indirect with post-increment—@ERn+

  The register field of the instruction code specifies an address register (ERn) the lower 24 bits of which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents (32 bits) and the sum is stored in the address register. The value added is 1 for byte access, 2 for word access, or 4 for longword access. For the word or longword access, the register value should be even.
- Register indirect with pre-decrement—@-ERn

  The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the lower 24 bits of the result is the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word access, or 4 for longword access. For the word or longword access, the register value should be even.

### (5) Absolute Address—@aa:8, @aa:16, @aa:24

The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24)

For an 8-bit absolute address, the upper 16 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 8 bits are a sign extension. A 24-bit absolute address can access the entire address space.

The access ranges of absolute addresses for the group of this LSI are those shown in table 2.11.

**Table 2.11 Absolute Address Access Ranges**

| Absolute Address | Access Range |
|---|---|
| 8 bits (@aa:8) | H'FFFF00 to H'FFFFFF |
| 16 bits (@aa:16) | H'000000 to H'007FFF<br>H'FF8000 to H'FFFFFF |
| 24 bits (@aa:24) | H'000000 to H'FFFFFF |

### (6) Immediate—#xx:8, #xx:16, or #xx:32

The instruction contains 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data as an operand.

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, specifying a bit number. The TRAPA instruction contains 2-bit immediate data in its instruction code, specifying a vector address.

### (7) Program-Counter Relative—@(d:8, PC) or @(d:16, PC)

This mode is used in the BSR instruction. An 8-bit or 16-bit displacement contained in the instruction is sign-extended and added to the 24-bit PC contents to generate a branch address. The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

### (8) Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The memory operand is accessed by longword access. The first byte of the memory operand is ignored, generating a 24-bit branch address. Figure 2.8 shows how to specify branch address for in memory indirect mode.

The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF). Note that the first part of the address range is also the exception vector area.

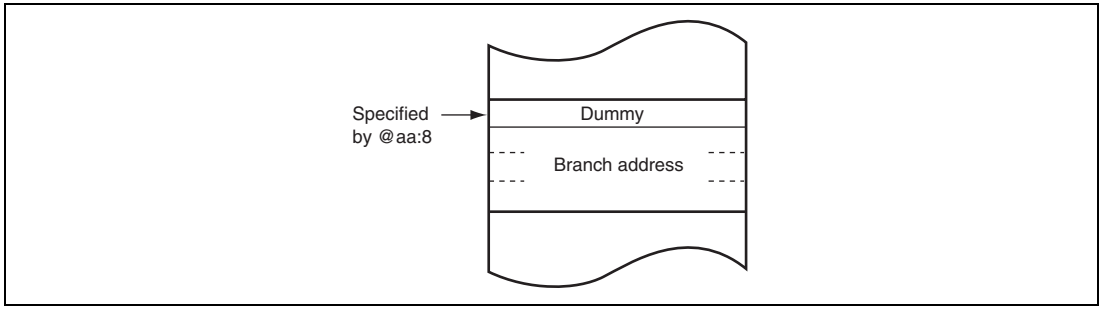
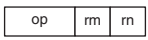
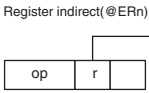
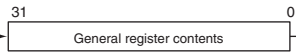
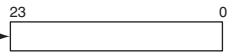
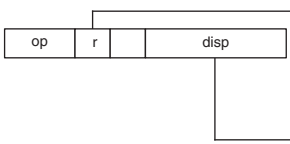
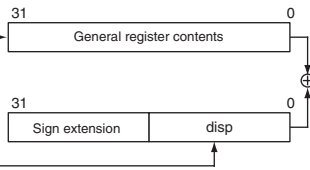
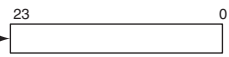
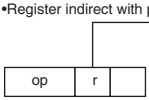
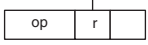
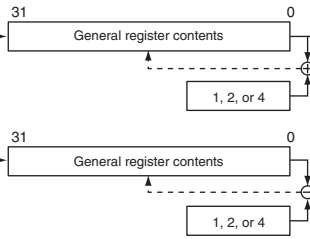
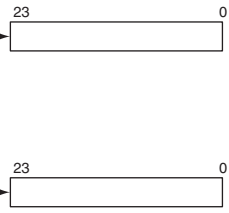



**Figure 2.8 Branch Address Specification in Memory Indirect Mode**

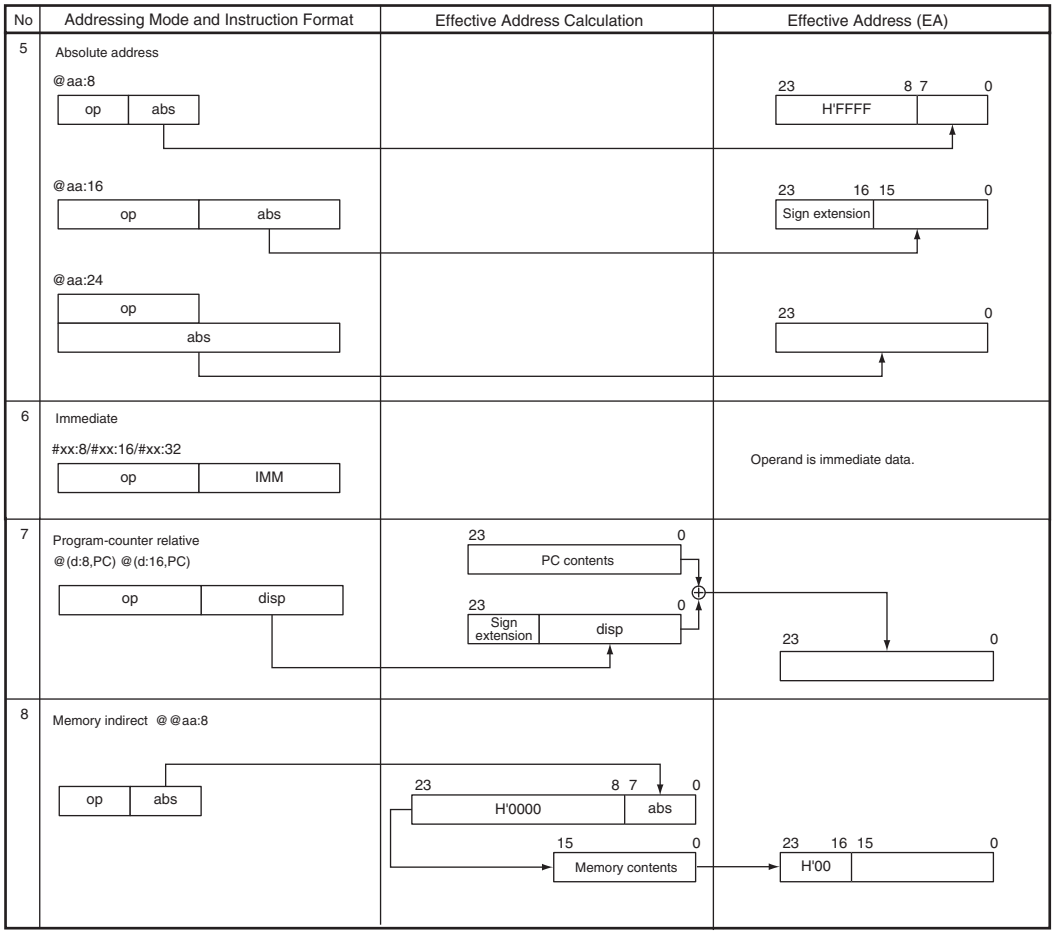
### 2.5.2 Effective Address Calculation

Table 2.12 indicates how effective addresses are calculated in each addressing mode. In this LSI, a 24-bit effective address is generated.

**Table 2.12 Effective Address Calculation (1)**

| No | Addressing Mode and Instruction Format | Effective Address Calculation | Effective Address (EA) |
|---|---|---|---|
| 1 | Register direct(Rn)<br>  | | Operand is general register contents. |
| 2 | Register indirect(@ERn)<br>  |  |  |
| 3 | Register indirect with displacement<br>@(d:16,ERn) or @(d:24,ERn)<br>  |  |  |
| 4 | Register indirect with post-increment or<br>pre-decrement<br>•Register indirect with post-increment @ERn+<br>  <br>•Register indirect with pre-decrement @-ERn<br>  |  <br>The value to be added or subtracted is 1 when the operand is byte size, 2 for word size, and 4 for longword size. |  |

## Table 2.12 Effective Address Calculation (2)




[Legend]
r, rm,rn : Register field
op : Operation field
disp : Displacement
IMM : Immediate data
abs : Absolute address

## 2.6 Basic Bus Cycle

CPU operation is synchronized by a system clock ($\phi$) or a subclock ($\phi_{SUB}$). The period from a rising edge of $\phi$ or $\phi_{SUB}$ to the next rising edge is called one state. A bus cycle consists of two states or three states. The cycle differs depending on whether access is to on-chip memory or to on-chip peripheral modules.

### 2.6.1 Access to On-Chip Memory (RAM, ROM)

Access to on-chip memory takes place in two states. The data bus width is 16 bits, allowing access in byte or word size. Figure 2.9 shows the on-chip memory access cycle.

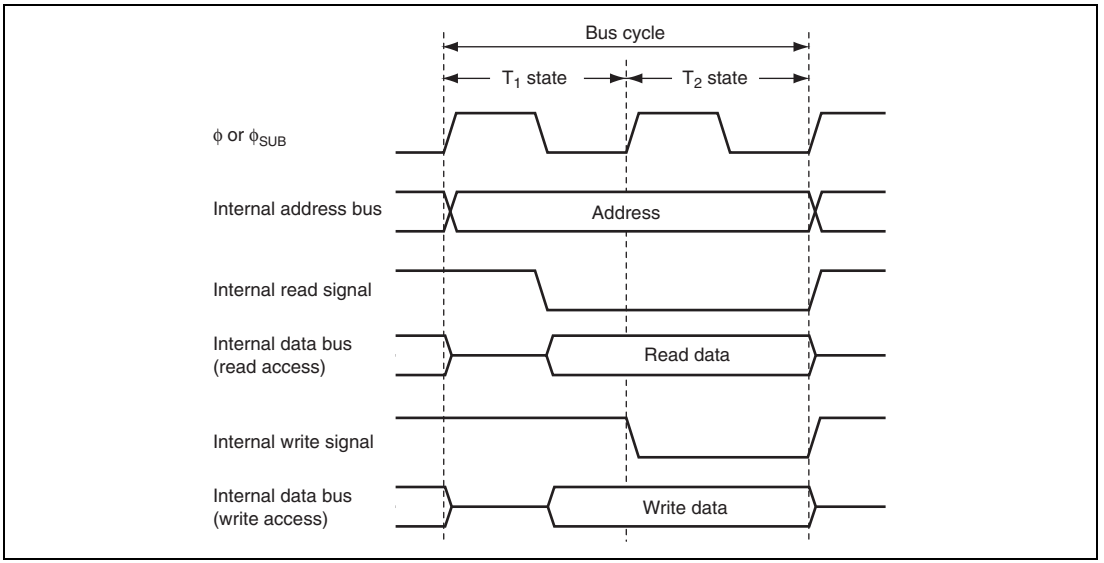



**Figure 2.9 On-Chip Memory Access Cycle**

### 2.6.2 On-Chip Peripheral Modules

On-chip peripheral modules are accessed in two to four states. The data bus width is 8 bits or 16 bits depending on the register. For details on the data bus width and number of accessing states of each register, refer to section 22, List of Registers. Registers with 16-bit data bus width can be accessed only in words. Registers with 8-bit data bus width can be accessed in bytes or words. When a register with 8-bit data bus width is accessed in words, two bus cycles for byte access are generated. In two-state access, the operation timing is the same as that for the on-chip memory. Figure 2.10 shows the operation timing in three-state access. In four-state access, a wait cycle is inserted between $T_2$ state and $T_3$ state.

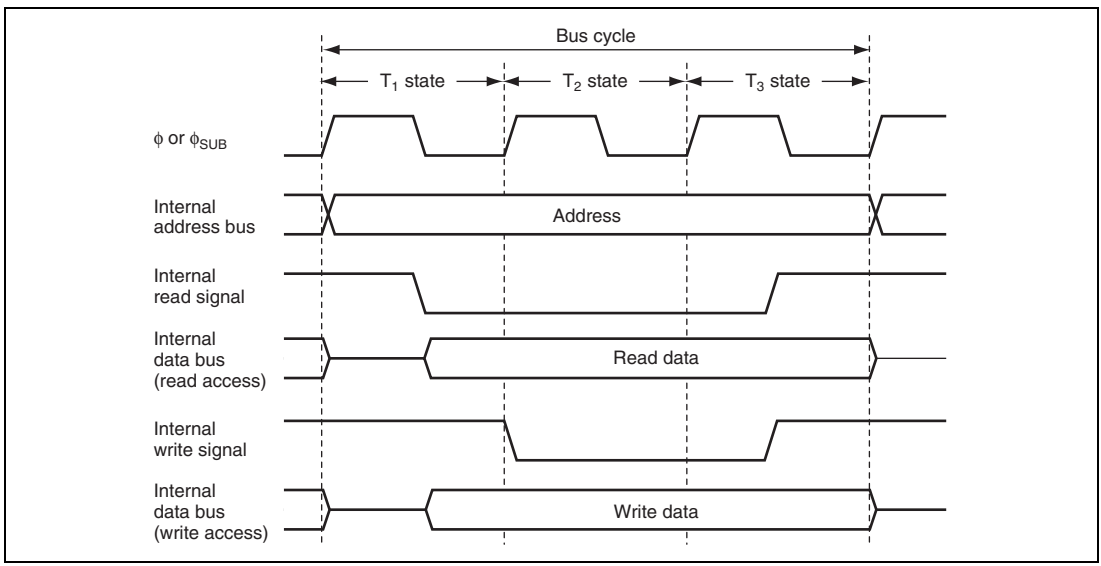



**Figure 2.10 On-Chip Peripheral Module Access Cycle (3-State Access)**

## 2.7 CPU States

There are four CPU states: the reset state, program execution state, program halt state, and exception-handling state. The program execution state includes active mode and subactive mode. For the program halt state, there are a sleep mode, standby mode, and sub-sleep mode. These states are shown in figure 2.11. Figure 2.12 shows the state transitions. For details on program execution state and program halt state, refer to section 6, Power-Down Modes. For details on exception processing, refer to section 3, Exception Handling.

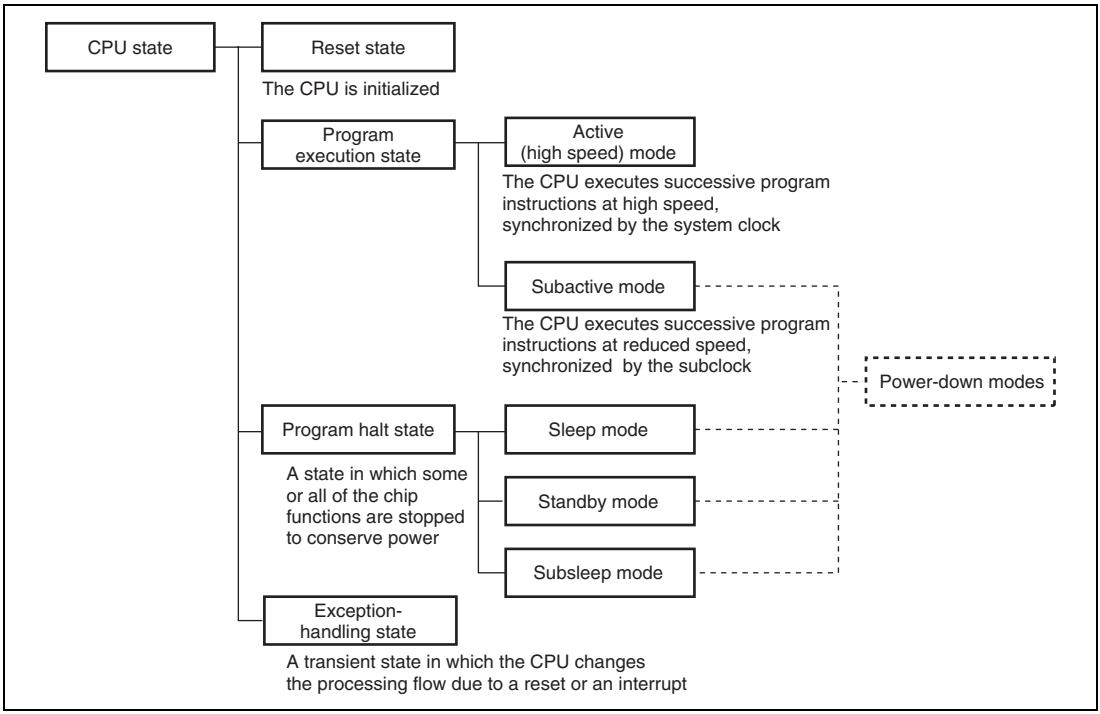



**Figure 2.11 CPU Operation States**

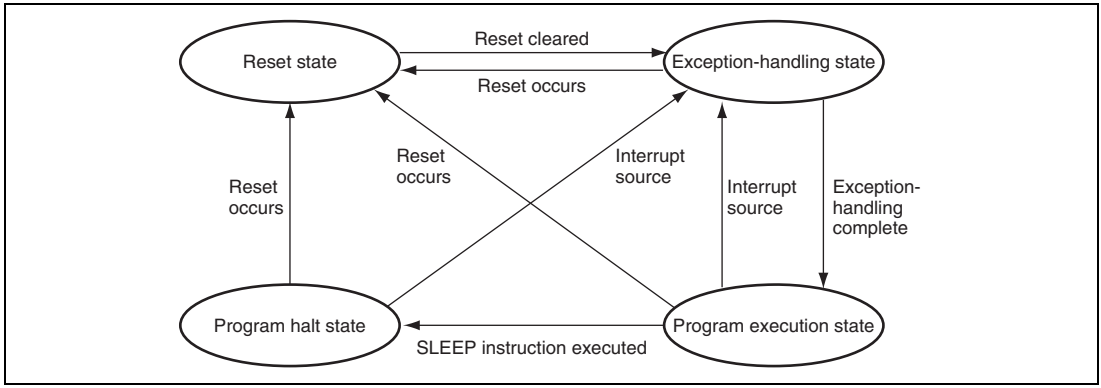



**Figure 2.12 State Transitions**

## 2.8 Usage Notes

### 2.8.1 Notes on Data Access to Empty Areas

The address space of this LSI includes empty areas in addition to the ROM, RAM, and on-chip I/O registers areas available to the user. When data is transferred from CPU to empty areas, the transferred data will be lost. This action may also cause the CPU to malfunction. When data is transferred from an empty area to CPU, the contents of the data cannot be guaranteed.

### 2.8.2 EEPMOV Instruction

EEPMOV is a block-transfer instruction and transfers the byte size of data indicated by R4 or R4L, which starts from the address indicated by ER5, to the address indicated by ER6. Set R4 or R4L and ER6 so that the end address of the destination address (value of ER6 + R4 or ER6 + R4L) does not exceed H'FFFFFF (the value of ER6 must not change from H'FFFFFF to H'000000 during execution).

### 2.8.3 Bit Manipulation Instruction

The BSET, BCLR, BNOT, BST, and BIST instructions read data from the specified address in byte units, manipulate the data of the target bit, and write data to the same address again in byte units. Special care is required when using these instructions in cases where two registers are assigned to the same address, or when a bit is directly manipulated for a port or a register containing a write-only bit, because this may rewrite data of a bit other than the bit to be manipulated.

### (1) Bit manipulation for two registers assigned to the same address

**Example 1: Bit manipulation for the timer load register and timer counter**

**(Applicable for timer B1 in the H8/36109 Group.)**

Figure 2.13 shows an example of a timer in which two timer registers are assigned to the same address. When a bit-manipulation instruction accesses the timer load register and timer counter of a reloadable timer, since these two registers share the same address, the following operations takes place.

1. Data is read in byte units.
2. The CPU sets or resets the bit to be manipulated with the bit-manipulation instruction.
3. The written data is written again in byte units to the timer load register.

The timer is counting, so the value read is not necessarily the same as the value in the timer load register. As a result, bits other than the intended bit in the timer counter may be modified and the modified value may be written to the timer load register.

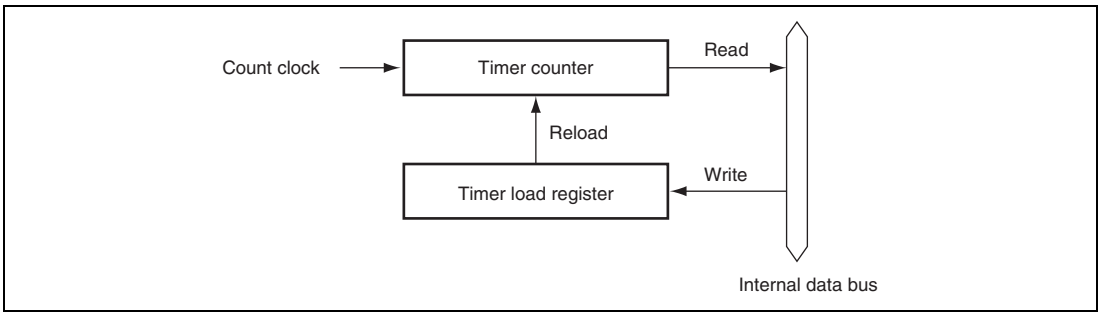



**Figure 2.13 Example of Timer Configuration with Two Registers Allocated to Same Address**

**Example 2: The BSET instruction is executed for port 5.**

P57 and P56 are input pins, with a low-level signal input at P57 and a high-level signal input at P56. P55 to P50 are output pins and output low-level signals. An example to output a high-level signal at P50 with a BSET instruction is shown below.

- Prior to executing BSET instruction

| | **P57** | **P56** | **P55** | **P54** | **P53** | **P52** | **P51** | **P50** |
|---|---|---|---|---|---|---|---|---|
| Input/output | Input | Input | Output | Output | Output | Output | Output | Output |
| Pin state | Low level | High level | Low level | Low level | Low level | Low level | Low level | Low level |
| PCR5 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| PDR5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- BSET instruction executed instruction

```
BSET    #0,     @PDR5
```

The BSET instruction is executed for port 5.

- After executing BSET instruction

| | **P57** | **P56** | **P55** | **P54** | **P53** | **P52** | **P51** | **P50** |
|---|---|---|---|---|---|---|---|---|
| Input/output | Input | Input | Output | Output | Output | Output | Output | Output |
| Pin state | Low level | High level | Low level | Low level | Low level | Low level | Low level | High level |
| PCR5 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| PDR5 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

- Description on operation

1. When the BSET instruction is executed, first the CPU reads port 5.
   Since P57 and P56 are input pins, the CPU reads the pin states (low-level and high-level input).
   P55 to P50 are output pins, so the CPU reads the value in PDR5. In this example PDR5 has a value of H'80, but the value read by the CPU is H'40.
2. Next, the CPU sets bit 0 of the read data to 1, changing the PDR5 data to H'41.
3. Finally, the CPU writes H'41 to PDR5, completing execution of BSET instruction.

As a result of the BSET instruction, bit 0 in PDR5 becomes 1, and P50 outputs a high-level signal. However, bits 7 and 6 of PDR5 end up with different values. To prevent this problem, store a copy of the PDR5 data in a work area in memory. Perform the bit manipulation on the data in the work area, then write this data to PDR5.

- Prior to executing BSET instruction

```
MOV.B   #80,  R0L
MOV.B   R0L,  @RAM0
MOV.B   R0L,  @PDR5
```

The PDR5 value (H'80) is written to a work area in memory (RAM0) as well as to PDR5.

| | **P57** | **P56** | **P55** | **P54** | **P53** | **P52** | **P51** | **P50** |
|---|---|---|---|---|---|---|---|---|
| Input/output | Input | Input | Output | Output | Output | Output | Output | Output |
| Pin state | Low level | High level | Low level | Low level | Low level | Low level | Low level | Low level |
| PCR5 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| PDR5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RAM0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- BSET instruction executed

```
BSET    #0,     @RAM0
```

The BSET instruction is executed designating the PDR5 work area (RAM0).

- After executing BSET instruction

```
MOV.B   @RAM0, R0L
MOV.B   R0L,  @PDR5
```

The work area (RAM0) value is written to PDR5.

| | **P57** | **P56** | **P55** | **P54** | **P53** | **P52** | **P51** | **P50** |
|---|---|---|---|---|---|---|---|---|
| Input/output | Input | Input | Output | Output | Output | Output | Output | Output |
| Pin state | Low level | High level | Low level | Low level | Low level | Low level | Low level | High level |
| PCR5 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| PDR5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| RAM0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

### (2) Bit Manipulation in a Register Containing a Write-Only Bit

**Example 3: BCLR instruction executed designating port 5 control register PCR5**

P57 and P56 are input pins, with a low-level signal input at P57 and a high-level signal input at P56. P55 to P50 are output pins that output low-level signals. An example of setting the P50 pin as an input pin by the BCLR instruction is shown below. It is assumed that a high-level signal will be input to this input pin.

- Prior to executing BCLR instruction

| | **P57** | **P56** | **P55** | **P54** | **P53** | **P52** | **P51** | **P50** |
|---|---|---|---|---|---|---|---|---|
| Input/output | Input | Input | Output | Output | Output | Output | Output | Output |
| Pin state | Low level | High level | Low level | Low level | Low level | Low level | Low level | Low level |
| PCR5 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| PDR5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- BCLR instruction executed

```
BCLR    #0,     @PCR5
```

The BCLR instruction is executed for PCR5.

- After executing BCLR instruction

| | **P57** | **P56** | **P55** | **P54** | **P53** | **P52** | **P51** | **P50** |
|---|---|---|---|---|---|---|---|---|
| Input/output | Output | Output | Output | Output | Output | Output | Output | Input |
| Pin state | Low level | High level | Low level | Low level | Low level | Low level | Low level | High level |
| PCR5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| PDR5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- Description on operation

1. When the BCLR instruction is executed, first the CPU reads PCR5. Since PCR5 is a write-only register, the CPU reads a value of H'FF, even though the PCR5 value is actually H'3F.
2. Next, the CPU clears bit 0 in the read data to 0, changing the data to H'FE.
3. Finally, H'FE is written to PCR5 and BCLR instruction execution ends.

As a result of this operation, bit 0 in PCR5 becomes 0, making P50 an input port. However, bits 7 and 6 in PCR5 change to 1, so that P57 and P56 change from input pins to output pins. To prevent this problem, store a copy of the PCR5 data in a work area in memory and manipulate data of the bit in the work area, then write this data to PCR5.

- Prior to executing BCLR instruction

```
MOV.B   #3F,  R0L
MOV.B   R0L,  @RAM0
MOV.B   R0L,  @PCR5
```

The PCR5 value (H'3F) is written to a work area in memory (RAM0) as well as to PCR5.

| | **P57** | **P56** | **P55** | **P54** | **P53** | **P52** | **P51** | **P50** |
|---|---|---|---|---|---|---|---|---|
| Input/output | Input | Input | Output | Output | Output | Output | Output | Output |
| Pin state | Low level | High level | Low level | Low level | Low level | Low level | Low level | Low level |
| PCR5 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| PDR5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RAM0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

- BCLR instruction executed

```
BCLR   #0,    @RAM0
```

The BCLR instructions executed for the PCR5 work area (RAM0).

- After executing BCLR instruction

```
MOV.B   @RAM0, R0L
MOV.B   R0L,  @PCR5
```

The work area (RAM0) value is written to PCR5.

| | **P57** | **P56** | **P55** | **P54** | **P53** | **P52** | **P51** | **P50** |
|---|---|---|---|---|---|---|---|---|
| Input/output | Input | Input | Output | Output | Output | Output | Output | Output |
| Pin state | Low level | High level | Low level | Low level | Low level | Low level | Low level | High level |
| PCR5 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| PDR5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| RAM0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |

# Section 3 Exception Handling

Exception handling is caused by a reset, a trap instruction (TRAPA), or interrupts.

- Reset

  A reset has the highest exception priority. Exception handling starts after the reset state is cleared by a negation of the $\overline{\text{RES}}$ signal. Exception handling is also started when the watchdog timer overflows. The exception handling executed at this time is the same as that for a reset by the $\overline{\text{RES}}$ pin.
- Trap Instruction

  Exception handling starts when a trap instruction (TRAPA) is executed. A vector address corresponding to a vector number from 0 to 3 which are specified in the instruction code is generated. Exception handling can be executed at all times in the program execution state, regardless of the setting of the I bit in CCR.
- Interrupts

  External interrupts other than the NMI and internal interrupts other than the address break are masked by the I bit in CCR, and kept pending while the I bit is set to 1. Exception handling starts when the current instruction or exception handling ends, if an interrupt is requested.
- Priority level

  The priority levels of interrupt sources other than the NMI and address break can be set for each module by the interrupt control register (ICR).

## 3.1 Exception Sources and Vector Address

Table 3.1 shows the vector addresses and priority of each exception handling. When more than one interrupt is requested, handling is performed from the interrupt with the highest priority. The priority level can be set for an interrupt source to which a bit in ICR is assigned. When priority level 1 (priority is given) is set for an interrupt source other than the NMI and address break, the execution of the exception handling for the interrupt request has priority that for an interrupt request whose source is set to priority level 0.

## Table 3.1 Exception Sources and Vector Address

| Related Module | Exception Sources | Vector Number | Vector Address | ICR | Priority |
|---|---|---|---|---|---|
| $\overline{\text{RES}}$ pin<br>Watchdog timer | Reset | 0 | H'000000 to H'000003 | — | High |
| — | Reserved for system use | 1 to 6 | H'000004 to H'00001B | — | |
| External interrupt pin | NMI | 7 | H'00001C to H'00001F | — | |
| CPU | Trap instruction #0 | 8 | H'000020 to H'000023 | — | |
| | Trap instruction #1 | 9 | H'000024 to H'000027 | — | |
| | Trap instruction #2 | 10 | H'000028 to H'00002B | — | |
| | Trap instruction #3 | 11 | H'00002C to H'00002F | — | |
| Address break | Break conditions satisfied | 12 | H'000030 to H'000033 | — | |
| CPU | Direct transition by executing the SLEEP instruction | 13 | H'000034 to H'000037 | ICRA7 | |
| External interrupt pin | IRQ0<br>Low-voltage detection interrupt* | 14 | H'000038 to H'00003B | ICRA6 | |
| | IRQ1 | 15 | H'00003C to H'00003F | ICRA5 | |
| | IRQ2 | 16 | H'000040 to H'000043 | ICRA4 | |
| | IRQ3 | 17 | H'000044 to H'000047 | ICRA3 | |
| | WKP | 18 | H'000048 to H'00004B | ICRA2 | |
| RTC | Overflow | 19 | H'00004C to H'00004F | ICRA1 | |
| — | Reserved for system use | 20, 21 | H'000050 to H'000053 | — | |
| Timer V | Compare match A<br>Compare match B<br>Overflow | 22 | H'000058 to H'00005B | ICRB6 | |
| SCI3 | Receive data full<br>Transmit data empty<br>Transmit end<br>Receive error | 23 | H'00005C to H'00005F | ICRB5 | |
| IIC2 | Transmit data empty<br>Transmit end<br>Receive data full<br>Arbitration lost/overrun error<br>NACK detection<br>Stop condition detected | 24 | H'000060 to H'000063 | ICRB4 | Low |

| Related Module | Exception Sources | Vector Number | Vector Address | ICR | Priority |
|---|---|---|---|---|---|
| — | Reserved for system use | 25 to 28 | H'000064 to H'000073 | — | High |
| Timer B1 | Overflow | 29 | H'000074 to H'000077 | ICRC7 | |
| — | Reserved for system use | 30, 31 | H'000078 to H'00007F | — | |
| SCI3_2 | Receive data full<br>Transmit data empty<br>Transmit end<br>Receive error | 32 | H'000080 to H'000083 | ICRC4 | |
| — | Reserved for system use | 33 | H'000084 to H'000087 | — | |
| SCI3_3 | Receive data full<br>Transmit data empty<br>Transmit end<br>Receive error | 34 | H'000088 to H'00008B | ICRC2 | |
| Timer RC | Input capture A/compare match A<br>Input capture B/compare match B<br>Input capture C/compare match C<br>Input capture D/compare match D<br>Overflow | 35 | H'00008C to H'00008F | ICRC1 | |
| A/D converter | A/D conversion end | 36 | H'000090 to H'000093 | ICRC0 | |
| Timer RD_0 | Compare match/input capture A0 to D0<br>Overflow | 37 | H'000094 to H'000097 | ICRD7 | |
| Timer RD_1 | Compare match/input capture A1 to D1<br>Overflow | 38 | H'000098 to H'00009B | ICRD6 | |
| Timer RD_2 | Compare match/input capture A2 to D2<br>Overflow | 39 | H'00009C to H'00009F | ICRD5 | |
| Timer RD_3 | Compare match/input capture A3 to D3<br>Overflow | 40 | H'0000A0 to H'0000A3 | ICRD4 | |
| Clock switching | When the system clock sources are switched from the external-input signal to the internal-generated signal | 41 | H'0000A4 to H'0000A7 | ICRD3 | Low |

Note: * A low-voltage detection interrupt is available only in the product with an on-chip power-on reset and low-voltage detection circuit.

## 3.2 Register Descriptions

Interrupts are controlled by the following registers.

- Interrupt edge select register 1 (IEGR1)
- Interrupt edge select register 2 (IEGR2)
- Interrupt enable register 1 (IENR1)
- Interrupt enable register 2 (IENR2)
- Interrupt flag register 1 (IRR1)
- Interrupt flag register 2 (IRR2)
- Wakeup interrupt flag register (IWPR)
- Interrupt control registers A to D (ICRA to ICRD)

### 3.2.1 Interrupt Edge Select Register 1 (IEGR1)

IEGR1 selects the direction of an edge that generates interrupt requests of pins $\overline{\text{NMI}}$ and $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | NMIEG | 0 | R/W | NMI Edge Select<br>0: Falling edge of $\overline{\text{NMI}}$ pin input is detected<br>1: Rising edge of $\overline{\text{NMI}}$ pin input is detected |
| 6 to 4 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 3 | IEG3 | 0 | R/W | IRQ3 Edge Select<br>0: Falling edge of $\overline{\text{IRQ3}}$ pin input is detected<br>1: Rising edge of $\overline{\text{IRQ3}}$ pin input is detected |
| 2 | IEG2 | 0 | R/W | IRQ2 Edge Select<br>0: Falling edge of $\overline{\text{IRQ2}}$ pin input is detected<br>1: Rising edge of $\overline{\text{IRQ2}}$ pin input is detected |
| 1 | IEG1 | 0 | R/W | IRQ1 Edge Select<br>0: Falling edge of $\overline{\text{IRQ1}}$ pin input is detected<br>1: Rising edge of $\overline{\text{IRQ1}}$ pin input is detected |
| 0 | IEG0 | 0 | R/W | IRQ0 Edge Select<br>0: Falling edge of $\overline{\text{IRQ0}}$ pin input is detected<br>1: Rising edge of $\overline{\text{IRQ0}}$ pin input is detected |

### 3.2.2 Interrupt Edge Select Register 2 (IEGR2)

IEGR2 selects the direction of an edge that generates interrupt requests of pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 5 | WPEG5 | 0 | R/W | WKP5 Edge Select<br>0: Falling edge of $\overline{\text{WKP5}}$ pin input is detected<br>1: Rising edge of $\overline{\text{WKP5}}$ pin input is detected |
| 4 | WPEG4 | 0 | R/W | WKP4 Edge Select<br>0: Falling edge of $\overline{\text{WKP4}}$ pin input is detected<br>1: Rising edge of $\overline{\text{WKP4}}$ pin input is detected |
| 3 | WPEG3 | 0 | R/W | WKP3 Edge Select<br>0: Falling edge of $\overline{\text{WKP3}}$ pin input is detected<br>1: Rising edge of $\overline{\text{WKP3}}$ pin input is detected |
| 2 | WPEG2 | 0 | R/W | WKP2 Edge Select<br>0: Falling edge of $\overline{\text{WKP2}}$ pin input is detected<br>1: Rising edge of $\overline{\text{WKP2}}$ pin input is detected |
| 1 | WPEG1 | 0 | R/W | WKP1Edge Select<br>0: Falling edge of $\overline{\text{WKP1}}$ pin input is detected<br>1: Rising edge of $\overline{\text{WKP1}}$ pin input is detected |
| 0 | WPEG0 | 0 | R/W | WKP0 Edge Select<br>0: Falling edge of $\overline{\text{WKP0}}$ pin input is detected<br>1: Rising edge of $\overline{\text{WKP0}}$ pin input is detected |

### 3.2.3 Interrupt Enable Register 1 (IENR1)

IENR1 enables direct transition interrupts, RTC interrupts, and external pin interrupts.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | IENDT | 0 | R/W | Direct Transition Interrupt Enable<br>When this bit is set to 1, direct transition interrupt requests are enabled. |
| 6 | IENTA | 0 | R/W | RTC Interrupt Enable<br>When this bit is set to 1, an RTC interrupt request is enabled. |
| 5 | IENWP | 0 | R/W | Wakeup Interrupt Enable<br>This bit is an enable bit for signals $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$. When this bit is set to 1, an interrupt request is enabled. |
| 4 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 3 | IEN3 | 0 | R/W | IRQ3 Interrupt Enable<br>When this bit is set to 1, an interrupt request of the $\overline{\text{IRQ3}}$ signal is enabled. |
| 2 | IEN2 | 0 | R/W | IRQ2 Interrupt Enable<br>When this bit is set to 1, an interrupt request of the $\overline{\text{IRQ2}}$ signal is enabled. |
| 1 | IEN1 | 0 | R/W | IRQ1 Interrupt Enable<br>When this bit is set to 1, an interrupt request of the $\overline{\text{IRQ1}}$ signal is enabled. |
| 0 | IEN0 | 0 | R/W | IRQ0 Interrupt Enable<br>When this bit is set to 1, an interrupt request of the $\overline{\text{IRQ0}}$ signal is enabled. |

A bit in an interrupt enable register to disable the interrupt or a bit in an interrupt flag register must be cleared while the interrupt is masked (I = 1). If the execution of clearing the above bit and an interrupt request occurs at the same time while I = 0, the exception handling for the interrupt is executed after the bit has been cleared.

### 3.2.4 Interrupt Enable Register 2 (IENR2)

IENR2 enables a timer B1 overflow interrupt.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | — | All 0 | — | Reserved<br>These bits are always read as 0. |
| 5 | IENTB1 | 0 | R/W | Timer B1 Interrupt Enable<br>When this bit is set to 1, a timer B1 overflow interrupt request is enabled. |
| 4 to 0 | — | All 1 | — | Reserved<br>These bits are always read as 1. |

A bit in an interrupt enable register to disable the interrupt or a bit in an interrupt flag register must be cleared while the interrupt is masked (I = 1). If the execution of clearing the above bit and an interrupt request occurs at the same time while I = 0, the exception handling for the interrupt is executed after the bit has been cleared.

### 3.2.5 Interrupt Flag Register 1 (IRR1)

IRR1 is a status flag register for a direct transition interrupt, an RTC interrupt, and $\overline{IRQ3}$ to $\overline{IRQ0}$ interrupts.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | IRRDT | 0 | R/W | Direct Transition Interrupt Request Flag<br>[Setting condition]<br>When a direct transition is made by executing the SLEEP instruction while the DTON bit in SYSCR2 is set to 1.<br>[Clearing condition]<br>When writing 0 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | IRRTA | — | R/W | RTC Interrupt Request Flag<br>[Setting condition]<br>When the RTC counter value overflows<br>[Clearing condition]<br>When writing 0 |
| 5, 4 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 3 | IRRI3 | 0 | R/W | IRQ3 Interrupt Request Flag<br>[Setting condition]<br>When the $\overline{\text{IRQ3}}$ pin is specified as an interrupt input and the specified edge is detected.<br>[Clearing condition]<br>When writing 0 |
| 2 | IRRI2 | 0 | R/W | IRQ2 Interrupt Request Flag<br>[Setting condition]<br>When the $\overline{\text{IRQ2}}$ pin is specified as an interrupt input and the specified edge is detected.<br>[Clearing condition]<br>When writing 0 |
| 1 | IRRI1 | 0 | R/W | IRQ1 Interrupt Request Flag<br>[Setting condition]<br>When the $\overline{\text{IRQ1}}$ pin is specified as an interrupt input and the specified edge is detected.<br>[Clearing condition]<br>When writing 0 |
| 0 | IRRI0 | 0 | R/W | IRQ0 Interrupt Request Flag<br>[Setting condition]<br>When the $\overline{\text{IRQ0}}$ pin is specified as an interrupt input and the specified edge is detected.<br>[Clearing condition]<br>When writing 0 |

### 3.2.6 Interrupt Flag Register 2 (IRR2)

IRR2 is a status flag register for timer B1 overflow interrupts.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | — | All 0 | — | Reserved<br>These bits are always read as 0. |
| 5 | IRRTB1 | 0 | R/W | Timer B1 Interrupt Request flag<br>[Setting condition]<br>When the timer B1 counter overflows<br>[Clearing condition]<br>When writing 0 |
| 4 to 0 | — | All 1 | — | Reserved<br>These bits are always read as 1. |

### 3.2.7 Wakeup Interrupt Flag Register (IWPR)

IWPR is a status flag register for $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ interrupt requests.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 5 | IWPF5 | 0 | R/W | WKP5 Interrupt Request Flag<br>[Setting condition]<br>When the $\overline{\text{WKP5}}$ pin is specified as an interrupt input and the specified edge is detected<br>[Clearing condition]<br>When writing 0 |
| 4 | IWPF4 | 0 | R/W | WKP4 Interrupt Request Flag<br>[Setting condition]<br>When $\overline{\text{WKP4}}$ pin is specified as an interrupt input and the specified edge is detected<br>[Clearing condition]<br>When writing 0 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | IWPF3 | 0 | R/W | WKP3 Interrupt Request Flag<br>[Setting condition]<br>When the $\overline{\text{WKP3}}$ pin is specified as an interrupt input and the specified edge is detected<br>[Clearing condition]<br>When writing 0 |
| 2 | IWPF2 | 0 | R/W | WKP2 Interrupt Request Flag<br>[Setting condition]<br>When the $\overline{\text{WKP2}}$ pin is specified as an interrupt input and the specified edge is detected.<br>[Clearing condition]<br>When writing 0 |
| 1 | IWPF1 | 0 | R/W | WKP1 Interrupt Request Flag<br>[Setting condition]<br>When the $\overline{\text{WKP1}}$ pin is specified as an interrupt input and the specified as an edge is detected<br>[Clearing condition]<br>When writing 0. |
| 0 | IWPF0 | 0 | R/W | WKP0 Interrupt Request Flag<br>[Setting condition]<br>When the $\overline{\text{WKP0}}$ pin is specified as an interrupt input and the specified edge is detected<br>[Clearing condition]<br>When writing 0 |

### 3.2.8 Interrupt Control Registers A to D (ICRA to ICRD)

ICR sets the priority level of an interrupt source other than the NMI and address break. The correspondence between interrupt requests and bits ICRA to ICRD is shown in table 3.2.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 0 | ICRn7 to ICRn0 | All 0* | R/W | Interrupt Priority Level<br>0: The corresponding interrupt source is set to interrupt priority level 0 (nonpriority)<br>1: The corresponding interrupt source is set to interrupt priority level 1 (priority) |

n = A to D

Note: * The initial values of the reserved bits are also all 0.

**Table 3.2 Interrupt request and ICR**

| | | Registers | | | |
|---|---|---|---|---|---|
| **Bit** | **Bit Name** | **ICRA** | **ICRB** | **ICRC** | **ICRD** |
| 7 | ICRn7 | Direct transition | — | Timer B1 | Timer RD_0 |
| 6 | ICRn6 | IRQ0, Low-voltage detection | Timer V | — | Timer RD_1 |
| 5 | ICRn5 | IRQ1 | SCI3 | — | Timer RD_2 |
| 4 | ICRn4 | IRQ2 | IIC2 | SCI3_2 | Timer RD_3 |
| 3 | ICRn3 | IRQ3 | — | — | Clock switching |
| 2 | ICRn2 | WKP | — | SCI3_3 | — |
| 1 | ICRn1 | RTC | — | Timer RC | — |
| 0 | ICRn0 | — | — | A/D converter | — |

n = A to D

—: Reserved. These bits are always read as 0.

## 3.3 Reset Exception Handling

When the $\overline{\text{RES}}$ signal goes low, all processing halts and this LSI enters the reset state. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized by the reset. When the power is turned on, hold the $\overline{\text{RES}}$ signal low until oscillation of the clock pulse generator settles to ensure that this LSI is reset. To reset this LSI during operation, hold the $\overline{\text{RES}}$ signal low for a given time. When the $\overline{\text{RES}}$ signal goes high after being held low for the given time, this LSI starts the reset exception handling. The reset exception handling sequence is shown in figure 3.1. However, for the reset exception handling sequence of the product with an on-chip power-on reset circuit, refer to section 20, Band-Gap Regulator, Power-On Reset (Optional), and Low-Voltage Detection Circuits (Optional).

The reset exception handling sequence is as follows:

1. Set the I bit in the condition code register (CCR) to 1.
2. The CPU generates the vector address for the reset exception handling (from H'000000 to H'000003), the data in the address is sent to the program counter (PC) as the start address, and program execution starts from the address.

## 3.4 Interrupt Exception Handling

### 3.4.1 External Interrupts

As the external interrupts, there are the NMI, IRQ3 to IRQ0, and WKP5 to WKP0 interrupts.

- NMI Interrupt

  An NMI interrupt is generated when the edge of the $\overline{\text{NMI}}$ signal is input. The detecting edge is selected from rising or falling, depending on the setting of the NMIEG bit in IEGR1.

  Since the NMI interrupt is given the highest priority level, it can always be accepted regardless of the setting of the I bit in CCR.
- IRQ3 to IRQ0 Interrupts

  IRQ3 to IRQ0 interrupts are generated when the edges of the $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ signals are input. These four interrupts are given different vector addresses, and the detecting edge of each signal can be selected from rising or falling, depending on the settings of bits IEG3 to IEG0 in IEGR1.

  When the $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$ pins are specified as an interrupt input by PMR1 and the specified edge is input, the corresponding bit in IRR1 is set to 1, requesting the interrupt to the CPU. These interrupts can be masked by setting bits IEN3 to IEN0 in IENR1.

- WKP5 to WKP0 Interrupts

  WKP5 to WKP0 interrupts are generated when the edges of the $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ signals are input. These six interrupts are assigned to the same vector addresses, and the detecting edge for each signal can be selected from rising or falling, depending on the settings of bits WPEG5 to WPEG0 in IEGR2.

  When pins $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$ are specified as an interrupt input by PMR5 and the specified edge is input, the corresponding bit in IWPR is set to 1, requesting an interrupt to the CPU. These interrupts can be masked by setting bit IENWP in IENR1.

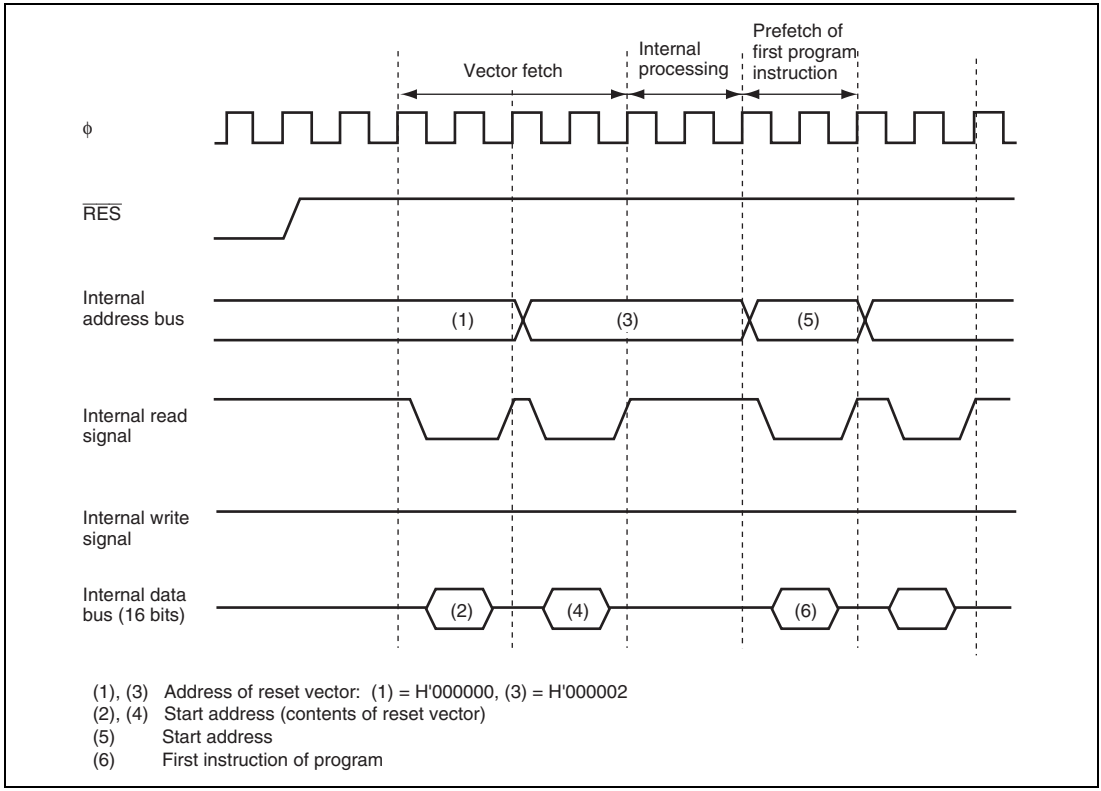



**Figure 3.1 Reset Sequence**

### 3.4.2 Internal Interrupts

Each on-chip peripheral module has a flag to indicate the interrupt request status and the enable bit to enable or disable the interrupt. For RTC interrupt requests and direct transition interrupt requests generated by execution of the SLEEP instruction, this function is included in IRR1, IRR2, IENR1, and IENR2.

When an on-chip peripheral module requests an interrupt, the corresponding interrupt request status flag is set to 1, requesting an interrupt to the CPU. These interrupts can be disabled by clearing the corresponding enable bit to 0.

### 3.4.3 Interrupt Handling Sequence

Interrupts are controlled by an interrupt controller.

Interrupt operation is described below.

1. If an NMI or an interrupt with its enable bit set to 1 is generated, an interrupt request signal is sent to the interrupt controller.
2. When multiple interrupt requests are generated, the interrupt controller requests the interrupt handling with the highest priority level which has been set in ICR to the CPU. Other interrupt requests are held pending. When the priority levels are the same, the interrupt controller selects an interrupt request according to the default priority levels shown in table 3.1.
3. The CPU accepts the NMI and address break regardless of the setting of the I bit. Other interrupt requests are accepted, if the I bit is cleared to 0 in CCR; if the I bit is set to 1, the interrupt request is held pending.
4. If the CPU accepts the interrupt after execution of the current instruction is completed, interrupt exception handling will begin. First, both PC and CCR are pushed onto the stack. The stack status at this time is shown in figure 3.3. The PC value pushed onto the stack is the address of the first instruction to be executed upon return from interrupt handling.
5. Then, the I bit in CCR is set to 1, masking further interrupts excluding the NMI and address break. Upon return from interrupt handling, the values of I bit and other bits in CCR will be restored and returned to the values prior to the start of interrupt exception handling.
6. Next, the CPU generates the vector address corresponding to the accepted interrupt, and transfers the address to PC as a start address of the interrupt handler. Then a program starts executing from the address indicated in PC.

Figure 3.2 shows the interrupt acceptance flowchart. Figure 3.4 shows a typical interrupt sequence where the program area is in the on-chip ROM and the stack area is in the on-chip RAM.

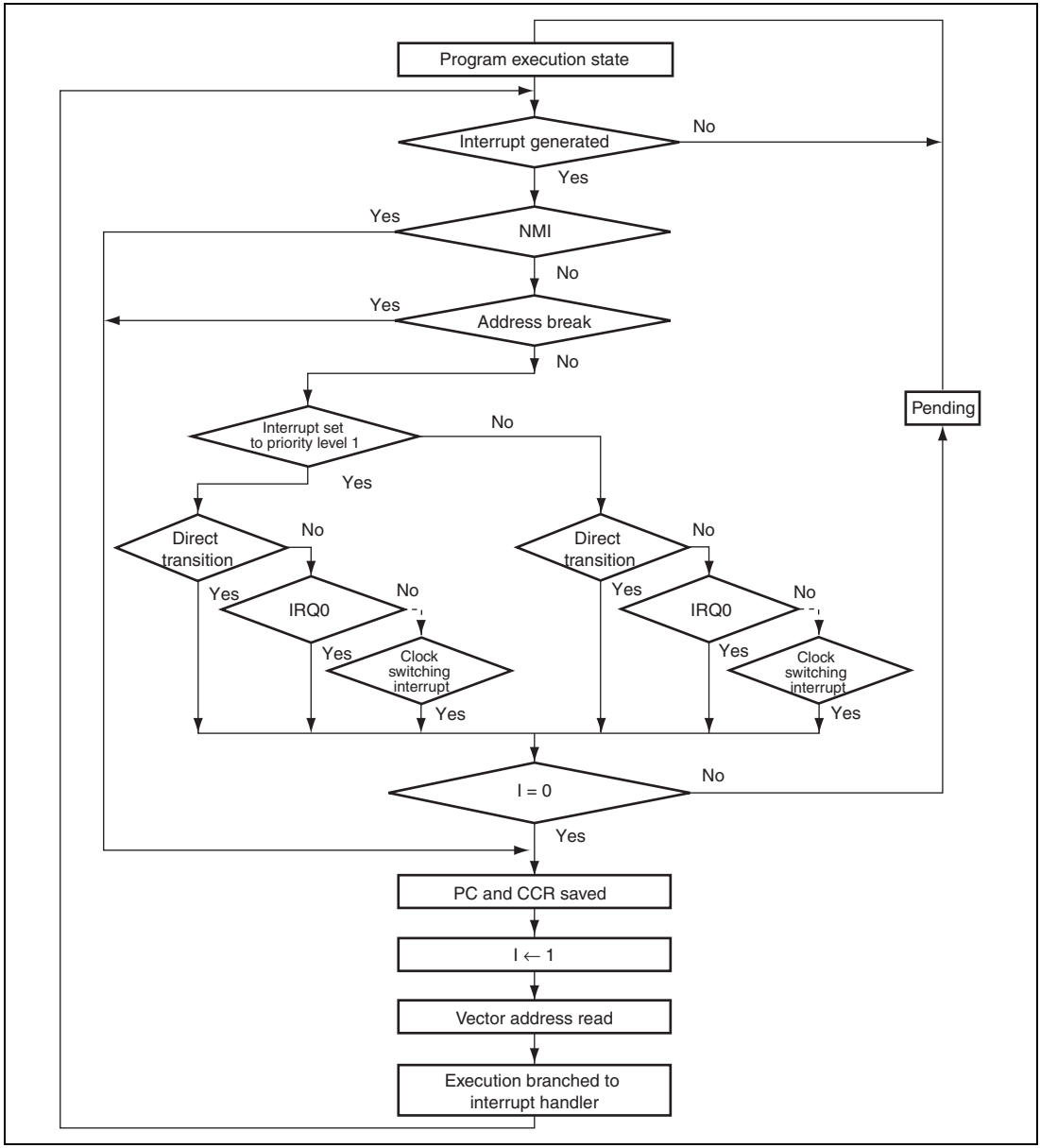
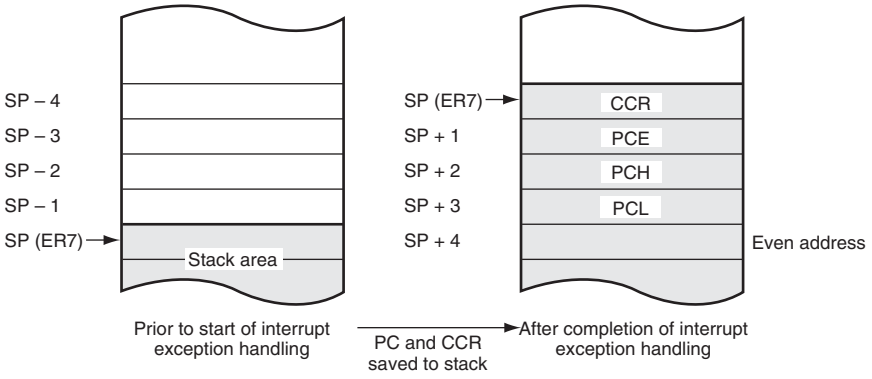



**Figure 3.2 Interrupt Acceptance Flowchart**




[Legend]
PCE: Bits 23 to 16 of program counter (PC)
PCH: Bits 15 to 8 of program counter (PC)
PCL: Bits 7 to 0 of program counter (PC)
CCR: Condition code register
SP: Stack pointer

Notes: 1. PC shows the address of the first instruction to be executed upon return from the interrupt handling routine.
2. Register contents must always be saved and restored by word length, starting from an even-numbered address.

**Figure 3.3 Stack Status after Exception Handling**

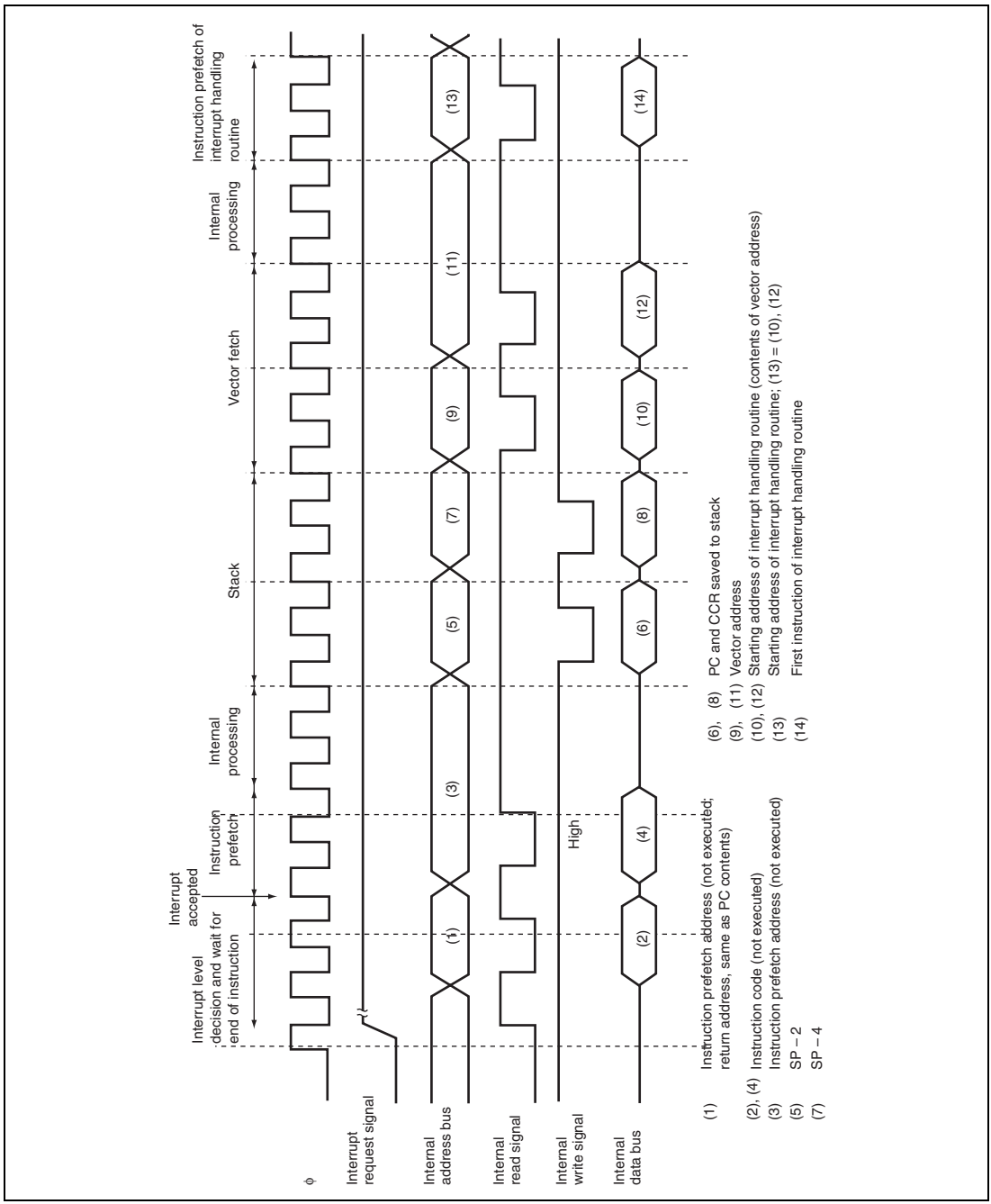



**Figure 3.4 Interrupt Sequence**

### 3.4.4 Interrupt Response Time

Table 3.3 shows the number of wait states after an interrupt request flag is set until the first instruction of the interrupt handling-routine is executed.

**Table 3.3 Interrupt Wait States**

| Item | States | Total |
|---|---|---|
| Interrupt priority determination | 2*[1] | 19 to 41 |
| Waiting time for completion of executing instruction*[2] | 1 to 23 | |
| Saving of PC and CCR to stack | 4 | |
| Vector fetch | 4 | |
| Instruction fetch | 4 | |
| Internal processing | 4 | |

Notes: 1. For internal interrupts, the number of states is 1.
2. Not including EEPMOV instruction.

## 3.5 Usage Notes

### 3.5.1 Interrupts after Reset

If an interrupt is accepted after a reset and before the stack pointer (SP) is initialized, the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset. Since the first instruction of a program is always executed immediately after the reset state ends, make sure that this instruction initializes the stack pointer (example: MOV.L #xx: 32, SP).

### 3.5.2 Notes on Stack Area Use

When word data is accessed, the least significant bit of the address is regarded as 0. Access to the stack always takes place in words, so the stack pointer (SP: ER7) should never indicate an odd address. Use PUSH Rn (MOV.W Rn, @–SP) or POP Rn (MOV.W @SP+, Rn) to save or restore register values.

### 3.5.3 Notes on Rewriting Port Mode Registers

When a port mode register is rewritten to switch the functions of external interrupt pins, $\overline{\text{IRQ3}}$ to $\overline{\text{IRQ0}}$, and $\overline{\text{WKP5}}$ to $\overline{\text{WKP0}}$, the interrupt request flag may be set to 1.

When switching pin functions, mask the interrupt before setting the bit in the port mode register. After accessing the port mode register, execute at least one instruction (e.g., NOP), then clear the interrupt request flag from 1 to 0.

Figure 3.5 shows a port mode register setting and interrupt request flag clearing procedure.

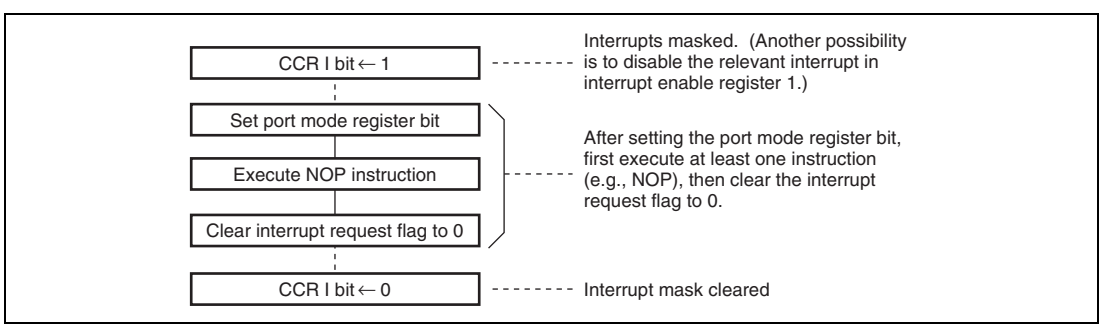



**Figure 3.5 Port Mode Register Setting and Interrupt Request Flag Clearing Procedure**

# Section 4 Address Break

The address break simplifies on-board program debugging. It requests an address break interrupt when the set break condition is satisfied. The interrupt request is not affected by the I bit of CCR. Break conditions that can be set include instruction execution at a specific address and a combination of access and data at a specific address. With the address break function, the execution start point of a program containing a bug is detected and execution is branched to the correcting program. Figure 4.1 shows a block diagram of the address break.

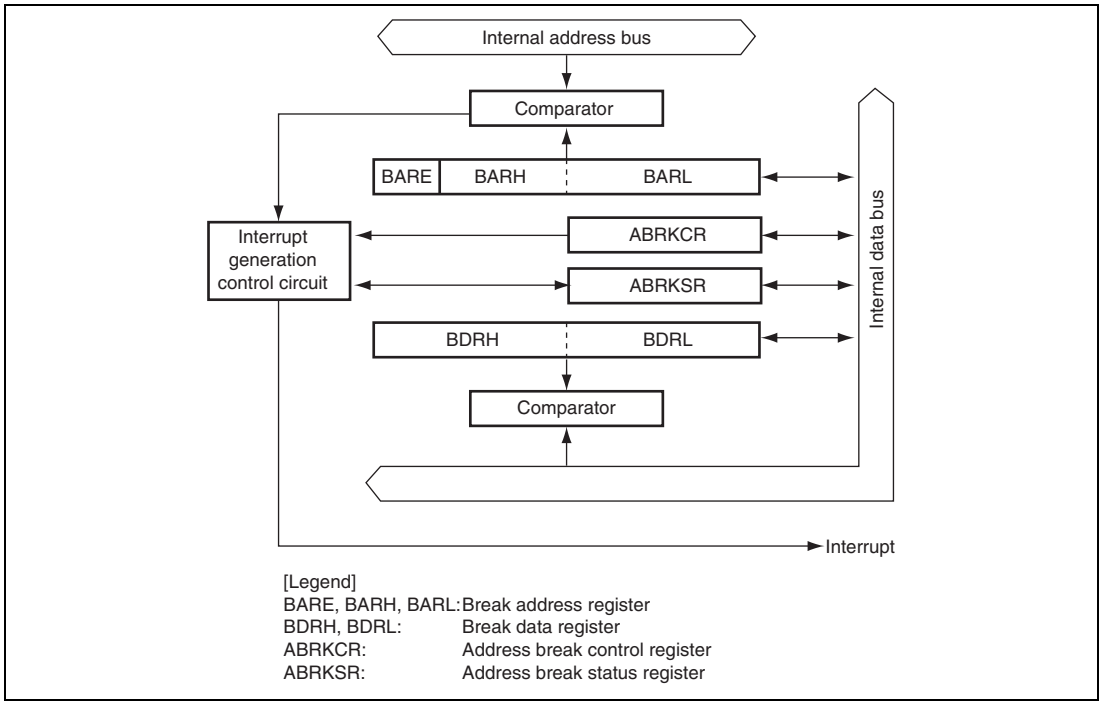



**Figure 4.1 Block Diagram of Address Break**

## 4.1 Register Descriptions

The address break has the following registers.

- Address break control register (ABRKCR)
- Address break status register (ABRKSR)
- Break address registers E, H, L (BARE, BARH, BARL)
- Break data register (BDRH, BDRL)

### 4.1.1 Address Break Control Register (ABRKCR)

ABRKCR sets address break conditions.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | RTINTE | 1 | R/W | RTE Interrupt Enable<br>When this bit is 0, the interrupt immediately after executing RTE is masked and then one instruction must be executed. When this bit is 1, the interrupt is not masked. |
| 6 | CSEL1 | 0 | R/W | Condition Select 1 and 0<br>These bits set address break conditions.<br>00: Instruction execution cycle<br>01: CPU data read cycle<br>10: CPU data write cycle<br>11: CPU data read/write cycle |
| 5 | CSEL0 | 0 | R/W | |
| 4 | ACMP2 | 0 | R/W | Address Compare 2 to 0<br>These bits set the comparison condition between the address set in BAR and the internal address bus.<br>000: Compares 24-bit addresses<br>001: Compares upper 20-bit addresses<br>010: Compares upper 16-bit addresses<br>011: Compares upper 12-bit addresses<br>1xx: Reserved |
| 3 | ACMP1 | 0 | R/W | |
| 2 | ACMP0 | 0 | R/W | |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | DCMP1 | 0 | R/W | Data Compare 1 and 0 |
| 0 | DCMP0 | 0 | R/W | These bits set the comparison condition between the data set in BDR and the internal data bus.<br>00: No data comparison<br>01: Compares lower 8-bit data between BDRL and data bus<br>10: Compares upper 8-bit data between BDRH and data bus<br>11: Compares 16-bit data between BDR and data bus |

[Legend]
x: Don't care.

When an address break is set in the data read cycle or data write cycle, the data bus used will depend on the combination of the byte/word access and address. Table 4.1 shows the access and data bus used. When an I/O register space with an 8-bit data bus width is accessed in word size, a byte access is generated twice. For details on data widths of each register, see section 22.1, Register Addresses (Address Order).

**Table 4.1 Access and Data Bus Used**

| | Word Access | | Byte Access | |
|---|---|---|---|---|
| | **Even Address** | **Odd Address** | **Even Address** | **Odd Address** |
| ROM space | Upper 8 bits | Lower 8 bits | Upper 8 bits | Upper 8 bits |
| RAM space | Upper 8 bits | Lower 8 bits | Upper 8 bits | Upper 8 bits |
| I/O register with 8-bit data bus width | Upper 8 bits | Upper 8 bits | Upper 8 bits | Upper 8 bits |
| I/O register with 16-bit data bus width | Upper 8 bits | Lower 8 bits | — | — |

### 4.1.2 Address Break Status Register (ABRKSR)

ABRKSR consists of the address break interrupt flag and the address break interrupt enable bit.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | ABIF | 0 | R/W | Address Break Interrupt Flag<br>[Setting condition]<br>When the condition set in ABRKCR is satisfied<br>[Clearing condition]<br>When 0 is written after ABIF=1 is read |
| 6 | ABIE | 0 | R/W | Address Break Interrupt Enable<br>When this bit is 1, an address break interrupt request is enabled. |
| 5 to 0 | — | All 1 | — | Reserved<br>These bits are always read as 1. |

### 4.1.3 Break Address Registers E, H, L (BARE, BARH, BARL)

BAR (BARE, BARH, BARL) is a 24-bit readable/writable register that sets the address for generating an address break interrupt. The initial value of this register is H'FFFFFF. When setting the address break condition to the instruction execution cycle, set the first byte address of the instruction.

### 4.1.4 Break Data Registers H, L (BDRH, BDRL)

BDR (BDRH, BDRL) is a 16-bit readable/writable register that sets the data for generating an address break interrupt. BDRH is compared with the upper 8-bit data bus. BDRL is compared with the lower 8-bit data bus. When memory or registers are accessed by byte, the upper 8-bit data bus is used for even and odd addresses in the data transmission. Therefore, comparison data must be set in BDRH for byte access. For word access, the data bus used depends on the address. See section 4.1.1, Address Break Control Register (ABRKCR), for details. The initial value of this register is undefined.

## 4.2 Operation

When the ABIE bit in ABRKSR is set to 1, if the ABIF bit in ABRKSR is set to 1 by the combination of the address set in BAR, the data set in BDR, and the conditions set in ABRKCR, the address break function generates an interrupt request to the CPU. When the interrupt request is accepted, interrupt exception handling starts after the instruction being executed ends. The address break interrupt is not masked because of the I bit in CCR of the CPU.

Figures 4.2 (1) to (2) show the operation examples of the address break interrupt setting.

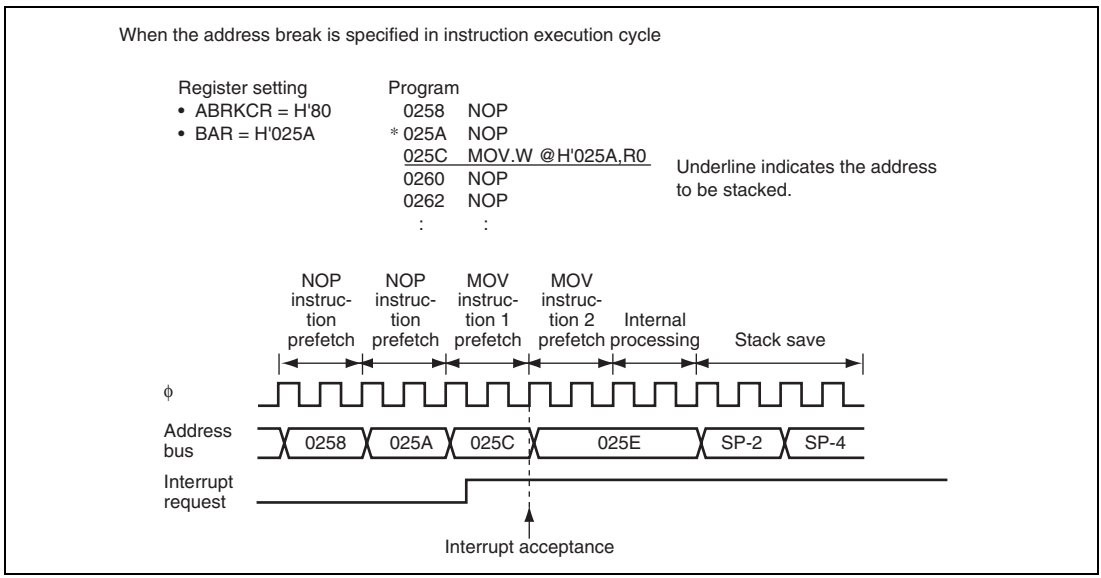



**Figure 4.2 Address Break Interrupt Operation Example (1)**

When the address break is specified in the data read cycle

Register setting
- ABRKCR = H'A0
- BAR = H'025A

Program

```
  0258  NOP
  025A  NOP
* 025C  MOV.W @H'025A,R0
  0260  NOP
  0262  NOP
    :     :
```

Underline indicates the address to be stacked.

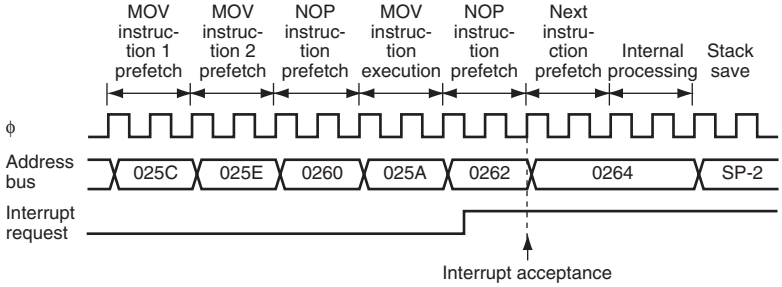



**Figure 4.2 Address Break Interrupt Operation Example (2)**

# Section 5 Clock Pulse Generators

The clock pulse generator consists of a system clock generating circuitry, a subclock generating circuitry, and two prescalers. The system clock generating circuitry includes a system clock oscillator, a duty correction circuit, an on-chip oscillator, an on-chip oscillator divider, a clock select circuit, and a system clock divider. The subclock generating circuitry includes a subclock oscillator and a subclock divider.

Figure 5.1 shows a block diagram of the clock pulse generator.

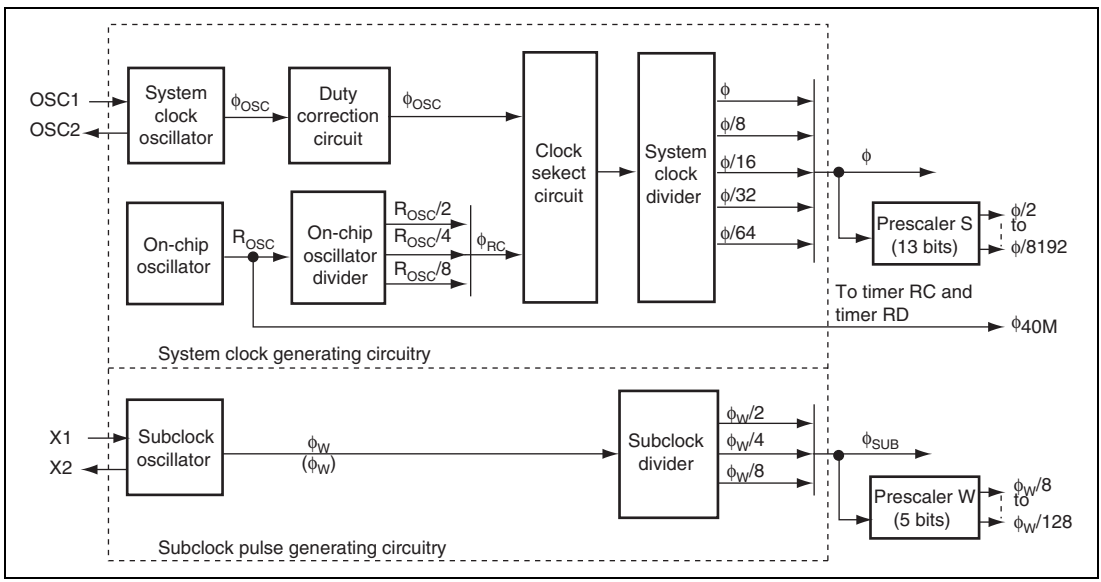



**Figure 5.1 Block Diagram of Clock Pulse Generators**

The system clock ($\phi$) and subclock ($\phi_{SUB}$) are basic clocks on which the CPU and on-chip peripheral modules operate. The system clock is divided by a value from 2 to 8192 in prescaler S, and the subclock is divided by a value from 8 to 128 in prescaler W. These divided clocks are supplied to respective on-chip peripheral modules. The on-chip oscillator can generate system clock $\phi_{RC}$, which is produced by dividing $R_{OSC}$ by 2, 4, or 8, and the $\phi_{40M}$ clock supplied to timer RC and timer RD.

## 5.1 Features

- Choice of two clock sources
  On-chip oscillator clock
  External oscillator clock
- Choice of two frequencies of the on-chip oscillator by the user software
  40 MHz
  32 MHz
  The signal generated by dividing the above clock by a value from 2 to 8 can be used as the system clock and the above clock can be used as the clock source for timer RC or timer RD.
- Frequency trimming
  The initial frequency of the on-chip oscillator is within the range shown above, so users do not need to trim the frequency. If needed, users can adjust the on-chip oscillator frequency to the range by rewriting the trimming registers.
- Interrupt can be requested to the CPU when the system clock is changed from the external clock to the on-chip oscillator clock.

## 5.2 Register Descriptions

Clock oscillators are controlled by the following registers.

- RC control register (RCCR)
- RC trimming data protect register (RCTRMDPR)
- RC trimming data register (RCTRMDR)
- Clock control/status register (CKCSR)

### 5.2.1 RC Control Register (RCCR)

RCCR controls the on-chip oscillator.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | RCSTP | 0 | R/W | On-Chip Oscillator Standby<br>The on-chip oscillator standby state is entered by setting this bit to 1. |
| 6 | FSEL | 1 | R/W | Frequency Select for On-Chip Oscillator<br>The system clock is generated by dividing this clock while this clock is supplied to timer RC or timer RD ($\phi_{40M}$).<br>0: 32MHz<br>1: 40MHz |
| 5 | VCLSEL | 0 | R/W | Power Supply Select for On-Chip Oscillator<br>0: Selects VBGR<br>1: Selects VCL<br>When VCL is selected, the accuracy of the on-chip oscillator frequency cannot be guaranteed. |
| 4 to 2 | — | All 0 | — | Reserved<br>These bits are always read as 0. |
| 1 | RCPSC1 | 1 | R/W | Division Ratio Select for On-Chip Oscillator |
| 0 | RCPSC0 | 0 | R/W | These bits select the operating clock frequency in active mode or sleep mode when the on-chip oscillator is in use. The division ratio for dividing $R_{osc}$ changes right after rewriting this bit.<br>These bits can only be written to when the CKSTA bit in CKCSR is 0.<br>0X: $R_{osc}/2$<br>10: $R_{osc}/4$<br>11: $R_{osc}/8$ |

[Legend]
X: Don't care

### 5.2.2 RC Trimming Data Protect Register (RCTRMDPR)

RCTRMDPR controls RCTRMDPR itself and writing to RCTRMDR. Use the MOV instruction to rewrite this register. Bit manipulation instruction cannot change the settings.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | WRI | 1 | W | Write Inhibit<br>Only when writing 0 to this bit, this register can be written to. This bit is always read as 1. |
| 6 | PRWE | 0 | R/W | Protect Information Write Enable<br>Bits 5 and 4 can be written to when this bit is set to 1.<br>[Setting condition]<br>• When writing 0 to the WRI bit and writing 1 to the PRWE bit<br>[Clearing conditions]<br>• Reset<br>• When writing 0 to the WRI bit and writing 0 to the PRWE bit |
| 5 | LOCKDW | 0 | R/W | Trimming Data Register Lock Down<br>The RC trimming data register (RCTRMDR) cannot be written to when this bit is set to 1. Once this bit is set to 1, this register cannot be written to until a reset is input even if 0 is written to this bit.<br>[Setting condition]<br>• When writing 0 to the WRI bit and writing 1 to the TRMDRWE bit while the PRWE bit is 1.<br>[Clearing condition]<br>• Reset |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | TRMDRWE | 0 | R/W | Trimming Date Register Write Enable<br>This register can be written to when the LOCKDW bit is 0 and this bit is 1.<br>[Setting condition]<br>• When writing 0 to the WRI bit and writing 1 to the TRMDRWE bit while the PRWE bit is 1.<br>[Clearing conditions]<br>• Reset<br>• When writing 0 to the WRI bit and writing 0 to the TRMDRWE bit while the PRWE bit is 1. |
| 3 to 0 | — | All 1 | — | Reserved<br>These bits are always read as 1. |

### 5.2.3 RC Trimming Data Register (RCTRMDR)

RCTRMDR stores the trimming data of the on-chip oscillator frequency.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TRMD7 | (0)* | R/W | Trimming Data |
| 6 | TRMD6 | (0)* | R/W | The trimming data is loaded to this register right after a reset. The read data from these bits is always undefined. |
| 5 | TRMD5 | (0)* | R/W | |
| 4 | TRMD4 | (0)* | R/W | The on-chip oscillator frequency can be trimmed by rewriting these bits. The frequency of the on-chip oscillator changes right after rewriting these bits. These bits are initialized to H'00. |
| 3 | TRMD3 | (0)* | R/W | |
| 2 | TRMD2 | (0)* | R/W | |
| 1 | TRMD1 | (0)* | R/W | Changes in frequency are shown below (bit 7 is a sign bit): |
| 0 | TRMD0 | (0)* | R/W | (Min.) H'80 ← H'FF ← H'00 →H'01 → H'7F (Max.) |

Note: * These values are initialized while loading the trimming data.

### 5.2.4 Clock Control/Status Register (CKCSR)

CKCSR selects the OSC pin function, controls switching system clocks, and indicates the system clock state. These bits must be written in active mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PMRJ1 | 0 | R/W | OSC Pin Function Select 1 and 0 |
| 6 | PMRJ0 | 0 | R/W | PMRJ1 PMRJ0 OSC2 OSC1<br>0 0 I/O I/O<br>1 0 CLKOUT I/O<br>0 1 (open) OSC1 (external clock input)<br>1 1 OSC2 OSC1 |
| 5 | — | 0 | — | Reserved<br>This bit is always read as 0. |
| 4 | OSCSEL | 0 | R/W | LSI Operating Clock Select<br>This bit selects the system clock of this LSI.<br>0: Selects the on-chip oscillator clock as the system clock.<br>1: Selects the external clock as the system clock.<br>[Setting condition]<br>• When writing 1 while the CKSWIF bit is 0*<br>[Clearing condition]<br>• When writing 0<br>Note: * When the on-chip oscillator is in the standby state (the RCTSP bit in RCCR is set to 1), do not write 1 to this bit. When this bit is written to 1, the on-chip oscillator should be in operation. |
| 3 | CKSWIE | 0 | R/W | Clock Switch Interrupt Enable<br>Setting this bit to 1 enables the clock switch interrupt request. |
| 2 | CKSWIF | 0 | R/W | Clock Switch Interrupt Request Flag<br>[Setting condition]<br>• When the external clock is switched to the on-chip oscillator clock<br>[Clearing condition]<br>• When writing 0 after reading 1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 0 | CKSTA | 0 | R | LSI Operating Clock Status<br>0: This LSI operates on on-chip oscillator clock.<br>1: This LSI operates on external clock. |

## 5.3 System Clock Oscillator

### 5.3.1 State Transition of System Clock

The system clock of this LSI is generated from the on-chip oscillator clock after a reset. System clock sources can be switched from the on-chip oscillator clock to the external clock and vice versa by the user software.

Figure 5.2 shows the state transition of the system clock.

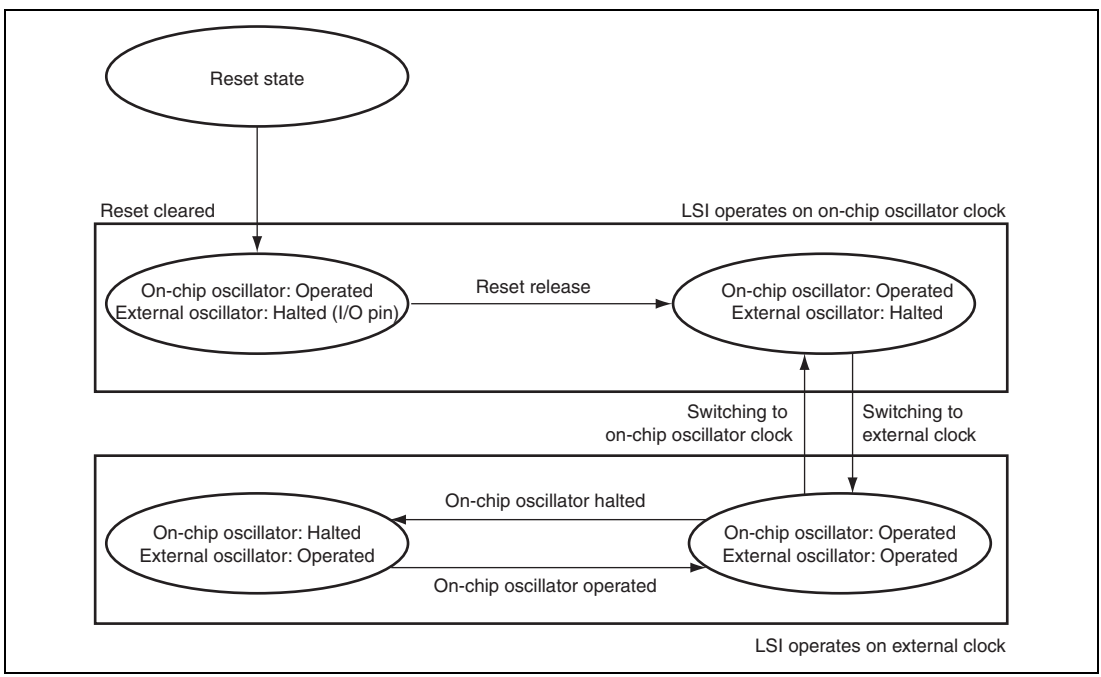



**Figure 5.2 State Transition of System Clock**

### 5.3.2 Clock Control Operation

Figure 5.3 shows the flowchart to switch clock sources from the on-chip oscillator to the external clock. Figure 5.4 shows the flowchart to switch clock sources from the external clock to the on-chip oscillator.

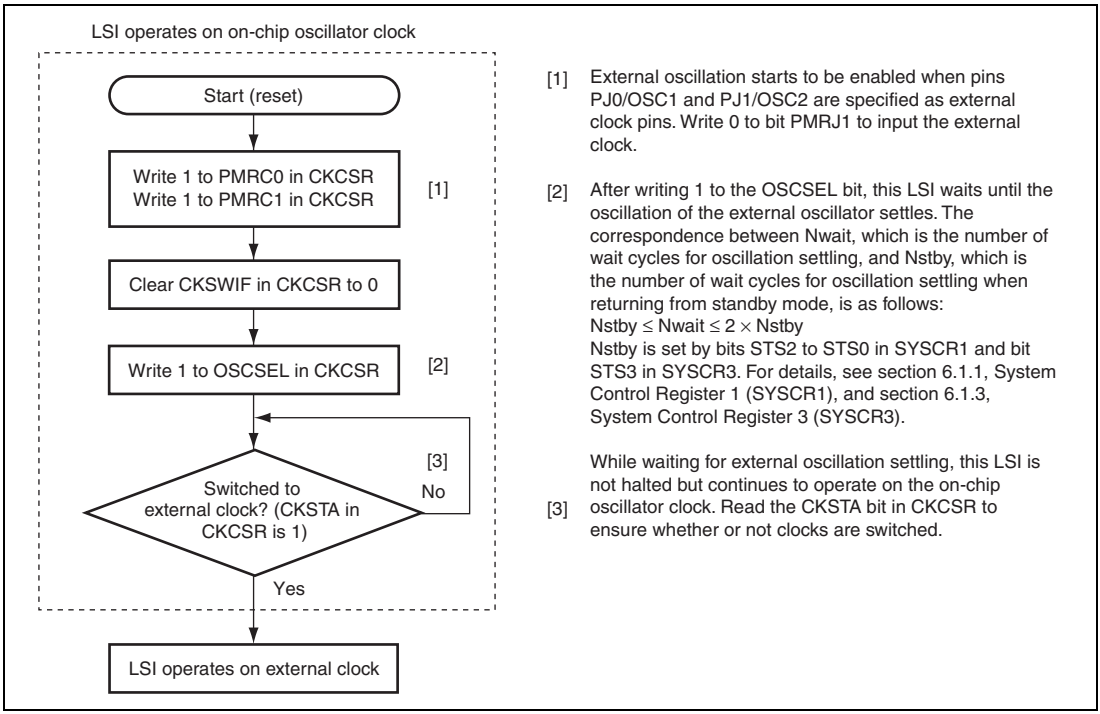



**Figure 5.3 Flowchart of Clock Switching (From On-Chip Oscillator Clock to External Clock)**

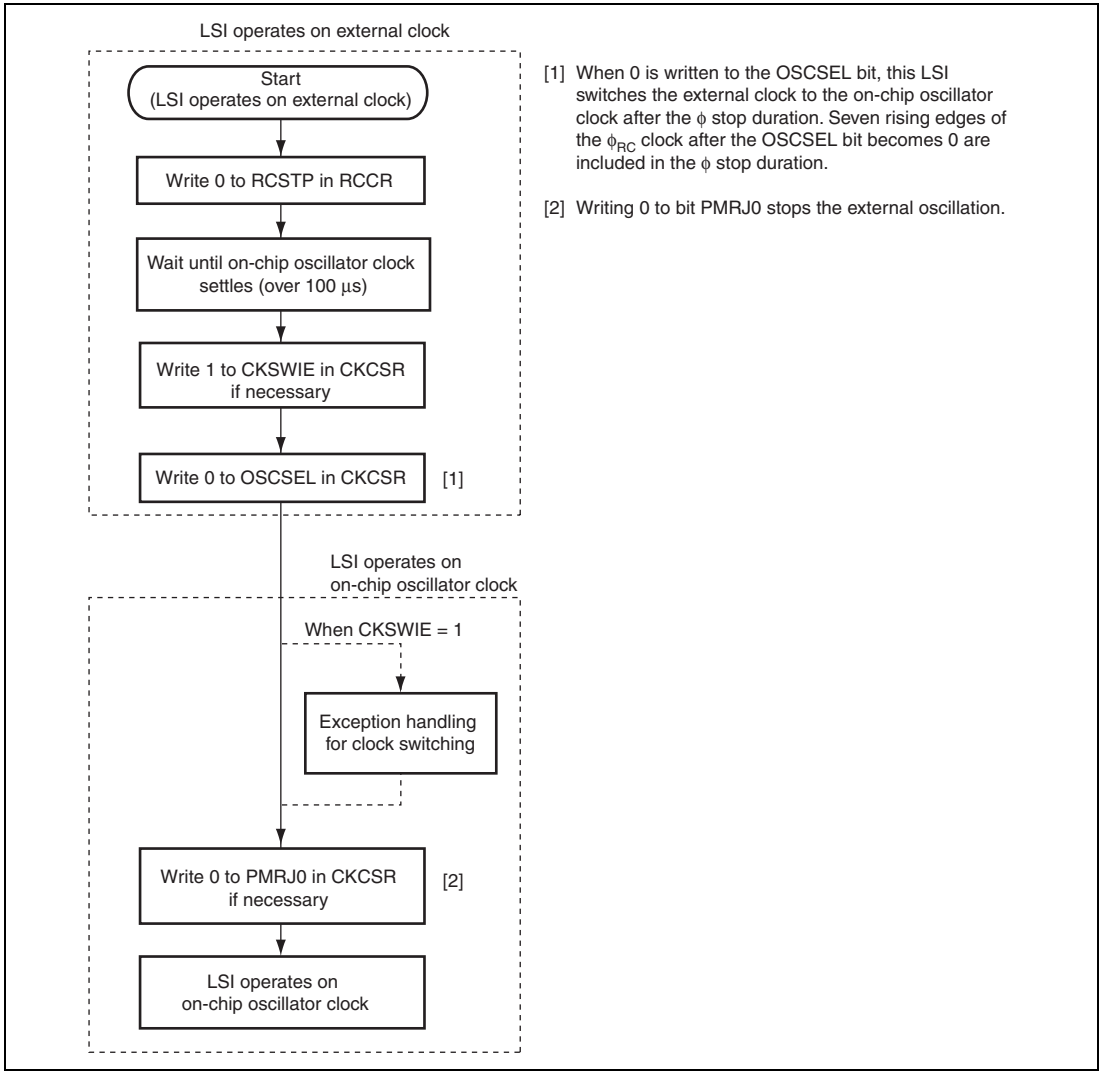
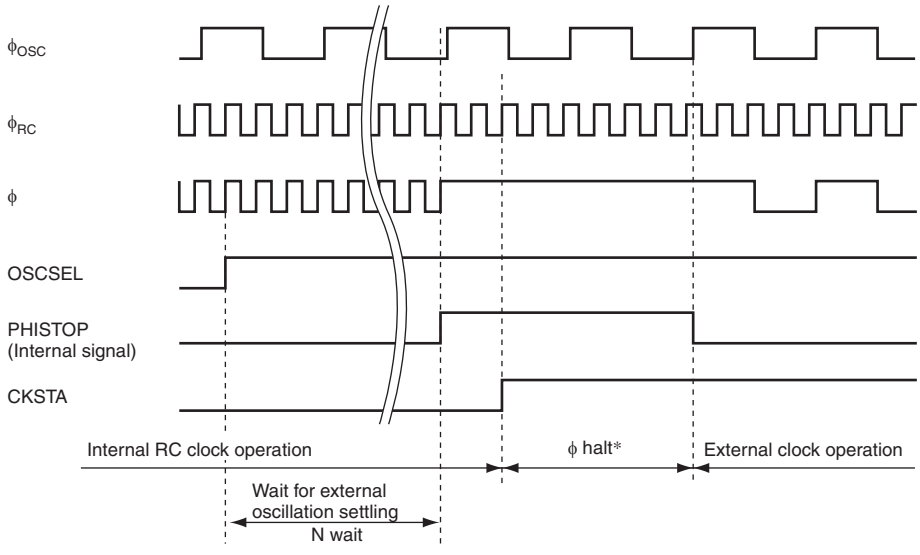



**Figure 5.4 Flowchart of Clock Switching (From External Clock to On-Chip Oscillator Clock)**

### 5.3.3 Clock Change Timing

The timing for changing clocks are shown in figures 5.5 and 5.6.

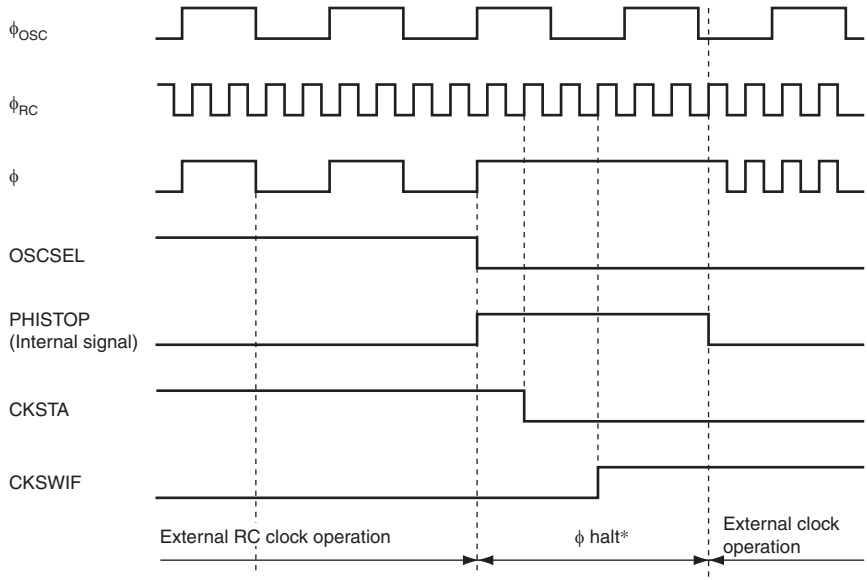



[Legend]

| | |
|---|---|
| $\phi_{OSC}$: | External clock |
| $\phi_{RC}$: | Internal RC clock |
| $\phi$: | System clock |
| OSCSEL: | Bit 4 in CKCSR |
| PHISTOP: | System clock stop control signal |
| CKSTA: | Bit 0 in CKCSR |

Note: * The $\phi$ halt duration is the duration from the timing when the $\phi$ clock stops to the first rising edge of the $\phi_{OSC}$ clock after seven clock cycles of the $\phi_{RC}$ clock have elapsed.

**Figure 5.5 Timing Chart of Switching from On-Chip Oscillator Clock to External Clock**




[Legend]

| | |
|---|---|
| $\phi_{OSC}$: | External clock |
| $\phi_{RC}$: | Internal RC clock |
| $\phi$: | System clock |
| OSCSEL: | Bit 4 in CKCSR |
| PHISTOP: | System clock stop control signal |
| CKSTA: | Bit 0 in CKCSR |
| CKSWIF: | Bit 2 in CKCSR |

Note: * The $\phi$ halt duration is the duration from the timing when the $\phi$ clock stops to the seventh rising edge of the $\phi_{RC}$ clock.

**Figure 5.6 Timing Chart to Switch from External Clock to On-Chip Oscillator Clock**

## 5.4 Trimming of On-Chip Oscillator Frequency

Users can trim the on-chip oscillator frequency, supplying the external reference pulses with the input capture function in timer RC or timer RD. An example of trimming flow using timer RC and a timing chart are shown in figures 5.7 and 5.8, respectively. Because RCTRMDR is initialized by a reset, when users have trimmed the oscillators, some operations after a reset are necessary, such as trimming it again or saving the trimming value in an external device for later reloading.

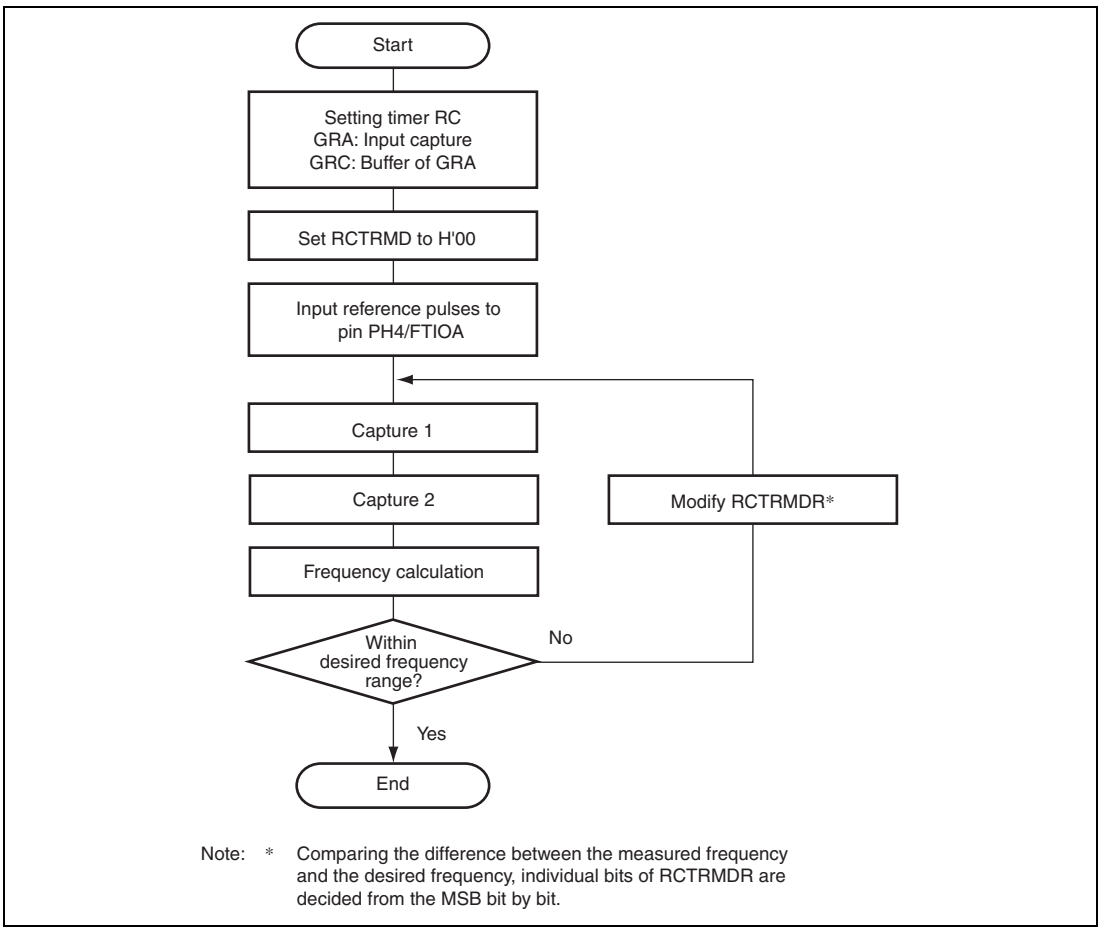



**Figure 5.7 Example of Trimming Flow for On-Chip Oscillator Frequency**

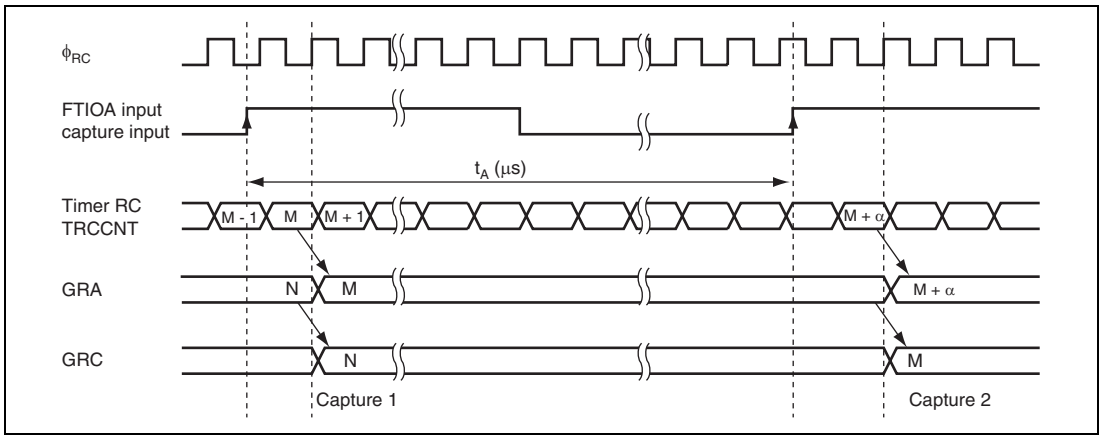



**Figure 5.8 Timing Chart of Trimming of On-Chip Oscillator Frequency**

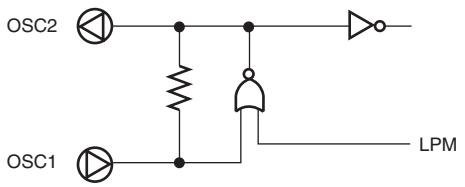
The on-chip oscillator frequency is obtained by the expression below. Since the input-capture input is sampled at the rate of $\phi_{RC}$, the calculated result includes a sampling error of ±1 $\phi_{RC}$ clock cycle.

$$\phi RC = \frac{(M + \alpha) - M}{t_A} \text{ (MHz)}$$

$\phi$RC: Frequency of divided on-chip oscillator clock (MHz)
$t_A$: Cycle of reference clock (µs)
M: Timer RC counter value

## 5.5 External Oscillator

This LSI has two methods to supply external clock pulses into it: connecting a crystal or ceramic resonator, and an external clock. Oscillation pins PJ0/OSC1 and PJ1/OSC2/CLKOUT are common with general ports PC0 and PC1, respectively. To set pins PC0 and PC1 as crystal resonator or external clock input ports, refer to section 5.3.2, Clock Control Operation.




LPM: Power-down mode (standby mode, subactive mode, or subsleep mode)

**Figure 5.9 Block Diagram of External Oscillator**

### 5.5.1 Connecting Crystal Resonator

Figure 5.10 shows an example of connecting a crystal resonator. An AT-cut parallel-resonance crystal resonator should be used. Figure 5.11 shows the equivalent circuit of a crystal resonator. A resonator having the characteristics given in table 5.1 should be used.

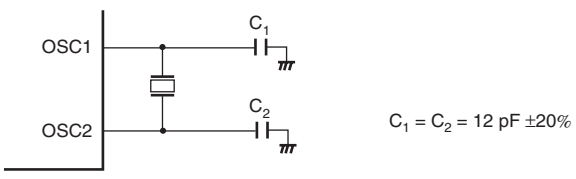



**Figure 5.10 Example of Connection to Crystal Resonator**

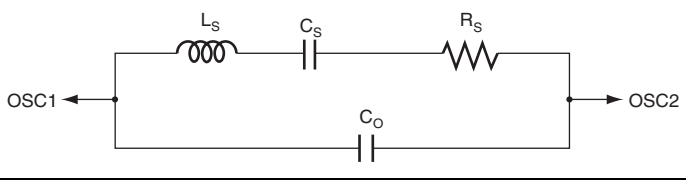



**Figure 5.11 Equivalent Circuit of Crystal Resonator**

**Table 5.1 Crystal Resonator Parameters**

| Frequency (MHz) | 4 | 8 | 10 | 16 | 20 |
|---|---|---|---|---|---|
| $R_S$ (Max.) | 120 Ω | 80 Ω | 60 Ω | 50 Ω | 40 Ω |
| $C_0$ (Max.) | 7 pF | | | | |

### 5.5.2 Connecting Ceramic Resonator

Figure 5.12 shows an example of connecting a ceramic resonator.

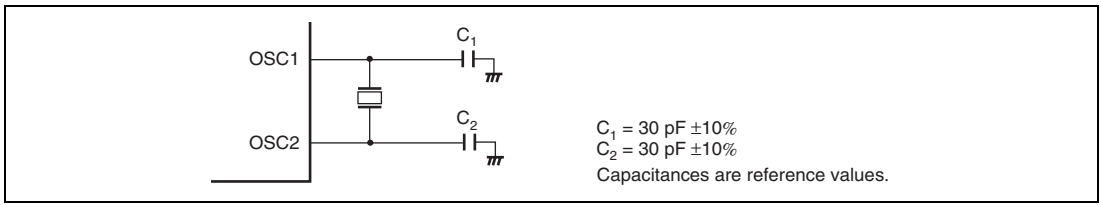



**Figure 5.12 Example of Connection to Ceramic Resonator**

### 5.5.3 External Clock Input Method

To use the external clock, input the external clock on pin OSC1 and leave pin OSC2 open. Figure 5.13 shows an example of connection. The duty cycle of the external clock signal must be 45 to 55%.

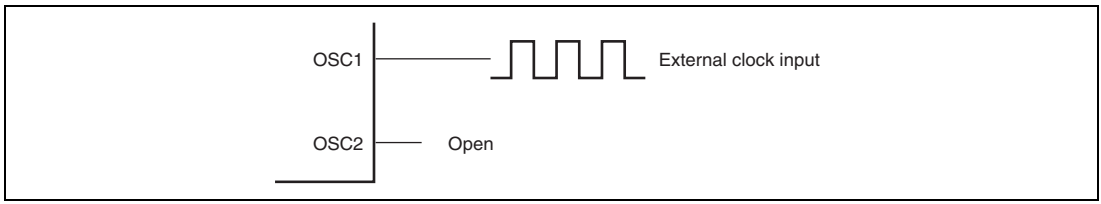
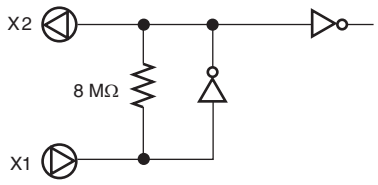



**Figure 5.13 Example of External Clock Input**

## 5.6 Subclock Generator

Figure 5.14 shows a block diagram of the subclock generator.

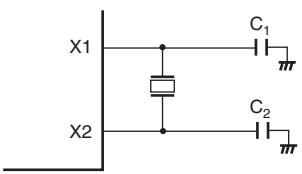



Note: Resistance is a reference value.

**Figure 5.14 Block Diagram of Subclock Generator**

### 5.6.1 Connecting 32.768-kHz Crystal Resonator

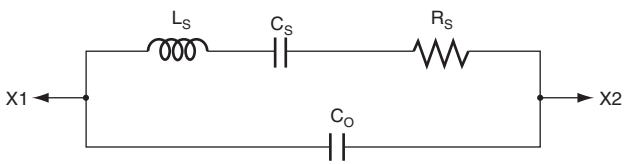
Clock pulses can be supplied to the subclock divider by connecting a 32.768-kHz crystal resonator, as shown in figure 5.15. Figure 5.16 shows the equivalent circuit of the 32.768-kHz crystal resonator.




$C_1 = C_2 = 15$ pF (typ.)

Capacitances are reference values.

**Figure 5.15 Typical Connection to 32.768-kHz Crystal Resonator**




$C_O = 1.5$ pF (typ.)
$R_S = 14$ kΩ (typ.)
fW = 32.768 kHz

Note: Parameters are reference values.

**Figure 5.16 Equivalent Circuit of 32.768-kHz Crystal Resonator**

### 5.6.2 Pin Connection when not Using Subclock

When the subclock is not used, connect pin X1 to VCL or VSS and leave pin X2 open, as shown in figure 5.17.

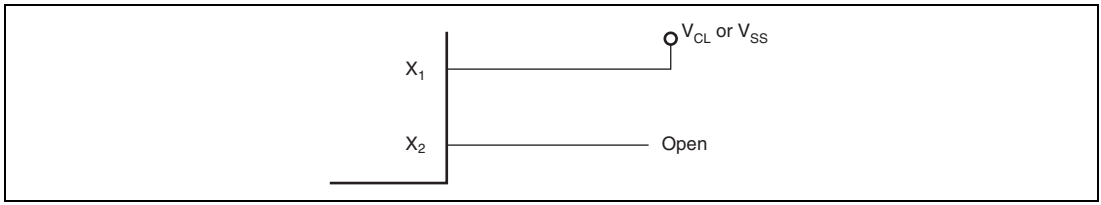



**Figure 5.17 Pin Connection when not Using Subclock**

## 5.7 Prescaler

### 5.7.1 Prescaler S

Prescaler S is a 13-bit counter using the system clock ($\phi$) as its input clock. The outputs, which are divided clocks, are used as internal clocks by the on-chip peripheral modules. Prescaler S is initialized to H'0000 by a reset, and starts counting on exit from the reset state. In standby mode and subsleep mode, the system clock pulse generator stops. Prescaler S also stops and is initialized to H'0000. It cannot be read from or written to by the CPU.

The outputs from prescaler S is shared by the on-chip peripheral modules. The division ratio can be set separately for each on-chip peripheral module. In active mode and sleep mode, the clock input to prescaler S is a system clock with the division ratio specified by bits MA2 to MA0 in SYSCR2.

### 5.7.2 Prescaler W

Prescaler W is a 5-bit counter using a 32.768 kHz signal divided by 4 ($\phi_W/4$) as its input clock. The divided output is used for clock time base operation of timer A. Prescaler W is initialized to H'00 by a reset, and starts counting on exit from the reset state. Even in standby mode, subactive mode, or subsleep mode, prescaler W continues functioning so long as clock signals are supplied to pins $X_1$ and $X_2$.

## 5.8 Usage Notes

### 5.8.1 Note on Resonators

Resonator characteristics are closely related to board design and should be carefully evaluated by the user, referring to the examples shown in this section. Resonator circuit parameters will differ depending on the resonator element, stray capacitance of the PCB, and other factors. Suitable values should be determined in consultation with the resonator element manufacturer. Design the circuit so that the resonator element never receives voltages exceeding its maximum rating.

### 5.8.2 Notes on Board Design

When using a crystal resonator (ceramic resonator), place the resonator and its load capacitors as close as possible to pins OSC1 and OSC2. Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation (see figure 5.18).

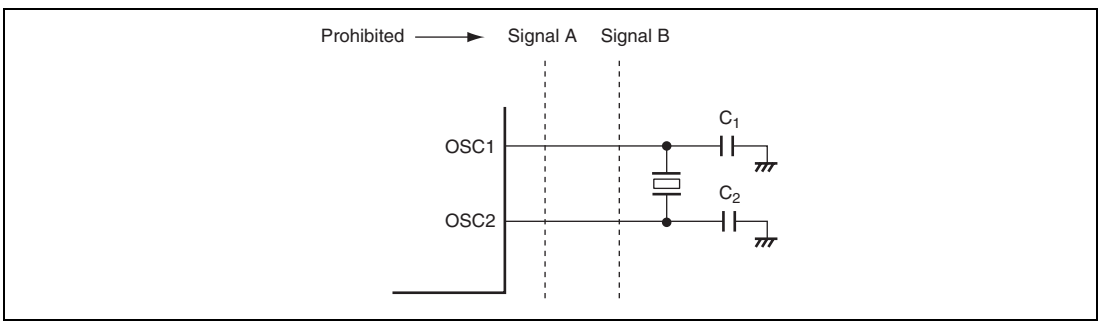



**Figure 5.18 Example of Incorrect Board Design**

# Section 6 Power-Down Modes

This LSI has five operating modes after a reset: a normal active mode and four power-down modes in which power consumption is significantly reduced. In addition to these modes, there is a module standby function in which power consumption is also reduced by selectively halting on-chip module functions.

- Active mode

  The CPU and all on-chip peripheral modules operate on the system clock. The system clock source can be selected from among ϕosc, Rosc/2, Rosc/4, and Rosc/8. The system clock frequency can be selected from among ϕ, ϕ/8, ϕ/16, ϕ/32, and ϕ/64.
- Subactive mode

  The CPU and all on-chip peripheral modules operate on the subclock. The subclock frequency can be selected from ϕw/2, ϕw/4, or ϕw/8.
- Sleep mode

  The CPU halts. On-chip peripheral modules operate on the system clock.
- Subsleep mode

  The CPU halts. On-chip peripheral modules operate on the subclock.
- Standby mode

  The CPU and all on-chip peripheral modules halt. When the clock time-base function is selected, the RTC operates.
- Module standby function

  Independent of the above modes, power consumption can be reduced by halting individual on-chip peripheral modules that are not in use.

## 6.1 Register Descriptions

The registers related to power-down modes are listed below. For details on the serial mode control register (SCI3_3 module standby), see section 17.1, Features.

- System control register 1 (SYSCR1)
- System control register 2 (SYSCR2)
- System control register 3 (SYSCR3)
- Module standby control register 1 (MSTCR1)
- Module standby control register 2 (MSTCR2)
- Module standby control register 4 (MSTCR4)
- Serial Mode Control Register (SMCR)

### 6.1.1 System Control Register 1 (SYSCR1)

SYSCR1, SYSCR2, and SYSCR3 control the power-down modes.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | SSBY | 0 | R/W | Software Standby<br>This bit selects the mode to transit after the execution of the SLEEP instruction.<br>0: Enters sleep mode or subsleep mode.<br>1: Enters standby mode.<br>For details, see table 6.2. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | STS2 | 0 | R/W | Standby Timer Select 2 to 0 |
| 5 | STS1 | 0 | R/W | These bits specify the waiting time in number of cycles until clocks are supplied after the system clock oscillator starts oscillation when making a transition from the standby, subactive, or subsleep mode to the active or sleep mode. The number of cycles for the waiting time should be specified so that the waiting time is 6.5 ms or more when the external oscillator is used as the system clock source after the transition. The waiting time should be 100 μs or more when the on-chip oscillator is used as the system clock source after the transition. The relationship between the setting and the number of cycles is shown in table 6.1. A clock used for counting the number of cycles is not divided regardless of the setting in bits MA2 to MA0 in SYSCR2. When the system clock source after a transition is the external oscillator or on-chip oscillator, the $\phi_{OSC}$ or $\phi$RC clock is used for counting, respectively.<br><br>These bits also specify the waiting time until the external oscillator settles when system clock sources are switched from the on-chip oscillator to the external clock by user software. The relationship of waiting times between the above transition and clock switching is shown below. The number of cycles for external oscillator settling should be specified so that the Nstby value multiplied by the external oscillator frequency is 6.5 ms or more. In this case, a clock used for counting the number of cycles is the $\phi_{RC}$ clock divided by the setting in bits MA2 to MA0 in SYSCR2.<br><br>Nstby $\leq$ Nwait $\leq 2 \times$ Nstby<br><br>Nwait: The number of waiting cycles for external oscillator settling<br><br>Nstby: The number of waiting cycles when returning from a standby mode |
| 4 | STS0 | 0 | R/W | |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | NESEL | 0 | R/W | Noise Elimination Sampling Frequency Select<br>This bit selects the clock frequency to sample the watch clock signal ($\phi_W$) generated by the subclock oscillator. The oscillator clock ($\phi_{OSC}$) generated by the system clock oscillator or the $\phi_{RC}$ clock generated by the on-chip oscillator can be used as the sampling clock source. When $\phi_{OSC}$ or $\phi_{RC}$ = 4 to 20 MHz, set this bit to 0.<br>0: Sampling rate is $\phi_{OSC}$/16 or $\phi_{RC}$/16<br>1: Sampling rate is $\phi_{OSC}$/4 or $\phi_{RC}$/4 |
| 2 to 0 | — | All 0 | — | Reserved<br>These bits are always read as 0. |

**Table 6.1 Operating Frequency and Waiting Time**

| Bit Name | | | | Cycle Count for Waiting Time | Operating Frequency | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STS3 | STS2 | STS1 | STS0 | | 20 MHz | 16 MHz | 10 MHz | 8 MHz | 4 MHz | 2 MHz | 1 MHz | 0.5 MHz |
| x | 0 | 0 | 0 | 8,192 cycles | 0.4 | 0.5 | 0.8 | 1.0 | 2.0 | 4.1 | 8.1 | 16.4 |
| x | 0 | 0 | 1 | 16,384 cycles | 0.8 | 1.0 | 1.6 | 2.0 | 4.1 | 8.2 | 16.4 | 32.8 |
| x | 0 | 1 | 0 | 32,768 cycles | 1.6 | 2.0 | 3.3 | 4.1 | 8.2 | 16.4 | 32.8 | 65.5 |
| x | 0 | 1 | 1 | 65,536 cycles | 3.3 | 4.1 | 6.6 | 8.2 | 16.4 | 32.8 | 65.5 | 131.1 |
| x | 1 | 0 | 0 | 131,072 cycles | 6.6 | 8.2 | 13.1 | 16.4 | 32.8 | 65.5 | 131.1 | 262.1 |
| 1 | 1 | 0 | 1 | 1,024 cycles | 0.05 | 0.06 | 0.10 | 0.13 | 0.26 | 0.51 | 1.02 | 2.05 |
| 1 | 1 | 1 | 0 | 128 cycles | 0.00 | 0.00 | 0.01 | 0.02 | 0.03 | 0.06 | 0.13 | 0.26 |
| 1 | 1 | 1 | 1 | 16 cycles | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.03 |
| 0 | 1 | 0 | 1 | 4,096 cycles | 0.20 | 0.25 | 0.40 | 0.51 | 1.02 | 2.05 | 4.01 | 8.19 |
| 0 | 1 | 1 | 0 | 2,048 cycles | 0.10 | 0.13 | 0.20 | 0.26 | 0.51 | 1.02 | 2.05 | 4.01 |
| 0 | 1 | 1 | 1 | 512 cycles | 0.02 | 0.03 | 0.05 | 0.06 | 0.13 | 0.26 | 0.51 | 1.02 |

[Legend]

x: Don't care

Note: Time unit is ms.

### 6.1.2 System Control Register 2 (SYSCR2)

SYSCR2 controls the power-down modes, as well as SYSCR1.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | SMSEL | 0 | R/W | Sleep Mode Select |
| 6 | LSON | 0 | R/W | Low Speed on Flag |
| 5 | DTON | 0 | R/W | Direct Transfer on Flag<br>These bits select the mode to enter after the execution of a SLEEP instruction, as well as bit SSBY of SYSCR1.<br>For details, see table 6.2. |
| 4 | MA2 | 0 | R/W | Active Mode Clock Select 2 to 0<br>These bits select the operating clock frequency in active and sleep modes. The operating clock frequency changes to the set frequency after the SLEEP instruction is executed. When the on-chip oscillator is selected as the system clock source, the on-chip oscillator output is further divided.<br>0xx: $\phi_{OSC}$ or $\phi_{RC}$<br>100: $\phi_{OSC}/8$ or $\phi_{RC}/8$<br>101: $\phi_{OSC}/16$ or $\phi_{RC}/16$<br>110: $\phi_{OSC}/32$ or $\phi_{RC}/32$<br>111: $\phi_{OSC}/64$ or $\phi_{RC}/64$ |
| 3 | MA1 | 0 | R/W | |
| 2 | MA0 | 0 | R/W | |
| 1 | SA1 | 0 | R/W | Subactive Mode Clock Select 1 and 0<br>These bits select the operating clock frequency in subactive and subsleep modes. The operating clock frequency changes to the set frequency after the SLEEP instruction is executed.<br>00: $\phi_W/8$<br>01: $\phi_W/4$<br>1x: $\phi_W/2$ |
| 0 | SA0 | 0 | R/W | |

[Legend]

x: Don't care.

### 6.1.3 System Control Register 3 (SYSCR3)

SYSCR3 controls waiting time in combination with SYSCR1.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 1 | — | Reserved<br>These bits are always read as 0. |
| 0 | STS3 | 1 | R/W | Standby Timer Select 3<br>This bit selects the waiting time in combination with bits STS2 to STS0 in SYSCR1.<br>The relationship between the register setting and waiting time is shown in table 6.1. |

### 6.1.4 Module Standby Control Register 1 (MSTCR1)

MSTCR1 allows the on-chip peripheral modules to enter a standby state in module units.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | — | Reserved<br>This bit is always read as 0. |
| 6 | MSTIIC | 0 | R/W | IIC2 Module Standby<br>IIC2 enters the standby mode when this bit is set to 1 |
| 5 | MSTS3 | 0 | R/W | SCI3 Module Standby<br>SCI3 enters the standby mode when this bit is set to 1 |
| 4 | — | 0 | — | Reserved<br>This bit is always read as 0. |
| 3 | MSTWD | 0 | R/W | Watchdog Timer Module Standby<br>Watchdog timer enters the standby mode when this bit is set to 1. When the on-chip oscillator is selected for the watchdog timer clock, the watchdog timer operates regardless of the setting of this bit |
| 2 | — | 0 | — | Reserved<br>This bit is always read as 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | MSTTV | 0 | R/W | Timer V Module Standby<br>Timer V enters the standby mode when this bit is set to 1 |
| 0 | MSTTA | 0 | R/W | RTC Module Standby<br>RTC enters the standby mode when this bit is set to 1 |

### 6.1.5 Module Standby Control Register 2 (MSTCR2)

MSTCR2 allows the on-chip peripheral modules to enter a standby state in module units.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | MSTS3_2 | 0 | R/W | SCI3_2 Module Standby<br>SCI3_2 enters the standby mode when this bit is set to 1 |
| 6 | — | 0 | — | Reserved |
| 5 | — | 0 | — | These bits are always read as 0. |
| 4 | MSTTB1 | 0 | R/W | Timer B1 Module Standby<br>Timer B1 enters the standby mode when this bit is set to 1 |
| 3 to 1 | — | All 0 | — | Reserved<br>These bits are always read as 0. |
| 0 | MSTPWM | 0 | R/W | PWM Module Standby<br>PWM enters the standby mode when this bit is set to 1 |

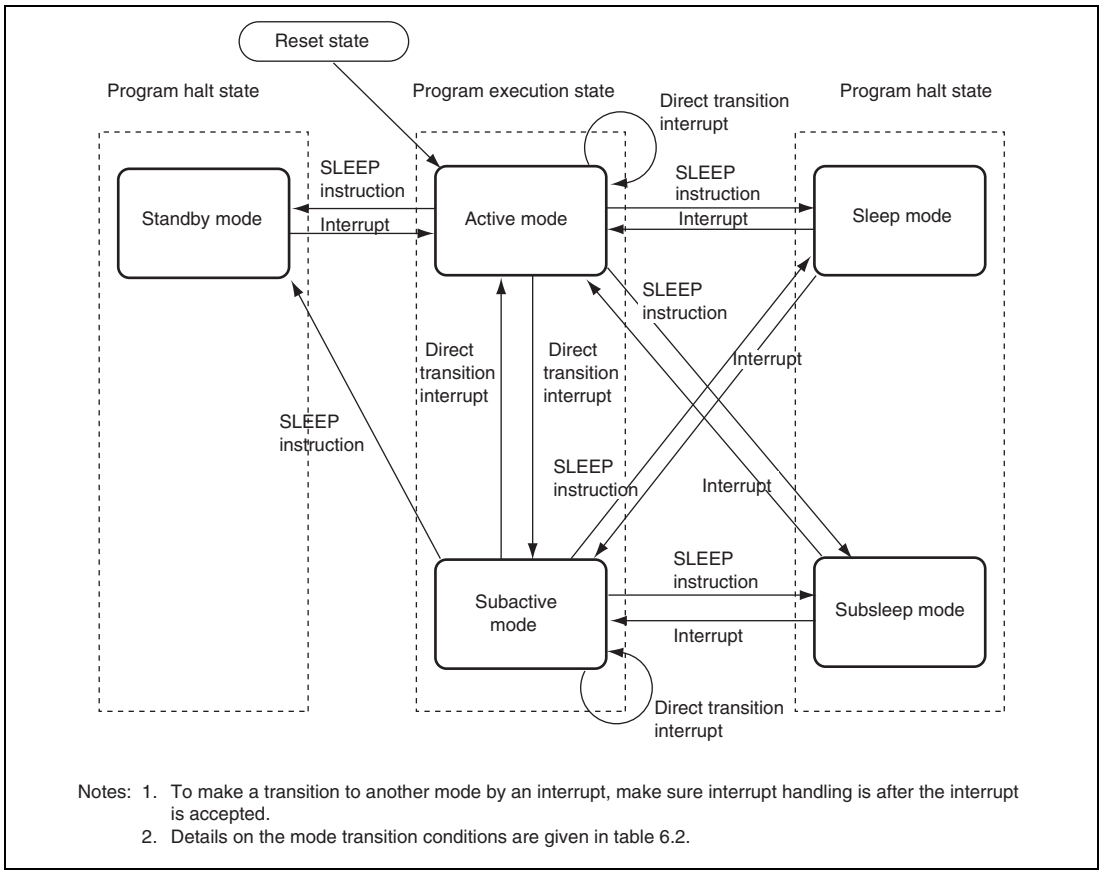
### 6.1.6 Module Standby Control Register 4 (MSTCR4)

MSTCR4 allows the on-chip peripheral modules to enter a standby state in module units.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | MSTTRC | 0 | R/W | Timer RC Module Standby<br>Timer RC enters the standby mode when this bit is set to 1 |
| 6 | MSTAD | 0 | R/W | A/D Converter Module Standby<br>A/D converter enters the standby mode when this bit is set to 1 |
| 5 | MSTTRD0 | 0 | R/W | Timer RD_0 Module Standby<br>Timer RD_0 enters the standby mode when this bit is set to 1 |
| 4 | MSTTRD1 | 0 | R/W | Timer RD_1 Module Standby<br>Timer RD_1 enters the standby mode when this bit is set to 1 |
| 3 to 0 | — | All 0 | — | Reserved<br>These bits are always read as 0. |

## 6.2 Mode Transitions and States of LSI

Figure 6.1 shows the possible transitions among these operating modes. A transition is made from the program execution state to the program halt state by executing a SLEEP instruction. Interrupts allow for returning from the program halt state to the program execution state. A direct transition between active mode and subactive mode, which are both program execution states, can be made without halting the program. The operating frequency can also be changed in the same modes by making a transition directly from active mode to active mode, and from subactive mode to subactive mode. $\overline{\text{RES}}$ input enables transitions from a mode to the reset state. Table 6.2 shows the transition conditions of each mode after the SLEEP instruction is executed and a mode to return by an interrupt. Table 6.3 shows the internal states of the LSI in each mode.




Notes: 1. To make a transition to another mode by an interrupt, make sure interrupt handling is after the interrupt is accepted.
2. Details on the mode transition conditions are given in table 6.2.

**Figure 6.1 Mode Transition Diagram**

**Table 6.2 Transition Mode after SLEEP Instruction Execution and Transition Mode due to Interrupt**

| DTON | SSBY | SMSEL | LSON | Transition Mode after SLEEP Instruction Execution | Transition Mode due to Interrupt |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Sleep mode | Active mode |
| | | | 1 | | Subactive mode |
| | | 1 | 0 | Subsleep mode | Active mode |
| | | | 1 | | Subactive mode |
| | 1 | X | X | Standby mode | Active mode |
| 1 | X | 0* | 0 | Active mode (direct transition) | — |
| | X | X | 1 | Subactive mode (direct transition) | — |

[Legend]

X: Don't care.

Note: * When a state transition is performed while SMSEL is 1, timer V, SCI3, SCI3_2, SCI3_3, and the A/D converter are reset, and all registers are set to their initial values. To use these functions after entering active mode, reset the registers.

## Table 6.3 Internal State in Each Operating Mode

| Function | | Active Mode | Sleep Mode | Subactive Mode | Subsleep Mode | Standby Mode |
|---|---|---|---|---|---|---|
| System clock oscillator | | Functioning | Functioning | Halted | Halted | Halted |
| Subclock oscillator | | Functioning | Functioning | Functioning | Functioning | Functioning |
| CPU operations | Instructions | Functioning | Halted | Functioning | Halted | Halted |
| | Registers | Functioning | Retained | Functioning | Retained | Retained |
| RAM | | Functioning | Retained | Functioning | Retained | Retained |
| IO ports | | Functioning | Retained | Functioning | Retained | Register contents are retained, but output is the high-impedance state. |
| External interrupts | IRQ3 to IRQ0 | Functioning | Functioning | Functioning | Functioning | Functioning |
| | WKP5 to WKP0 | Functioning | Functioning | Functioning | Functioning | Functioning |
| Peripheral functions | RTC | Functioning | Functioning | Functioning if the timekeeping time-base function is selected, and retained if not selected | | |
| | Timer V | Functioning | Functioning | Reset | Reset | Reset |
| | Watchdog timer | Functioning | Functioning | Retained (functioning if the internal oscillator is selected as a count clock*) | | |
| | SCI3, SCI3_2, SCI3_3 | Functioning | Functioning | Reset | Reset | Reset |
| | IIC2 | Functioning | Functioning | Retained* | Retained | Retained |
| | Timer B1 | Functioning | Functioning | Retained* | Retained | Retained |
| | Timer RD | Functioning | Functioning | Retained (the counter is incremented by a subclock if the internal clock $\phi$ is selected as a count clock*) | | Retained |
| | Timer RC | Functioning | Functioning | | | Retained |
| | 14-bit PWM | Functioning | Functioning | Retained* | Retained | Retained |
| | A/D converter | Functioning | Functioning | Reset | Reset | Reset |

Note: * Registers can be read or written in subactive mode.

### 6.2.1 Sleep Mode

In sleep mode, CPU operation is halted but the on-chip peripheral modules function at the clock frequency set by the MA2, MA1, and MA0 bits in SYSCR2. CPU register contents are retained. When an interrupt is requested, sleep mode is cleared and interrupt exception handling starts. Sleep mode is not cleared if the I bit of the condition code register (CCR) is set to 1 or the requested interrupt is disabled in the interrupt enable register. After sleep mode is cleared, a transition is made to active mode when the LSON bit in SYSCR2 is 0, and a transition is made to subactive mode when the bit is 1. When the $\overline{\text{RES}}$ pin goes low, the CPU goes into the reset state and sleep mode is cleared.

### 6.2.2 Standby Mode

In standby mode, the system clock oscillator stops, so the CPU and on-chip peripheral modules stop functioning. However, as long as the rated voltage is supplied, the contents of CPU registers, on-chip RAM, and some on-chip peripheral module registers are retained. On-chip RAM contents will be retained as long as the voltage set by the RAM data retention voltage is provided. The I/O ports go to the high-impedance state.

The standby mode is cleared by an interrupt. When an interrupt is requested, the system clock oscillator starts. After the time set in bits STS2 to STS0 in SYSCR1 and bit STS3 in SYSCR3 has elapsed, the standby mode is lifted and the interrupt exception handling starts. The standby mode is not lifted if the I bit in CCR is set to 1 or the requested interrupt is disabled in the interrupt enable register.

When the $\overline{\text{RES}}$ signal goes low, the on-chip oscillator starts oscillation. Since clock signals are supplied to the entire chip as soon as the on-chip oscillator starts oscillation, the $\overline{\text{RES}}$ signal must be kept low over a given time. After the given time, the CPU starts the reset exception handling when the $\overline{\text{RES}}$ signal is driven high.

### 6.2.3 Subsleep Mode

In subsleep mode, operation of the CPU and on-chip peripheral modules other than the RTC is halted. As long as a required voltage is applied, the contents of CPU registers, the on-chip RAM, and some registers of the on-chip peripheral modules are retained. I/O ports keep the same states as before the transition.

The subsleep mode is lifted by an interrupt. When an interrupt is requested, the subsleep mode is lifted and the interrupt exception handling starts. The subsleep mode is not lifted if the I bit in CCR is set to 1 or the requested interrupt is disabled in the interrupt enable register. The mode after the subsleep mode is lifted, a transition is made to the active mode or subactive mode according to the LSON bit in SYSCR2 is 0. After the time set in bits STS2 to STS0 in SYSCR1 and bit STS3 in SYSCR has elapsed, a transition is made to active mode.

When the $\overline{\text{RES}}$ signal goes low, the on-chip oscillator starts oscillation. Since clock signals are supplied to the entire chip as soon as the on-chip oscillator starts oscillation, the $\overline{\text{RES}}$ signal must be kept low over a given time. After the given time, the CPU starts the reset exception handling when the $\overline{\text{RES}}$ signal is driven high.

### 6.2.4 Subactive Mode

The operating frequency in subactive mode is selected from $\phi_w/2$, $\phi_w/4$, and $\phi_w/8$ by the SA1 and SA0 bits in SYSCR2. After the SLEEP instruction is executed, the operating frequency changes to the frequency which is set before the execution.

When the SLEEP instruction is executed in subactive mode, a transition to sleep mode, subsleep mode, standby mode, active mode, or subactive mode is made, depending on the combination of SYSCR1 and SYSCR2.

When the $\overline{\text{RES}}$ signal goes low, the on-chip oscillator starts oscillation. Since clock signals are supplied to the entire chip as soon as the on-chip oscillator starts oscillation, the $\overline{\text{RES}}$ signal must be kept low over a given time. After the given time, the CPU starts the reset exception handling when the $\overline{\text{RES}}$ signal is driven high.

## 6.3 Operating Frequency in Active Mode

This LSI operates in active mode at the frequency specified by bits MA2, MA1, and MA0 in SYSCR2. The operating frequency changes to the set frequency after the SLEEP instruction execution.

## 6.4 Direct Transition

The CPU can execute programs in two modes: active and subactive modes. A direct transition is a transition between these two modes without stopping program execution. A direct transition can be made by executing the SLEEP instruction while the DTON bit in SYSCR2 is set to 1. The direct transition also enables operating frequency modification in active or subactive mode. After the mode transition, direct transition interrupt exception handling starts.

If the direct transition interrupt is disabled in interrupt enable register 1, a transition is made instead to sleep or subsleep mode. Note that if a direct transition is attempted while the I bit in CCR is set to 1, sleep or subsleep mode will be entered, and the resulting mode cannot be cleared by means of an interrupt.

### 6.4.1 Direct Transition from Active Mode to Subactive Mode

The time from the start of the SLEEP instruction execution to the end of the interrupt exception handling (the direct transition time) is calculated by equation (1).

Direct transition time = {(number of SLEEP instruction execution cycles) + (number of internal clock cycles)}× (tcyc before transition) + (number of interrupt exception handling cycles) × (tsubcyc after transition) …. (1)

Example 1: Case when the CPU operating clock changes from $\phi_{osc}$ to $\phi_w/8$

$$\text{Direct transition time} = (2 + 1) \times t_{OSC} + 16 \times 8\ t_w = 3\ t_{OSC} + 128\ t_w$$

Example 2: Case when the system clock source is Rosc/4 and the division ratio is 16; the CPU operating clock changes from $\phi/16$ to $\phi_w/2$

$$\text{Direct transition time} = (2 + 1) \times 4\ t_{ROSC} \times 16 + 16 \times 2\ t_w = 192\ t_{ROSC} + 32\ t_w$$

[Legend]

$t_{OSC}$: OSC clock cycle time
$t_{ROSC}$: Period of oscillation of the on-chip oscillator
$t_w$: Watch clock cycle time
$t_{cyc}$: System clock ($\phi$) cycle time
$t_{subcyc}$: Subclock ($\phi_{SUB}$) cycle time

### 6.4.2 Direct Transition from Subactive Mode to Active Mode

The time from the start of the SLEEP instruction execution to the end of the interrupt exception handling (the direct transition time) is calculated by equation (2).

Direct transition time = {(number of SLEEP instruction execution cycles) + (number of internal processing cycles)} × (tsubcyc before transition) + {(waiting time set in bits STS2 to STS0) + (number of interrupt exception handling cycles)} × ($t_{cyc}$ after transition) …. (2)

Example 1: Case when the CPU operating clock changes from $\phi_w/8$ to $\phi_{osc}$, and a waiting time of 32768 cycles is set

$$\text{Direct transition time} = (2 + 1) \times 8\ t_w + (32768 + 16) \times t_{OSC} = 24\ t_w + 32784\ t_{OSC}$$

Example 2: Case when the CPU operating clock changes from $\phi_w/4$ to Rosc/2, and a waiting time of 4096 cycles is set

$$\text{Direct transition time} = (2 + 1) \times 4\ t_w + (4096 + 16) \times t_{ROSC} = 12\ t_w + 8224\ t_{OSC}$$

[Legend]

$t_{OSC}$: OSC clock cycle time
$t_{ROSC}$: Period of oscillation of the on-chip oscillator
$t_w$: Watch clock cycle time
tcyc: System clock ($\phi$) cycle time
tsubcyc: Subclock ($\phi_{SUB}$) cycle time

## 6.5 Module Standby Function

The module-standby function can be set to any peripheral module. In the module standby state, the clock supply to modules stops to enter the power-down mode. Setting a bit in MSTCR1, MSTCR2, MSTCR4, or SMCR that corresponds to each module to 1 enables each on-chip peripheral module to enter the module standby state and the module standby state is canceled by clearing the bit to 0.

# Section 7 ROM

The features of the 128-kbyte flash memory in this LSI are summarized below.

- Programming/erasing methods
  - The flash memory is programmed 128 bytes at a time. Erasure is performed in single-block units. The flash memory is configured as follows: four 1-kbyte blocks, one 28-kbyte block, and three 32-kbyte blocks. To erase the entire flash memory, each block must be erased in turn.
- Reprogramming capability
  - The flash memory can be reprogrammed up to 1,000 times.
- On-board programming
  - On-board programming/erasing can be done in boot mode, in which the boot program built into the chip is started to erase or program of the entire flash memory. In normal user programming mode, individual blocks can be erased or programmed.
- Programmer mode
  - Flash memory can be programmed/erased in programmer mode using a PROM programmer, as well as in on-board programming mode.
- Automatic bit rate adjustment
  - For data transfer in boot mode, this LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Programming/erasing protection
  - Sets software protection against flash memory programming/erasing.
- Power-down mode
  - Operation of the power supply circuit can be partly halted in subactive mode. As a result, flash memory can be read with low power consumption.

## 7.1 Block Configuration

Figure 7.1 shows the block configuration of flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values in the frames are addresses. The 128-kbyte flash memory is divided into four 1-kbyte blocks, one 28-kbyte block, and three 32-kbyte blocks. Erasing is performed in these units.

Programming is performed in 128-byte units, each starting at an address with H'00 or H'80 as the low-order byte.

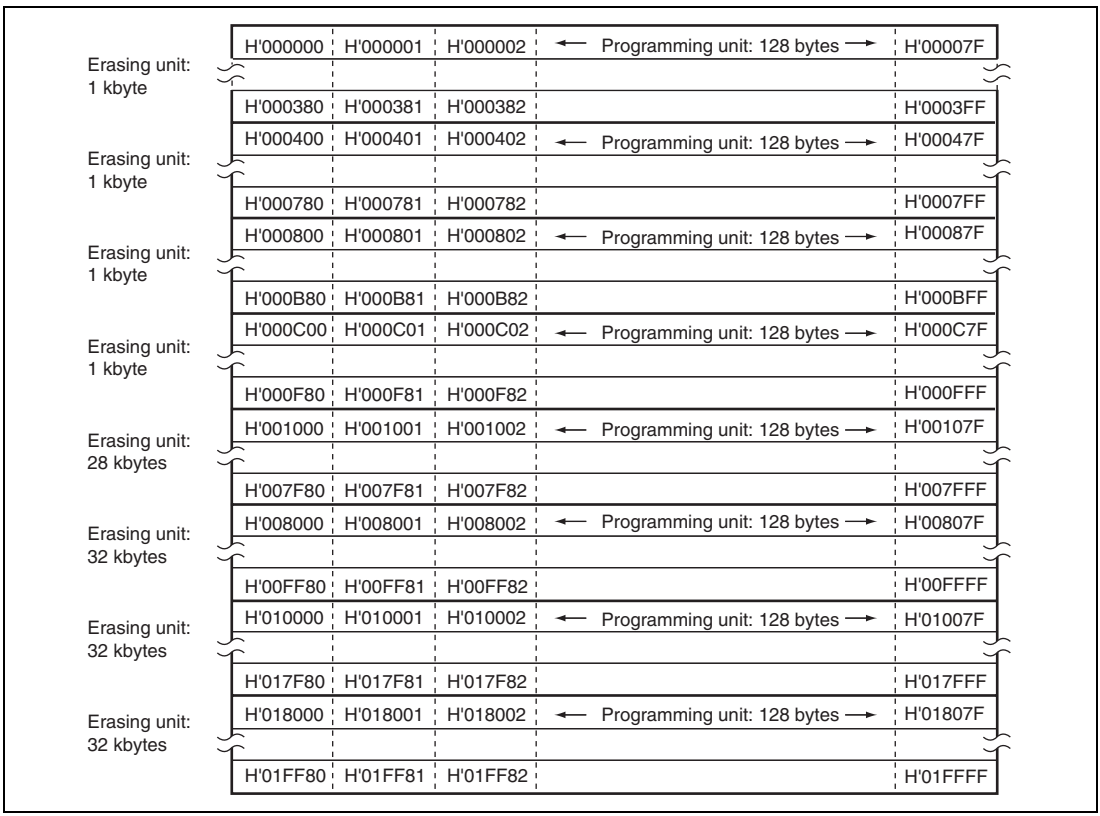



**Figure 7.1 Block Configuration of Flash Memory**

## 7.2 Register Descriptions

The flash memory has the following registers.

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Erase block register 1 (EBR1)
- Flash memory power control register (FLPWCR)
- Flash memory enable register (FENR)

### 7.2.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is a register that makes the flash memory change to programming mode, program-verify mode, erasing mode, or erase-verify mode. For details on register setting, refer to section 7.4, Flash Memory Programming/Erasing.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | — | Reserved<br>This bit is always read as 0. |
| 6 | SWE | 0 | R/W | Software Write Enable<br>When this bit is set to 1, flash memory programming/erasing is enabled. When this bit is cleared to 0, other FLMCR1 register bits and all EBR1 bits cannot be set. |
| 5 | ESU | 0 | R/W | Erasure Setup<br>When this bit is set to 1, the flash memory changes to the erasure setup state. When it is cleared to 0, the erasure setup state is cancelled. Set this bit to 1 before setting the E bit to 1 in FLMCR1. |
| 4 | PSU | 0 | R/W | Program Setup<br>When this bit is set to 1, the flash memory changes to the program setup state. When it is cleared to 0, the program setup state is cancelled. Set this bit to 1 before setting the P bit in FLMCR1. |
| 3 | EV | 0 | R/W | Erase-Verify<br>When this bit is set to 1, the flash memory changes to erase-verify mode. When it is cleared to 0, erase-verify mode is cancelled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | PV | 0 | R/W | Program-Verify<br>When this bit is set to 1, the flash memory changes to program-verify mode. When it is cleared to 0, program-verify mode is cancelled. |
| 1 | E | 0 | R/W | Erasure<br>When this bit is set to 1 while SWE=1 and ESU=1, the flash memory changes to erasing mode. When it is cleared to 0, erasing mode is cancelled. |
| 0 | P | 0 | R/W | Program<br>When this bit is set to 1 while SWE=1 and PSU=1, the flash memory changes to programming mode. When it is cleared to 0, programming mode is cancelled. |

### 7.2.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is a register that displays the state of flash memory programming/erasing. FLMCR2 is a read-only register, and should not be written to.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | FLER | 0 | R | Flash Memory Error<br>Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.<br>See section 7.5.3, Error Protection, for details. |
| 6 to 0 | — | All 0 | — | Reserved<br>These bits are always read as 0. |

### 7.2.3 Erase Block Register 1 (EBR1)

EBR1 specifies whether or not a block in the flash memory is erased. EBR1 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Do not set more than one bit at a time, as this will cause all the bits in EBR1 to be automatically cleared to 0.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | EB7 | 0 | R/W | When this bit is set to 1, 32 kbytes of H'018000 to H'01FFFF will be erased. |
| 6 | EB6 | 0 | R/W | When this bit is set to 1, 32 kbytes of H'010000 to H'017FFF will be erased. |
| 5 | EB5 | 0 | R/W | When this bit is set to 1, 32 kbytes of H'008000 to H'00FFFF will be erased. |
| 4 | EB4 | 0 | R/W | When this bit is set to 1, 28 kbytes of H'001000 to H'007FFF will be erased. |
| 3 | EB3 | 0 | R/W | When this bit is set to 1, 1 kbyte of H'000C00 to H'000FFF will be erased. |
| 2 | EB2 | 0 | R/W | When this bit is set to 1, 1 kbyte of H'000800 to H'000BFF will be erased. |
| 1 | EB1 | 0 | R/W | When this bit is set to 1, 1 kbyte of H'000400 to H'0007FF will be erased. |
| 0 | EB0 | 0 | R/W | When this bit is set to 1, 1 kbyte of H'000000 to H'0003FF will be erased. |

### 7.2.4 Flash Memory Power Control Register (FLPWCR)

FLPWCR enables or disables a transition to the flash memory power-down mode when the LSI switches to subactive mode. There are two modes: mode in which operation of the power supply circuit of flash memory is partly halted in power-down mode and flash memory can be read, and mode in which even if a transition is made to subactive mode, operation of the power supply circuit of flash memory is retained and flash memory can be read.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PDWND | 0 | R/W | Power-Down Disable<br>When this bit is 0 and a transition is made to subactive mode, the flash memory enters the power-down mode. When this bit is 1, the flash memory remains in the normal mode even after a transition is made to subactive mode. |
| 6 to 0 | — | All 0 | — | Reserved<br>These bits are always read as 0. |

### 7.2.5 Flash Memory Enable Register (FENR)

Bit 7 (FLSHE) in FENR enables or disables the CPU access to the flash memory control registers, FLMCR1, FLMCR2, EBR1, and FLPWCR.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | FLSHE | 0 | R/W | Flash Memory Control Register Enable<br>Flash memory control registers can be accessed when this bit is set to 1. Flash memory control registers cannot be accessed when this bit is set to 0. |
| 6 | — | 0 | R/W | Reserved<br>This bit can be read from or written to, but should not be set to 1. |
| 5 to 0 | — | All 0 | — | Reserved<br>These bits are always read as 0. |

## 7.3 On-Board Programming Modes

There are two modes for programming/erasing of the flash memory; boot mode, which enables on-board programming/erasing, and programmer mode, in which programming/erasing is performed with a PROM programmer. On-board programming/erasing can also be performed in user programming mode. At reset-start in reset mode, this LSI changes to a mode depending on the TEST pin settings, $\overline{\text{NMI}}$ pin settings, and input level of each port, as shown in table 7.1. The input level of each pin must be defined four states before the reset ends.

When changing to boot mode, the boot program built into this LSI is initiated. The boot program transfers the programming control program from the externally-connected host to on-chip RAM via SCI3. After erasing the entire flash memory, the programming control program is executed. This can be used for programming initial values in the on-board state or for a forcible return when programming/erasing can no longer be done in user programming mode. In user programming mode, individual blocks can be erased and programmed by branching to the user programming/erasure control program prepared by the user.

**Table 7.1 Setting Programming Modes**

| TEST | $\overline{\text{NMI}}$ | P85 | PC0 | PC1 | PC2 | LSI State after Reset End |
|---|---|---|---|---|---|---|
| 0 | 1 | X | X | X | X | User Mode |
| 0 | 0 | 1 | X | X | X | Boot Mode |
| 1 | X | X | 0 | 0 | 0 | Programmer Mode |

[Legend]
X : Don't care.

### 7.3.1 Boot Mode

Table 7.2 shows the boot mode operations between reset end and branching to the programming control program.

1. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 7.4, Flash Memory Programming/Erasing.
2. SCI3 should be set to asynchronous mode, and the transfer format as follows: 8-bit data, 1 stop bit, and no parity.

3. When the boot program is initiated, the chip measures the low-level period of asynchronous SCI communication data (H'00) transmitted continuously from the host. The chip then calculates the bit rate of transmission from the host, and adjusts the SCI3 bit rate to match that of the host. The reset should end with the RxD pin high. The RxD and TxD pins should be pulled up on the board if necessary. After the reset is complete, it takes approximately 100 states before the chip is ready to measure the low-level period.
4. After matching the bit rates, the chip transmits one H'00 byte to the host to indicate the completion of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the chip. If reception could not be performed normally, initiate boot mode again by a reset. Depending on the host's transfer bit rate and system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and the chip. To operate the SCI properly, set the host's transfer bit rate and system clock frequency of this LSI within the ranges listed in table 7.3.
5. In boot mode, a part of the on-chip RAM area is used by the boot program. The area H'FFF780 to H'FFFEEF is the area to which the programming control program is transferred from the host. The boot program area cannot be used until the execution state in boot mode switches to the programming control program.
6. Before branching to the programming control program, the chip terminates transfer operations by SCI3 (by clearing the RE and TE bits in SCR to 0), however the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transfer of program data or verify data with the host. The TxD pin is high (PCR22 = 1, P22 = 1). The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program, as the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
7. Boot mode can be cleared by a reset. End the reset after driving the reset pin low, waiting at least 20 states, and then setting the TEST pin and $\overline{\text{NMI}}$ pin. Boot mode is also cleared when a WDT overflow occurs.
8. Do not change the TEST pin and $\overline{\text{NMI}}$ pin input levels in boot mode.

## Table 7.2 Boot Mode Operation

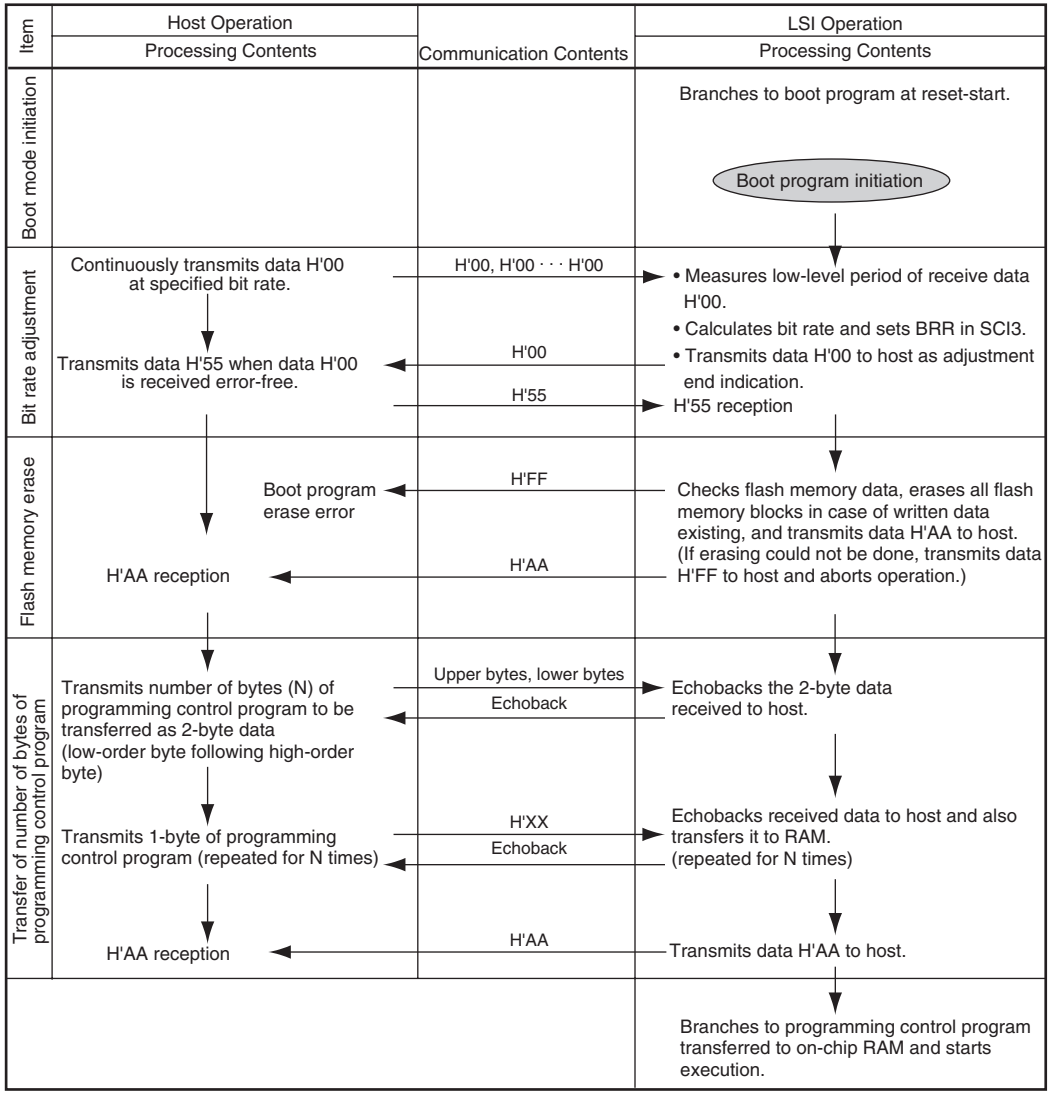


**Table 7.3 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible**

| Host Bit Rate | System Clock Frequency Range of LSI |
|---|---|
| 9,600 bps | On-chip oscillator (10 MHz) |
| 4,800 bps | |
| 2,400 bps | |

### 7.3.2 Programming/Erasing in User Programming Mode

On-board programming/erasing of an individual flash memory block can also be performed in user programming mode by branching to a user programming/erasure control program. The user must set branching conditions and provide on-board means of supplying programming data. The flash memory must contain the user programming/erasure control program or a program that provides the user programming/erasure control program from external memory. As the flash memory itself cannot be read during programming/erasing, transfer the user programming/erasure control program to on-chip RAM, as in boot mode. Figure 7.2 shows a sample procedure for programming/erasing in user programming mode. Prepare a user programming/erasure control program in accordance with the description in section 7.4, Flash Memory Programming/Erasing.

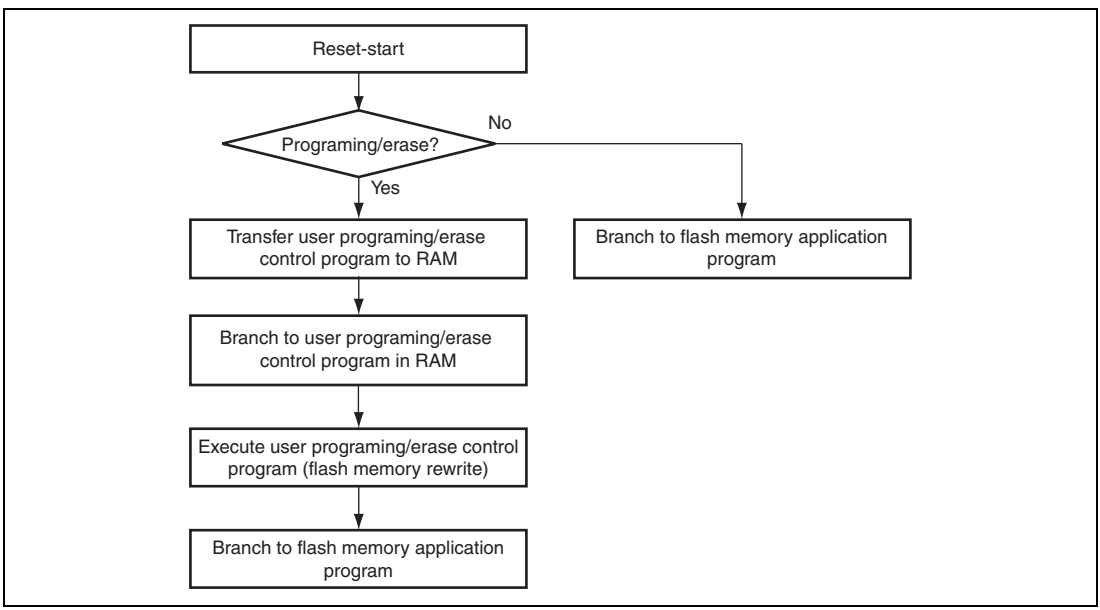



**Figure 7.2 Programming/Erasing Flowchart Example in User Programming Mode**

## 7.4 Flash Memory Programming/Erasing

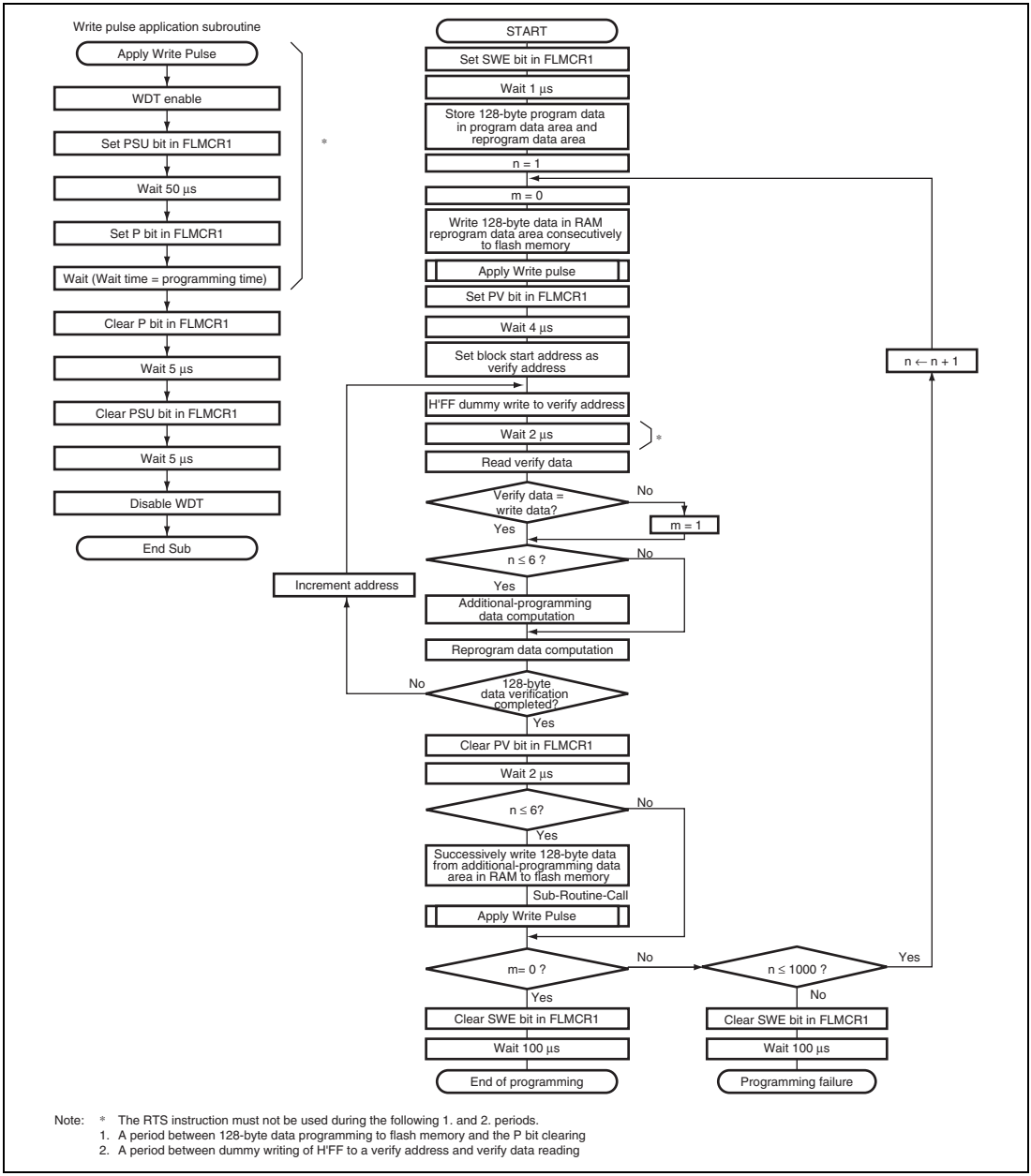
A software method using the CPU is employed to program and erase flash memory in the on-board programming modes. Depending on the FLMCR1 setting, the flash memory operates in one of the following four modes: Programming mode, program-verify mode, erasing mode, and erase-verify mode. The programming control program in boot mode and the user programming/erasure control program in user programming mode use these operating modes in combination to perform programming/erasing. Flash memory programming and erasing should be performed in accordance with the descriptions in section 7.4.1, Programming/Program-Verify and section 7.4.2, Erasure/Erase-Verify, respectively.

### 7.4.1 Programming/Program-Verify

When writing data or programs to the flash memory, the programming/program-verify flowchart shown in figure 7.3 should be followed. Performing programming operations according to this flowchart will enable data or programs to be written to the flash memory without subjecting the chip to voltage stress or sacrificing program data reliability.

1. Programming must be done to an empty address. Do not reprogram an address to which programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer must be performed even if writing fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. Prepare the following data storage areas in RAM: A 128-byte programming data area, a 128-byte reprogramming data area, and a 128-byte additional-programming data area. Perform reprogramming data computation according to table 7.4, and additional programming data computation according to table 7.5.
4. Consecutively transfer 128 bytes of data in byte units from the reprogramming data area or additional-programming data area to the flash memory. The program address and 128-byte data are latched in the flash memory. The lower 8 bits of the start address in the flash memory destination area must be H'00 or H'80.
5. The time during which the P bit is set to 1 is the programming time. Table 7.6 shows the allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. An overflow cycle of approximately 6.6 ms is allowed.
7. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower 2 bits are B'00. Verify data can be read in words or in longwords from the address to which a dummy write was performed.

8. The maximum number of repetitions of the programming/program-verify sequence of the same bit is 1,000.




Note: * The RTS instruction must not be used during the following 1. and 2. periods.
1. A period between 128-byte data programming to flash memory and the P bit clearing
2. A period between dummy writing of H'FF to a verify address and verify data reading

**Figure 7.3 Programming/Program-Verify Flowchart**

**Table 7.4 Reprogramming Data Computation Table**

| Programming Data | Verify Data | Reprogramming Data | Comments |
|---|---|---|---|
| 0 | 0 | 1 | Programming completed |
| 0 | 1 | 0 | Reprogramming bit |
| 1 | 0 | 1 | — |
| 1 | 1 | 1 | Remains in erased state |

**Table 7.5 Additional-Program Data Computation Table**

| Reprogramming Data | Verify Data | Additional-Program Data | Comments |
|---|---|---|---|
| 0 | 0 | 0 | Additional-program bit |
| 0 | 1 | 1 | No additional programming |
| 1 | 0 | 1 | No additional programming |
| 1 | 1 | 1 | No additional programming |

**Table 7.6 Programming Time**

| n (Number of Writes) | Programming Time | In Additional Programming | Comments |
|---|---|---|---|
| 1 to 6 | 30 | 10 | |
| 7 to 1,000 | 200 | — | |

Note: Time shown in μs.

### 7.4.2 Erasure/Erase-Verify

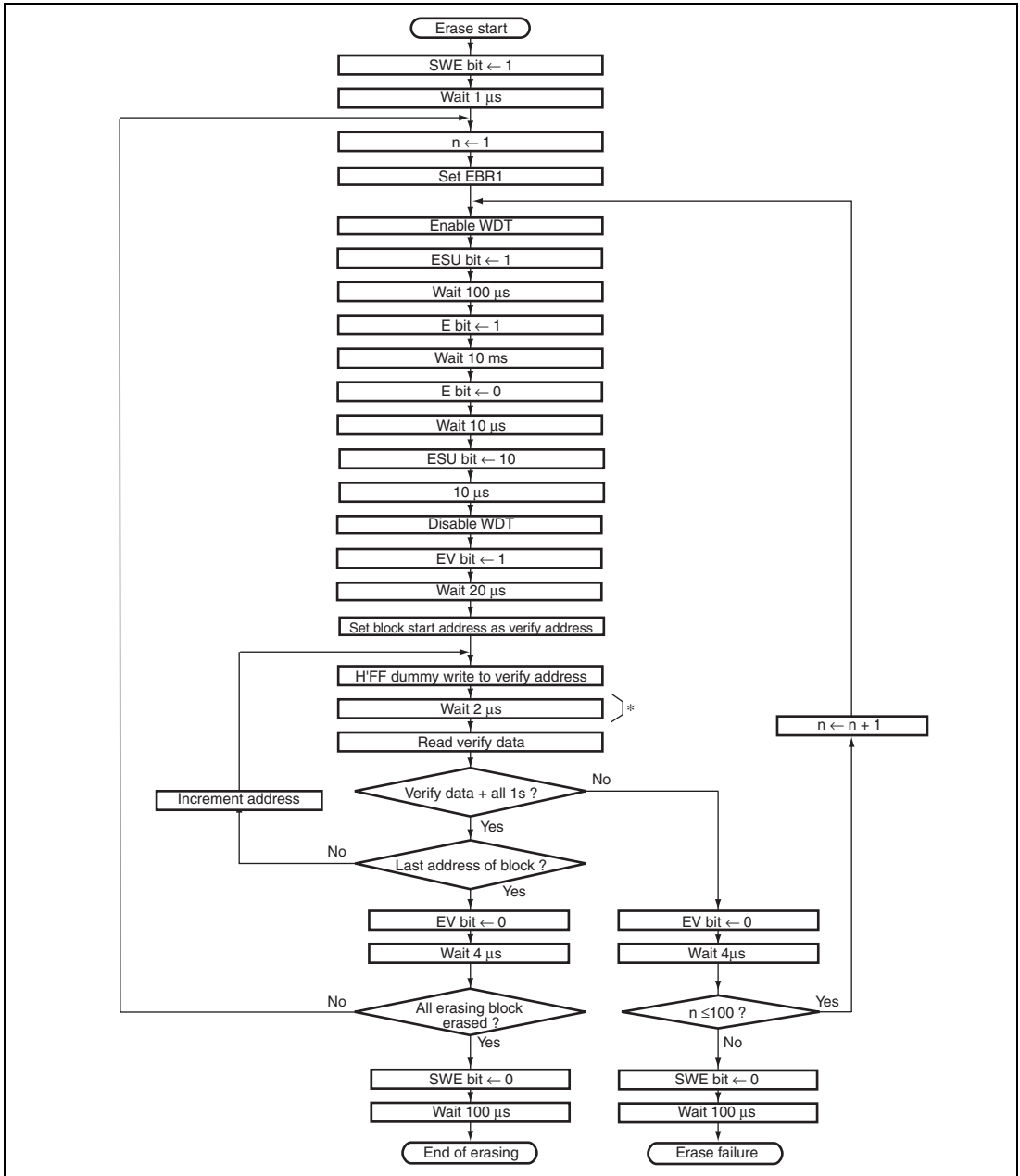
When erasing flash memory, the erasure/erase-verify flowchart shown in figure 7.4 should be followed.

1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Erasing is performed in block units. Make only a single-bit specification in the erase block register (EBR1). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. An overflow cycle of approximately 19.8 ms is allowed.
5. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower two bits are B'00. Verify data can be read in longwords from the address to which a dummy write was performed.
6. If the read data is not erased successfully, set erasing mode again, and repeat the erase/erase-verify sequence as before. The maximum number of repetitions of the erase/erase-verify sequence is 100.

### 7.4.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including the NMI interrupt, are disabled while flash memory is being programmed or erased, or while the boot program is executing, for the following three reasons:

1. Interrupt during programming/erasing may cause a violation of the programming or erasing algorithm, with the result that normal operation cannot be assured.
2. If interrupt exception handling starts before the vector address is written or during programming/erasing, a correct vector cannot be fetched and the CPU malfunctions.
3. If an interrupt occurs during boot program execution, normal boot mode sequence cannot be carried out.




Note: * The RTS instruction must not be used during a period between dummy writing of H'FF to a verify address and verify data reading.

**Figure 7.4 Erasure/Erase-Verify Flowchart**

## 7.5 Programming/Erasing Protection

There are three types of flash memory programming/erasing protection; hardware protection, software protection, and error protection.

### 7.5.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted because of a transition to reset, subactive mode, subsleep mode, or standby mode. Flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), and erase block register 1 (EBR1) are initialized. In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.

### 7.5.2 Software Protection

Software protection can be implemented against programming/erasing of all flash memory blocks by clearing the SWE bit in FLMCR1. When software protection is in effect, setting the P or E bit in FLMCR1 does not cause a transition to programming mode or erasing mode. By setting the erase block register 1 (EBR1), erase protection can be set for individual blocks. When EBR1 is set to H'00, erase protection is set for all blocks.

### 7.5.3 Error Protection

In error protection, an error is detected when CPU runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the programming/erasing algorithm, and the programming/erasing operation is forcibly aborted. Aborting the programming/erasing operation prevents damage to the flash memory due to overprogramming or overerasing.

When the following errors are detected during programming/erasing of flash memory, the FLER bit in FLMCR2 is set to 1, and the error protection state is entered.

- When the flash memory of the relevant address area is read during programming/erasing (including vector read and instruction fetch)
- Immediately after exception handling excluding a reset during programming/erasing
- When a SLEEP instruction is executed during programming/erasing

The FLMCR1, FLMCR2, and EBR1 settings are retained, however programming mode or erasing mode is aborted at the point at which the error occurred. Programming mode or erasing mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit settings are retained, and a transition can be made to verify mode. Error protection can be cleared only by a power-on reset.

## 7.6 Programmer Mode

In programmer mode, a PROM programmer can be used to perform programming/erasing via a socket adapter, just as a discrete flash memory. Use a PROM programmer that supports the MCU device type with the on-chip Renesas Electronics 128-kbyte flash memory.

## 7.7 Power-Down States for Flash Memory

In user mode, the flash memory will operate in either of the following states:

- Normal operating mode
  The flash memory can be read and written to at high speed.
- Power-down operating mode
  The power supply circuit of flash memory can be partly halted. As a result, flash memory can be read with low power consumption.
- Standby mode
  All flash memory circuits are halted.

Table 7.7 shows the correspondence between the operating modes of this LSI and the flash memory. In subactive mode, the flash memory can be set to operate in power-down mode with the PDWND bit in FLPWCR. When the flash memory returns to its normal operating state from power-down mode or standby mode, a period to stabilize operation of the power supply circuits that were stopped is needed. When the flash memory returns to its normal operating state, bits STS2 to STS0 in SYSCR1 and bit STS3 in SYSCR3 must be set so that the waiting time is100 µs or more, even when the external clock is being used.

**Table 7.7 Flash Memory Operating States**

| | Flash Memory Operating State | |
|---|---|---|
| **LSI Operating State** | **PDWND = 0 (Initial Value)** | **PDWND = 1** |
| Active mode | Normal operating mode | Normal operating mode |
| Subactive mode | Power-down mode | Normal operating mode |
| Sleep mode | Normal operating mode | Normal operating mode |
| Subsleep mode | Standby mode | Standby mode |
| Standby mode | Standby mode | Standby mode |

# Section 8 RAM

This LSI has an on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling the CPU to access both byte data and word data in two states.

| Product Classification | | RAM Size | RAM Address |
|---|---|---|---|
| Flash memory version (F-ZTAT version) | H8/36109F | 5 kbytes | H'FFE400 to H'FFEFFF, H'FFF780 to H'FFFF7F* |

Note: * When the E7 is used, the area from H'FFF780 to H'FFFB7F must not be accessed.

# Section 9 I/O Ports

This LSI has seventy-nine general I/O ports and eight general input-only ports. Twenty ports are large current ports, which can drive 20 mA ($@V_{OL}$ = 1.5 V) when a low level signal is output. Any of these ports can become an input port immediately after a reset. They can also be used as I/O pins of the on-chip peripheral modules or external interrupt input pins, and these functions can be switched depending on the register settings. The registers for selecting these functions can be divided into two types: those included in I/O ports and those included in each on-chip peripheral module. General I/O ports are comprised of the port control register for controlling inputs/outputs and the port data register for storing output data and can select inputs/outputs in bit units.

For functions in each port, see appendix B.1, I/O Port Block Diagrams. For the execution of bit-manipulation instructions to the port control register and port data register, see section 2.8.3, Bit Manipulation Instruction.

## 9.1 Port 1

Port 1 is a general I/O port also functioning as IRQ interrupt input pins, an RTC output pin, a 14-bit PWM output pin, a timer B1 input pin, and a timer V input pin. Figure 9.1 shows its pin configuration.

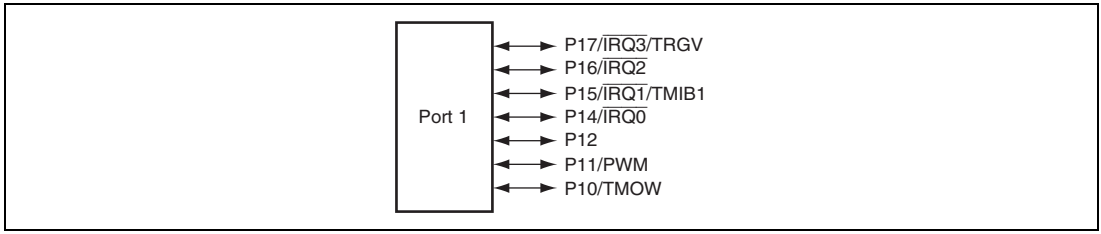



**Figure 9.1 Port 1 Pin Configuration**

Port 1 has the following registers.

- Port mode register 1 (PMR1)
- Port control register 1 (PCR1)
- Port data register 1 (PDR1)
- Port pull-up control register 1 (PUCR1)

### 9.1.1 Port Mode Register 1 (PMR1)

PMR1 switches functions of pins in port 1 and port 2.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | IRQ3 | 0 | R/W | Selects the function of pin P17/$\overline{\text{IRQ3}}$/TRGV.<br>0: General I/O port<br>1: $\overline{\text{IRQ3}}$/TRGV input pin |
| 6 | IRQ2 | 0 | R/W | Selects the function of pin P16/$\overline{\text{IRQ2}}$.<br>0: General I/O port<br>1: $\overline{\text{IRQ2}}$ input pin |
| 5 | IRQ1 | 0 | R/W | Selects the function of pin P15/$\overline{\text{IRQ1}}$/TMIB1.<br>0: General I/O port<br>1: $\overline{\text{IRQ1}}$/TMIB1 input pin |
| 4 | IRQ0 | 0 | R/W | Selects the function of pin P14/$\overline{\text{IRQ0}}$.<br>0: General I/O port<br>1: $\overline{\text{IRQ0}}$ input pin |
| 3 | TXD2 | 0 | R/W | Selects the function of pin P72/TXD_2.<br>0: General I/O port<br>1: TXD_2 output pin |
| 2 | PWM | 0 | R/W | Selects the function of pin P11/PWM.<br>0: General I/O port<br>1: PWM output pin |
| 1 | TXD | 0 | R/W | Selects the function of pin P22/TXD.<br>0: General I/O port<br>1: TXD output pin |
| 0 | TMOW | 0 | R/W | Selects the function of pin P10/TMOW.<br>0: General I/O port<br>1: TMOW output pin |

### 9.1.2 Port Control Register 1 (PCR1)

PCR1 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 1.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PCR17 | 0 | W | When the corresponding pin is designated in PMR1 as a general I/O pin, setting a PCR1 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.<br>Bit 3 is a reserved bit. |
| 6 | PCR16 | 0 | W | |
| 5 | PCR15 | 0 | W | |
| 4 | PCR14 | 0 | W | |
| 3 | — | — | — | |
| 2 | PCR12 | 0 | W | |
| 1 | PCR11 | 0 | W | |
| 0 | PCR10 | 0 | W | |

### 9.1.3 Port Data Register 1 (PDR1)

PDR1 is a general I/O port data register of port 1.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | P17 | 0 | R/W | PDR1 stores output data for port 1 pins. |
| 6 | P16 | 0 | R/W | If PDR1 is read while PCR1 bits are set to 1, the values stored in PDR1 are read. If PDR1 is read while PCR1 bits are cleared to 0, the pin states are read regardless of the value stored in PDR1.<br>Bit 3 is a reserved bit. This bit is always read as 1. |
| 5 | P15 | 0 | R/W | |
| 4 | P14 | 0 | R/W | |
| 3 | — | 1 | — | |
| 2 | P12 | 0 | R/W | |
| 1 | P11 | 0 | R/W | |
| 0 | P10 | 0 | R/W | |

### 9.1.4 Port Pull-Up Control Register 1 (PUCR1)

PUCR1 controls the pull-up MOS in bit units of the pins set as the input ports.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PUCR17 | 0 | R/W | Only bits for which PCR1 is cleared are valid. The pull-up MOSs of P17 to P14 and P12 to P10 pins enter the on-state when these bits are set to 1, while they enter the off-state when these bits are cleared to 0.<br>Bit 3 is a reserved bit. This bit is always read as 1. |
| 6 | PUCR16 | 0 | R/W | |
| 5 | PUCR15 | 0 | R/W | |
| 4 | PUCR14 | 0 | R/W | |
| 3 | — | 1 | — | |
| 2 | PUCR12 | 0 | R/W | |
| 1 | PUCR11 | 0 | R/W | |
| 0 | PUCR10 | 0 | R/W | |

### 9.1.5 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P17/$\overline{\text{IRQ3}}$/TRGV pin

| Register | PMR1 | PCR1 | |
|---|---|---|---|
| **Bit Name** | **IRQ3** | **PCR17** | **Pin Function** |
| Setting value | 0 | 0 | P17 input pin |
| | | 1 | P17 output pin |
| | 1 | X | $\overline{\text{IRQ3}}$ input/TRGV input pin |

[Legend] X: Don't care.

- P16/$\overline{\text{IRQ2}}$ pin

| Register | PMR1 | PCR1 | |
|---|---|---|---|
| **Bit Name** | **IRQ2** | **PCR16** | **Pin Function** |
| Setting value | 0 | 0 | P16 input pin |
| | | 1 | P16 output pin |
| | 1 | X | $\overline{\text{IRQ2}}$ input pin |

[Legend] X: Don't care.

- P15/$\overline{\text{IRQ1}}$/TMIB1 pin

| Register | PMR1 | PCR1 | |
|---|---|---|---|
| **Bit Name** | **IRQ1** | **PCR15** | **Pin Function** |
| Setting value | 0 | 0 | P15 input pin |
| | | 1 | P15 output pin |
| | 1 | X | $\overline{\text{IRQ1}}$ input/TMIB1 input pin |

[Legend] X: Don't care.

- P14/$\overline{\text{IRQ0}}$ pin

| Register | PMR1 | PCR1 | |
|---|---|---|---|
| **Bit Name** | **IRQ0** | **PCR14** | **Pin Function** |
| Setting value | 0 | 0 | P14 input pin |
| | | 1 | P14 output pin |
| | 1 | X | $\overline{\text{IRQ0}}$ input pin |

[Legend] X: Don't care.

- P12 pin

| Register | PCR1 | |
|---|---|---|
| **Bit Name** | **PCR12** | **Pin Function** |
| Setting value | 0 | P12 input pin |
| | 1 | P12 output pin |

- P11/PWM pin

| Register | PMR1 | PCR1 | |
|---|---|---|---|
| **Bit Name** | **PWM** | **PCR11** | **Pin Function** |
| Setting value | 0 | 0 | P11 input pin |
| | | 1 | P11 output pin |
| | 1 | X | PWM output pin |

[Legend] X: Don't care.

- P10/TMOW pin

| Register | PMR1 | PCR1 | |
|---|---|---|---|
| **Bit Name** | **TMOW** | **PCR10** | **Pin Function** |
| Setting value | 0 | 0 | P10 input pin |
| | | 1 | P10 output pin |
| | 1 | X | TMOW output pin |

[Legend] X: Don't care.

## 9.2 Port 2

Port 2 is a general I/O port also functioning as SCI3 I/O pins. Each pin of port 2 is shown in figure 9.2. The register settings of PMR1 and SCI3 have priority for functions of the pins for both uses.

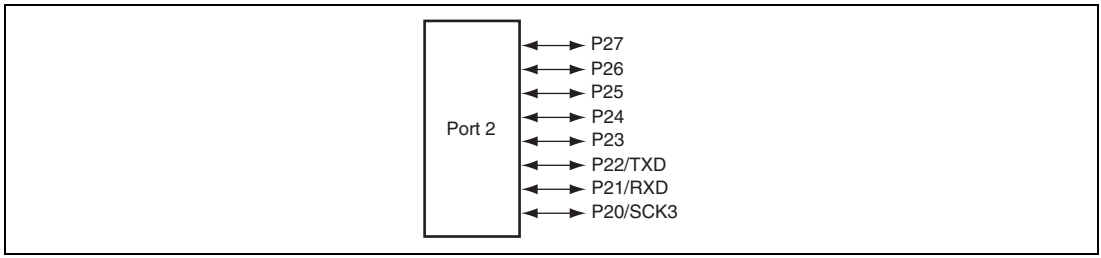



**Figure 9.2 Port 2 Pin Configuration**

Port 2 has the following registers.

- Port control register 2 (PCR2)
- Port data register 2 (PDR2)
- Port mode register 3 (PMR3)

### 9.2.1 Port Control Register 2 (PCR2)

PCR2 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 2.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PCR27 | 0 | W | When each of the port 2 pins P24 to P20 functions as a general I/O port, setting a PCR2 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. |
| 6 | PCR26 | 0 | W | |
| 5 | PCR25 | 0 | W | |
| 4 | PCR24 | 0 | W | |
| 3 | PCR23 | 0 | W | |
| 2 | PCR22 | 0 | W | |
| 1 | PCR21 | 0 | W | |
| 0 | PCR20 | 0 | W | |

### 9.2.2 Port Data Register 2 (PDR2)

PDR2 is a general I/O port data register of port 2.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | P27 | 0 | R/W | PDR2 stores output data for port 2 pins. |
| 6 | P26 | 0 | R/W | If PDR2 is read while PCR2 bits are set to 1, the values stored in PDR2 are read. If PDR2 is read while PCR2 bits are cleared to 0, the pin states are read regardless of the value stored in PDR2. |
| 5 | P25 | 0 | R/W | |
| 4 | P24 | 0 | R/W | |
| 3 | P23 | 0 | R/W | |
| 2 | P22 | 0 | R/W | |
| 1 | P21 | 0 | R/W | |
| 0 | P20 | 0 | R/W | |

### 9.2.3 Port Mode Register 3 (PMR3)

PMR3 selects the CMOS output or NMOS open-drain output for port 2.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | POF27 | 0 | R/W | When the bit is set to 1, the corresponding pin is cut off by PMOS and it functions as the NMOS open-drain output. When cleared to 0, the pin functions as the CMOS output. |
| 6 | POF26 | 0 | R/W | |
| 5 | POF25 | 0 | R/W | |
| 4 | POF24 | 0 | R/W | |
| 3 | POF23 | 0 | R/W | |
| 2 to 0 | — | All 1 | — | Reserved<br>These bits are always read as 1. |

### 9.2.4 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P27 pin

| Register | PCR2 | |
|---|---|---|
| **Bit Name** | **PCR27** | **Pin Function** |
| Setting Value | 0 | P27 input pin |
| | 1 | P27 output pin |

- P26 pin

| Register | PCR2 | |
|---|---|---|
| **Bit Name** | **PCR26** | **Pin Function** |
| Setting Value | 0 | P26 input pin |
| | 1 | P26 output pin |

- P25 pin

| Register | PCR2 | |
|---|---|---|
| **Bit Name** | **PCR25** | **Pin Function** |
| Setting Value | 0 | P25 input pin |
| | 1 | P25 output pin |

- P24 pin

| Register | PCR2 | |
|---|---|---|
| **Bit Name** | **PCR24** | **Pin Function** |
| Setting Value | 0 | P24 input pin |
| | 1 | P24 output pin |

- P23 pin

| Register | PCR2 | |
|---|---|---|
| **Bit Name** | **PCR23** | **Pin Function** |
| Setting Value | 0 | P23 input pin |
| | 1 | P23 output pin |

- P22/TXD pin

| Register | PMR1 | PCR2 | |
|---|---|---|---|
| **Bit Name** | **TXD** | **PCR22** | **Pin Function** |
| Setting Value | 0 | 0 | P22 input pin |
| | | 1 | P22 output pin |
| | 1 | X | TXD output pin |

[Legend] X: Don't care.

- P21/RXD pin

| Register | SCR3 | PCR2 | |
|---|---|---|---|
| **Bit Name** | **RE** | **PCR21** | **Pin Function** |
| Setting Value | 0 | 0 | P21 input pin |
| | | 1 | P21 output pin |
| | 1 | X | RXD input pin |

[Legend] X: Don't care.

- P20/SCK3 pin

| Register | SCR3 | | SMR | PCR2 | |
|---|---|---|---|---|---|
| **Bit Name** | **CKE1** | **CKE0** | **COM** | **PCR20** | **Pin Function** |
| Setting Value | 0 | 0 | 0 | 0 | P20 input pin |
| | | | | 1 | P20 output pin |
| | 0 | 0 | 1 | X | SCK3 output pin |
| | 0 | 1 | X | X | SCK3 output pin |
| | 1 | X | X | X | SCK3 input pin |

[Legend] X: Don't care.

## 9.3 Port 3

Port 3 is a general I/O port. Each pin of port 3 is shown in figure 9.3.

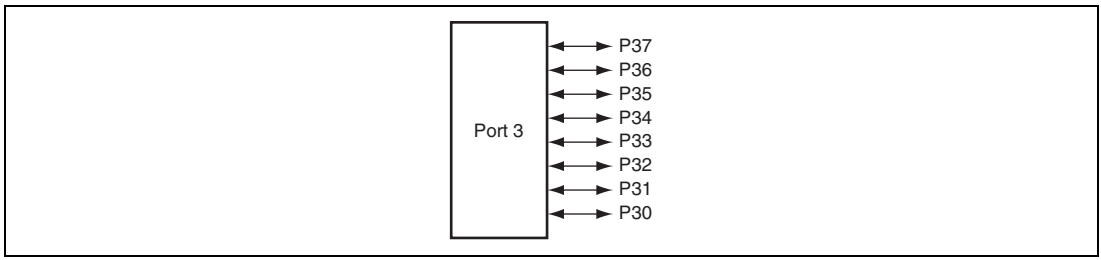



**Figure 9.3 Port 3 Pin Configuration**

Port 3 has the following registers.

- Port control register 3 (PCR3)
- Port data register 3 (PDR3)

### 9.3.1 Port Control Register 3 (PCR3)

PCR3 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 3.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PCR37 | 0 | W | Setting a PCR3 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. |
| 6 | PCR36 | 0 | W | |
| 5 | PCR35 | 0 | W | |
| 4 | PCR34 | 0 | W | |
| 3 | PCR33 | 0 | W | |
| 2 | PCR32 | 0 | W | |
| 1 | PCR31 | 0 | W | |
| 0 | PCR30 | 0 | W | |

### 9.3.2 Port Data Register 3 (PDR3)

PDR3 is a general I/O port data register of port 3.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | P37 | 0 | R/W | PDR3 stores output data for port 3 pins. |
| 6 | P36 | 0 | R/W | If PDR3 is read while PCR3 bits are set to 1, the values stored in PDR3 are read. If PDR3 is read while PCR3 bits are cleared to 0, the pin states are read regardless of the value stored in PDR3. |
| 5 | P35 | 0 | R/W | |
| 4 | P34 | 0 | R/W | |
| 3 | P33 | 0 | R/W | |
| 2 | P32 | 0 | R/W | |
| 1 | P31 | 0 | R/W | |
| 0 | P30 | 0 | R/W | |

### 9.3.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P37 pin

| Register | PCR3 | |
|---|---|---|
| Bit Name | PCR37 | Pin Function |
| Setting Value | 0 | P37 input pin |
| | 1 | P37 output pin |

- P36 pin

| Register | PCR3 | |
|---|---|---|
| Bit Name | PCR36 | Pin Function |
| Setting Value | 0 | P36 input pin |
| | 1 | P36 output pin |

- P35 pin

| Register | PCR3 | |
|---|---|---|
| **Bit Name** | **PCR35** | **Pin Function** |
| Setting Value | 0 | P35 input pin |
| | 1 | P35 output pin |

- P34 pin

| Register | PCR3 | |
|---|---|---|
| **Bit Name** | **PCR34** | **Pin Function** |
| Setting Value | 0 | P34 input pin |
| | 1 | P34 output pin |

- P33 pin

| Register | PCR3 | |
|---|---|---|
| **Bit Name** | **PCR33** | **Pin Function** |
| Setting Value | 0 | P33 input pin |
| | 1 | P33 output pin |

- P32 pin

| Register | PCR3 | |
|---|---|---|
| **Bit Name** | **PCR32** | **Pin Function** |
| Setting Value | 0 | P32 input pin |
| | 1 | P32 output pin |

- P31 pin

| Register | PCR3 | |
|---|---|---|
| **Bit Name** | **PCR31** | **Pin Function** |
| Setting Value | 0 | P31 input pin |
| | 1 | P31 output pin |

- P30 pin

| Register | PCR3 | |
|---|---|---|
| **Bit Name** | **PCR30** | **Pin Function** |
| Setting Value | 0 | P30 input pin |
| | 1 | P30 output pin |

## 9.4 Port 5

Port 5 is a general I/O port also functioning as an $I^2C$ bus interface I/O pin and a wakeup interrupt input pin. Each pin of port 5 is shown in figure 9.4. The register setting of the $I^2C$ bus interface has priority for functions of the pins P57/SCL and P56/SDA. Since the output buffer for pins P56 and P57 has the NMOS push-pull structure, it differs from an output buffer with the CMOS structure in the high-level output characteristics (see section 23, Electrical Characteristics).

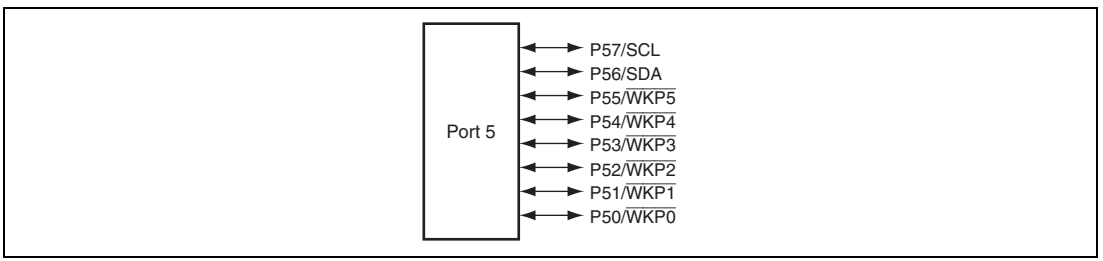



**Figure 9.4 Port 5 Pin Configuration**

Port 5 has the following registers.

- Port mode register 5 (PMR5)
- Port control register 5 (PCR5)
- Port data register 5 (PDR5)
- Port pull-up control register 5 (PUCR5)

### 9.4.1 Port Mode Register 5 (PMR5)

PMR5 switches functions of pins in port 5.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | POF57 | 0 | R/W | When the bit is set to 1, the corresponding pin functions as the NMOS open-drain output. When cleared to 0, the pin functions as the NMOS push-pull output. |
| 6 | POF56 | 0 | R/W | |
| 5 | WKP5 | 0 | R/W | Selects the function of pin P55/$\overline{\text{WKP5}}$.<br>0: General I/O port<br>1: $\overline{\text{WKP5}}$ input pin |
| 4 | WKP4 | 0 | R/W | Selects the function of pin P54/$\overline{\text{WKP4}}$.<br>0: General I/O port<br>1: $\overline{\text{WKP4}}$ input pin |
| 3 | WKP3 | 0 | R/W | Selects the function of pin P53/$\overline{\text{WKP3}}$.<br>0: General I/O port<br>1: $\overline{\text{WKP3}}$ input pin |
| 2 | WKP2 | 0 | R/W | Selects the function of pin P52/$\overline{\text{WKP2}}$.<br>0: General I/O port<br>1: $\overline{\text{WKP2}}$ input pin |
| 1 | WKP1 | 0 | R/W | Selects the function of pin P51/$\overline{\text{WKP1}}$.<br>0: General I/O port<br>1: $\overline{\text{WKP1}}$ input pin |
| 0 | WKP0 | 0 | R/W | Selects the function of pin P50/$\overline{\text{WKP0}}$.<br>0: General I/O port<br>1: $\overline{\text{WKP0}}$ input pin |

### 9.4.2 Port Control Register 5 (PCR5)

PCR5 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 5.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PCR57 | 0 | W | When each of the port 5 pins P57 to P50 functions as a general I/O port, setting a PCR5 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. |
| 6 | PCR56 | 0 | W | |
| 5 | PCR55 | 0 | W | |
| 4 | PCR54 | 0 | W | |
| 3 | PCR53 | 0 | W | |
| 2 | PCR52 | 0 | W | |
| 1 | PCR51 | 0 | W | |
| 0 | PCR50 | 0 | W | |

### 9.4.3 Port Data Register 5 (PDR5)

PDR5 is a general I/O port data register of port 5.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | P57 | 0 | R/W | PDR5 stores output data for port 5 pins. |
| 6 | P56 | 0 | R/W | If PDR5 is read while PCR5 bits are set to 1, the values stored in PDR5 are read. If PDR5 is read while PCR5 bits are cleared to 0, the pin states are read regardless of the value stored in PDR5. |
| 5 | P55 | 0 | R/W | |
| 4 | P54 | 0 | R/W | |
| 3 | P53 | 0 | R/W | |
| 2 | P52 | 0 | R/W | |
| 1 | P51 | 0 | R/W | |
| 0 | P50 | 0 | R/W | |

### 9.4.4 Port Pull-Up Control Register 5 (PUCR5)

PUCR5 controls the pull-up MOS in bit units of the pins set as the input ports.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | — | All 0 | — | Reserved<br>These bits are always read as 0. |
| 5 | PUCR55 | 0 | R/W | Only bits for which PCR5 is cleared are valid. The pull-up MOSs of the corresponding pins enter the on-state when these bits are set to 1, while they enter the off-state when these bits are cleared to 0. |
| 4 | PUCR54 | 0 | R/W | |
| 3 | PUCR53 | 0 | R/W | |
| 2 | PUCR52 | 0 | R/W | |
| 1 | PUCR51 | 0 | R/W | |
| 0 | PUCR50 | 0 | R/W | |

### 9.4.5 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P57/SCL pin

| Register | ICCR1 | PCR5 | |
|---|---|---|---|
| Bit Name | ICE | PCR57 | Pin Function |
| Setting Value | 0 | 0 | P57 input pin |
| | | 1 | P57 output pin |
| | 1 | X | SCL I/O pin |

[Legend] X: Don't care.

SCL performs the NMOS open-drain output, that enables a direct bus drive.

- P56/SDA pin

| Register | ICCR1 | PCR5 | |
|---|---|---|---|
| **Bit Name** | **ICE** | **PCR56** | **Pin Function** |
| Setting Value | 0 | 0 | P56 input pin |
| | | 1 | P56 output pin |
| | 1 | X | SDA I/O pin |

[Legend] X: Don't care.

SDA performs the NMOS open-drain output, that enables a direct bus drive.

- P55/$\overline{\text{WKP5}}$ pin

| Register | PMR5 | PCR5 | |
|---|---|---|---|
| **Bit Name** | **WKP5** | **PCR55** | **Pin Function** |
| Setting Value | 0 | 0 | P55 input pin |
| | | 1 | P55 output pin |
| | 1 | X | $\overline{\text{WKP5}}$ input pin |

[Legend] X: Don't care.

- P54/$\overline{\text{WKP4}}$ pin

| Register | PMR5 | PCR5 | |
|---|---|---|---|
| **Bit Name** | **WKP4** | **PCR54** | **Pin Function** |
| Setting Value | 0 | 0 | P54 input pin |
| | | 1 | P54 output pin |
| | 1 | X | $\overline{\text{WKP4}}$ input pin |

[Legend] X: Don't care.

- P53/$\overline{\text{WKP3}}$ pin

| Register | PMR5 | PCR5 | |
|---|---|---|---|
| **Bit Name** | **WKP3** | **PCR53** | **Pin Function** |
| Setting Value | 0 | 0 | P53 input pin |
| | | 1 | P53 output pin |
| | 1 | X | $\overline{\text{WKP3}}$ input pin |

[Legend] X: Don't care.

- P52/$\overline{\text{WKP2}}$ pin

| Register | PMR5 | PCR5 | |
|---|---|---|---|
| **Bit Name** | **WKP2** | **PCR52** | **Pin Function** |
| Setting Value | 0 | 0 | P52 input pin |
| | | 1 | P52 output pin |
| | 1 | X | $\overline{\text{WKP2}}$ input pin |

[Legend] X: Don't care.

- P51/$\overline{\text{WKP1}}$ pin

| Register | PMR5 | PCR5 | |
|---|---|---|---|
| **Bit Name** | **WKP1** | **PCR51** | **Pin Function** |
| Setting Value | 0 | 0 | P51 input pin |
| | | 1 | P51 output pin |
| | 1 | X | $\overline{\text{WKP1}}$ input pin |

[Legend] X: Don't care.

- P50/$\overline{\text{WKP0}}$ pin

| Register | PMR5 | PCR5 | |
|---|---|---|---|
| **Bit Name** | **WKP0** | **PCR50** | **Pin Function** |
| Setting Value | 0 | 0 | P50 input pin |
| | | 1 | P50 output pin |
| | 1 | X | $\overline{\text{WKP0}}$ input pin |

[Legend] X: Don't care.

## 9.5 Port 7

Port 7 is a general I/O port also functioning as a timer V I/O pin and SCI3_2 I/O pin. Each pin of port 7 is shown in figure 9.5. The register settings of the timer V and SCI3_2 have priority for functions of the pins for both uses.

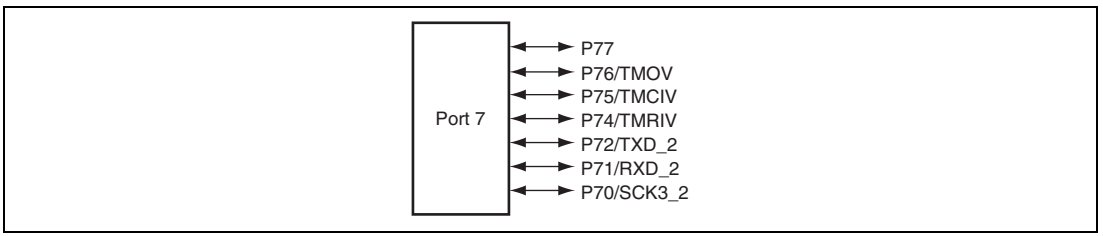



**Figure 9.5 Port 7 Pin Configuration**

Port 7 has the following registers.

- Port control register 7 (PCR7)
- Port data register 7 (PDR7)

### 9.5.1 Port Control Register 7 (PCR7)

PCR7 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 7.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PCR77 | 0 | W | When each of the port 7 pins P77 to P74 and P72 to P70 functions as a general I/O port, setting a PCR7 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. |
| 6 | PCR76 | 0 | W | |
| 5 | PCR75 | 0 | W | |
| 4 | PCR74 | 0 | W | |
| 3 | — | — | — | Bit 3 is a reserved bit. |
| 2 | PCR72 | 0 | W | |
| 1 | PCR71 | 0 | W | |
| 0 | PCR70 | 0 | W | |

### 9.5.2 Port Data Register 7 (PDR7)

PDR7 is a general I/O port data register of port 7.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | P77 | 0 | R/W | PDR7 stores output data for port 7 pins. |
| 6 | P76 | 0 | R/W | If PDR7 is read while PCR7 bits are set to 1, the values stored in PDR7 are read. If PDR7 is read while PCR7 bits are cleared to 0, the pin states are read regardless of the value stored in PDR7. |
| 5 | P75 | 0 | R/W | |
| 4 | P74 | 0 | R/W | |
| 3 | — | 1 | — | Bit 3 is a reserved bit. This bit is always read as 1. |
| 2 | P72 | 0 | R/W | |
| 1 | P71 | 0 | R/W | |
| 0 | P70 | 0 | R/W | |

### 9.5.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P77 pin

| Register | PCR7 | |
|---|---|---|
| Bit Name | PCR77 | Pin Function |
| Setting Value | 0 | P77 input pin |
| | 1 | P77 output pin |

- P76/TMOV pin

| Register | TCSRV | PCR7 | |
|---|---|---|---|
| Bit Name | OS3 to OS0 | PCR76 | Pin Function |
| Setting Value | 0000 | 0 | P76 input pin |
| | | 1 | P76 output pin |
| | Other than above | X | TMOV output pin |

[Legend] X: Don't care.

- P75/TMCIV pin

| Register | PCR7 | |
|---|---|---|
| **Bit Name** | **PCR75** | **Pin Function** |
| Setting Value | 0 | P75 input/TMCIV input pin |
| | 1 | P75 output/TMCIV input pin |

- P74/TMRIV pin

| Register | PCR7 | |
|---|---|---|
| **Bit Name** | **PCR74** | **Pin Function** |
| Setting Value | 0 | P74 input/TMRIV input pin |
| | 1 | P74 output/TMRIV input pin |

- P72/TXD_2 pin

| Register | PMR1 | PCR7 | |
|---|---|---|---|
| **Bit Name** | **TXD2** | **PCR72** | **Pin Function** |
| Setting Value | 0 | 0 | P72 input pin |
| | | 1 | P72 output pin |
| | 1 | X | TXD_2 output pin |

[Legend] X: Don't care.

- P71/RXD_2 pin

| Register | SCR3_2 | PCR7 | |
|---|---|---|---|
| **Bit Name** | **RE** | **PCR71** | **Pin Function** |
| Setting Value | 0 | 0 | P71 input pin |
| | | 1 | P71 output pin |
| | 1 | X | RXD_2 input pin |

[Legend] X: Don't care.

- P70/SCK3_2 pin

| Register | | SCR3_2 | SMR_2 | PCR7 | |
|---|---|---|---|---|---|
| **Bit Name** | **CKE1** | **CKE0** | **COM** | **PCR70** | **Pin Function** |
| Setting Value | 0 | 0 | 0 | 0 | P70 input pin |
| | | | | 1 | P70 output pin |
| | 0 | 0 | 1 | X | SCK3_2 output pin |
| | 0 | 1 | X | X | SCK3_2 output pin |
| | 1 | X | X | X | SCK3_2 input pin |

[Legend] X: Don't care.

## 9.6 Port 8

Port 8 is a general I/O port. Each pin of port 8 is shown in figure 9.6.

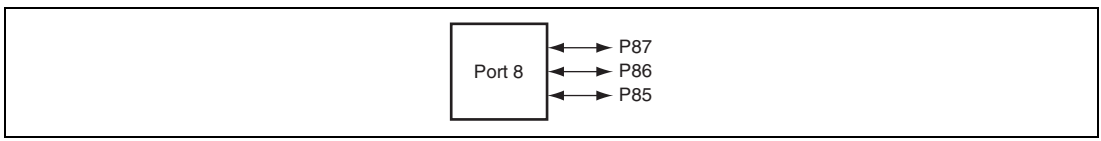



**Figure 9.6 Port 8 Pin Configuration**

Port 8 has the following registers.

- Port control register 8 (PCR8)
- Port data register 8 (PDR8)

### 9.6.1 Port Control Register 8 (PCR8)

PCR8 selects inputs/outputs in bit units for pins to be used as general I/O ports of port 8.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PCR87 | 0 | W | When each of the port 8 pins P87 to P80 functions as a general I/O port, setting a PCR8 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. |
| 6 | PCR86 | 0 | W | |
| 5 | PCR85 | 0 | W | |
| 4 to 0 | — | — | — | Reserved |

### 9.6.2 Port Data Register 8 (PDR8)

PDR8 is a general I/O port data register of port 8.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | P87 | 0 | R/W | PDR8 stores output data for port 8 pins. |
| 6 | P86 | 0 | R/W | If PDR8 is read while PCR8 bits are set to 1, the values stored in PDR8 are read. If PDR8 is read while PCR8 bits are cleared to 0, the pin states are read regardless of the value stored in PDR8. |
| 5 | P85 | 0 | R/W | |
| 4 to 0 | — | All 1 | — | Reserved<br>These bits are always read as 1. |

### 9.6.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- P87 pin

| Register | PCR8 | |
|---|---|---|
| **Bit Name** | **PCR87** | **Pin Function** |
| Setting Value | 0 | P87 input pin |
| | 1 | P87 output pin |

- P86 pin

| Register | PCR8 | |
|---|---|---|
| **Bit Name** | **PCR86** | **Pin Function** |
| Setting Value | 0 | P86 input pin |
| | 1 | P86 output pin |

- P85 pin

| Register | PCR8 | |
|---|---|---|
| **Bit Name** | **PCR85** | **Pin Function** |
| Setting Value | 0 | P85 input pin |
| | 1 | P85 output pin |

## 9.7 Port C

Port C is a general I/O port. Each pin of port C is shown in figure 9.7.

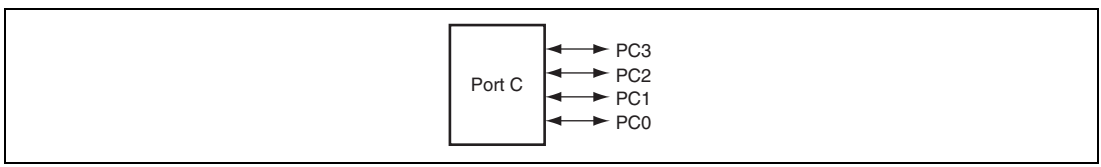



**Figure 9.7 Port C Pin Configuration**

Port C has the following registers.

- Port control register C (PCRC)
- Port data register C (PDRC)

### 9.7.1 Port Control Register C (PCRC)

PCRC selects inputs/outputs in bit units for pins to be used as general I/O ports of port C.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 4 | — | — | — | Reserved |
| 3 | PCRC3 | 0 | W | Setting a PCR9 bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. |
| 2 | PCRC2 | 0 | W | |
| 1 | PCRC1 | 0 | W | |
| 0 | PCRC0 | 0 | W | |

### 9.7.2 Port Data Register C (PDRC)

PDR9 is a general I/O port data register of port C.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 4 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 3 | PC3 | 0 | R/W | PDRC stores output data for port C pins. |
| 2 | PC2 | 0 | R/W | If PDRC is read while PCRC bits are set to 1, the values stored in PDRC are read. If PDRC is read while PCRC bits are cleared to 0, the pin states are read regardless of the value stored in PDRC. |
| 1 | PC1 | 0 | R/W | |
| 0 | PC0 | 0 | R/W | |

### 9.7.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- PC3 pin

| Register | PCRC | |
|---|---|---|
| Bit Name | PCRC3 | Pin Function |
| Setting Value | 0 | PC3 input pin |
| | 1 | PC3 output pin |

- PC2 pin

| Register | PCRC | |
|---|---|---|
| Bit Name | PCRC2 | Pin Function |
| Setting Value | 0 | PC2 input pin |
| | 1 | PC2 output pin |

- PC1 pin

| Register | PCRC | |
|---|---|---|
| **Bit Name** | **PCRC1** | **Pin Function** |
| Setting Value | 0 | PC1 input pin |
| | 1 | PC1 output pin |

- PC0 pin

| Register | PCRC | |
|---|---|---|
| **Bit Name** | **PCRC0** | **Pin Function** |
| Setting Value | 0 | PC0 input pin |
| | 1 | PC0 output pin |

## 9.8 Port D

Port D is a general I/O port also functioning as timer RD_0 I/O pins. Each pin of port D is shown in figure 9.8. The setting for the timer RD_0 function has priority over those for other functions.

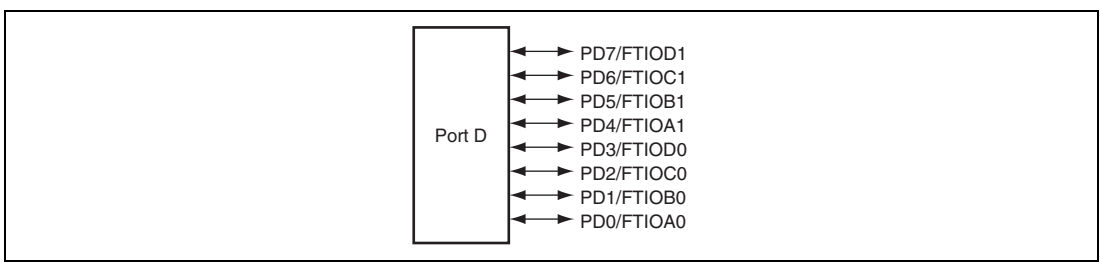



**Figure 9.8 Port D Pin Configuration**

Port D has the following registers.

- Port control register D (PCRD)
- Port data register D (PDRD)

### 9.8.1 Port Control Register D (PCRD)

PCRD selects inputs/outputs in bit units for pins to be used as general I/O ports of port D.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PCRD7 | 0 | W | When each of the port D pins functions as a general I/O port, setting a PCRD bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. |
| 6 | PCRD6 | 0 | W | |
| 5 | PCRD5 | 0 | W | |
| 4 | PCRD4 | 0 | W | |
| 3 | PCRD3 | 0 | W | |
| 2 | PCRD2 | 0 | W | |
| 1 | PCRD1 | 0 | W | |
| 0 | PCRD0 | 0 | W | |

### 9.8.2 Port Data Register D (PDRD)

PDRD is a general I/O port data register of port D.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PD7 | 0 | R/W | PDRD stores output data for port D pins. |
| 6 | PD6 | 0 | R/W | If PDRD is read while PCRD bits are set to 1, the values stored in PDRD are read. If PDRD is read while PCRD bits are cleared to 0, the pin states are read regardless of the value stored in PDRD. |
| 5 | PD5 | 0 | R/W | |
| 4 | PD4 | 0 | R/W | |
| 3 | PD3 | 0 | R/W | |
| 2 | PD2 | 0 | R/W | |
| 1 | PD1 | 0 | R/W | |
| 0 | PD0 | 0 | R/W | |

### 9.8.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- PD7/FTIOD1 pin

| Register | TRDOER1_01 | TRDFCR_01 | | TRDPMR_01 | TRDIORC_1 | PCRD | |
|---|---|---|---|---|---|---|---|
| Bit Name | ED1 | CMD1 and CMD0 | PWM3 | PWMD1 | IOD3 to IOD0 | PCRD7 | Pin Function |
| Setting Value | 1 | XX | X | X | XXXX | 0 | PD7 input/FTIOD1 input pin |
| | | | | | | 1 | PD7 output pin |
| | 0 | 00 | 0 | X | XXXX | 0 | PD7 input/FTIOD1 input pin |
| | | | | | | 1 | PD7 output pin |
| | | | 1 | 1 | XXXX | X | FTIOD1 output pin |
| | | | | 0 | 0XXX | 0 | PD7 input/FTIOD1 input pin |
| | | | | | | 1 | PD7 output pin |
| | | | | | 101X or 1001 | X | FTIOD1 output pin |
| | | | | | 11XX or 1000 | 0 | PD7 input/FTIOD1 input pin |
| | | | | | | 1 | PD7 output pin |
| | | Other than 00 | X | X | XXXX | X | FTIOD1 output pin |

[Legend] X: Don't care.

- PD6/FTIOC1 pin

| Register | TRDOER1_01 | TRDFCR_01 | | TRDPMR_01 | TRDIORC_1 | PCRD | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **EC1** | **CMD1 and CMD0** | **PWM3** | **PWMC1** | **IOC3 to IOC0** | **PCRD6** | **Pin Function** |
| Setting Value | 1 | XX | X | X | XXXX | 0 | PD6 input/FTIOC1 input pin |
| | | | | | | 1 | PD6 output pin |
| | 0 | 00 | 0 | X | XXXX | 0 | PD6 input/FTIOC1 input pin |
| | | | | | | 1 | PD6 output pin |
| | | | 1 | 1 | XXXX | X | FTIOC1 output pin |
| | | | | 0 | 0XXX | 0 | PD6 input/FTIOC1 input pin |
| | | | | | | 1 | PD6 output pin |
| | | | | | 101X or 1001 | X | FTIOC1 output pin |
| | | | | | 11XX or 1000 | 0 | PD6 input/FTIOC1 input pin |
| | | | | | | 1 | PD6 output pin |
| | | Other than 00 | X | X | XXXX | X | FTIOC1 output pin |

[Legend] X: Don't care.

- PD5/FTIOB1 pin

| Register | TRDOER1_01 | TRDFCR_01 | | TRDPMR_01 | TRDIORA_1 | PCRD | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **EB1** | **CMD1 and CMD0** | **PWM3** | **PWMB1** | **IOB2 to IOB0** | **PCRD5** | **Pin Function** |
| Setting Value | 1 | XX | X | X | XXX | 0 | PD5 input/FTIOB1 input pin |
| | | | | | | 1 | PD5 output pin |
| | 0 | 00 | 0 | X | XXX | 0 | PD5 input/FTIOB1 input pin |
| | | | | | | 1 | PD5 output pin |
| | | | 1 | 1 | XXX | X | FTIOB1 output pin |
| | | | | 0 | 01X or 001 | X | FTIOB1 output pin |
| | | | | | 1XX or 000 | 0 | PD5 input/FTIOB1 input pin |
| | | | | | | 1 | PD5 output pin |
| | | Other than 00 | X | X | XXX | X | FTIOB1 output pin |

[Legend] X: Don't care.

- PD4/FTIOA1 pin

| Register | TRDOER1_01 | TRDFCR_01 | | TRDIORA_1 | PCRD | |
|---|---|---|---|---|---|---|
| **Bit Name** | **EA1** | **CMD1 and CMD0** | **PWM3** | **IOA2 to IOA0** | **PCRD4** | **Pin Function** |
| Setting Value | 1 | XX | X | XXX | 0 | PD4 input/FTIOA1 input pin |
| | | | | | 1 | PD4 output pin |
| | 0 | 00 | 0 | XXX | 0 | PD4 input/FTIOA1 input pin |
| | | | | | 1 | PD4 output pin |
| | | | 1 | 01X or 001 | X | FTIOA1 output pin |
| | | | | 1XX or 000 | 0 | PD4 input/FTIOA1 input pin |
| | | | | | 1 | PD4 output pin |
| | | Other than 00 | X | XXX | X | FTIOA1 output pin |

[Legend] X: Don't care.

- PD3/FTIOD0 pin

| Register | TRDOER1_01 | TRDFCR_01 | | TRDPMR_01 | TRDIORC_0 | PCRD | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **ED0** | **CMD1 and CMD0** | **PWM3** | **PWMD0** | **IOD3 to IOD0** | **PCRD3** | **Pin Function** |
| Setting Value | 1 | XX | X | X | XXXX | 0 | PD3 input/FTIOD0 input pin |
| | | | | | | 1 | PD3 output pin |
| | 0 | 00 | 0 | X | XXXX | 0 | PD3 input/FTIOD0 input pin |
| | | | | | | 1 | PD3 output pin |
| | | | 1 | 1 | XXXX | X | FTIOD0 output pin |
| | | | | 0 | 0XXX | 0 | PD3 input/FTIOD0 input pin |
| | | | | | | 1 | PD3 output pin |
| | | | | | 101X or 1001 | X | FTIOD0 output pin |
| | | | | | 11XX or 1000 | 0 | PD3 input/FTIOD0 input pin |
| | | | | | | 1 | PD3 output pin |
| | | Other than 00 | X | X | XXXX | X | FTIOD0 output pin |

[Legend] X: Don't care.

- PD2/FTIOC0 pin

| Register | TRDOER1_01 | TRDFCR_01 | | TRDPMR_01 | TRDIORC_0 | PCRD | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **EC0** | **CMD1 and CMD0** | **PWM3** | **PWMC0** | **IOC3 to IOC0** | **PCRD2** | **Pin Function** |
| Setting Value | 1 | XX | X | X | XXXX | 0 | PD2 input/FTIOC0 input pin |
| | | | | | | 1 | PD2 output pin |
| | 0 | 00 | 0 | X | XXXX | 0 | PD2 input/FTIOC0 input pin |
| | | | | | | 1 | PD2 output pin |
| | | | 1 | 1 | XXXX | X | FTIOC0 output pin |
| | | | | 0 | 0XXX | 0 | PD2 input/FTIOC0 input pin |
| | | | | | | 1 | PD2 output pin |
| | | | | | 101X or 1001 | X | FTIOC0 output pin |
| | | | | | 11XX or 1000 | 0 | PD2 input/FTIOC0 input pin |
| | | | | | | 1 | PD2 output pin |
| | | Other than 00 | X | X | XXXX | X | FTIOC0 output pin |

[Legend] X: Don't care.

- PD1/FTIOB0 pin

| Register | TRDOER1_01 | TRDFCR_01 | | TRDPMR_01 | TRDIORA_0 | PCRD | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **EB0** | **CMD1 and CMD0** | **PWM3** | **PWMB0** | **IOB2 to IOB0** | **PCRD1** | **Pin Function** |
| Setting Value | 1 | XX | X | X | XXX | 0 | PD1 input/FTIOB0 input pin |
| | | | | | | 1 | PD1 output pin |
| | 0 | 00 | 0 | X | XXX | X | FTIOB0 output pin |
| | | | 1 | 1 | XXX | X | FTIOB0 output pin |
| | | | | 0 | 01X or 001 | X | FTIOB0 output pin |
| | | | | | 1XX or 000 | 0 | PD1 input/FTIOB0 input pin |
| | | | | | | 1 | PD1 output pin |
| | | Other than 00 | X | X | XXX | X | FTIOB0 output pin |

[Legend] X: Don't care.

- PD0/FTIOA0 pin

| Register | TRDOER1_01 | | TRDFCR_01 | | TRDIORA_0 | PCRD | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **EA0** | **STCLK** | **CMD1 and CMD0** | **PWM3** | **IOA2 to IOA0** | **PCRD0** | **Pin Function** |
| Setting Value | 1 | X | XX | X | XXX | 0 | PD0 input/FTIOA0 input pin |
| | | | | | | 1 | PD0 output pin |
| | 0 | 1 | XX | X | XXX | 0 | PD0 input/FTIOA0 input pin |
| | | | | | | 1 | PD0 output pin |
| | | 0 | 00 | 0 | XXX | X | FTIOA0 output pin |
| | | | | 1 | 01X or 001 | X | FTIOA0 output pin |
| | | | | | 1XX or 000 | 0 | PD0 input/FTIOA0 input pin |
| | | | | | | 1 | PD0 output pin |
| | | | Other than 00 | X | XXX | 0 | PD0 input/FTIOA0 input pin |
| | | | | | | 1 | PD0 output pin |

[Legend] X: Don't care.

## 9.9 Port E

Port E is a general I/O port also functioning as timer RD_1 I/O pins. Each pin of port E is shown in figure 9.9. The setting of the timer RD_1 function has priority over those for other functions.

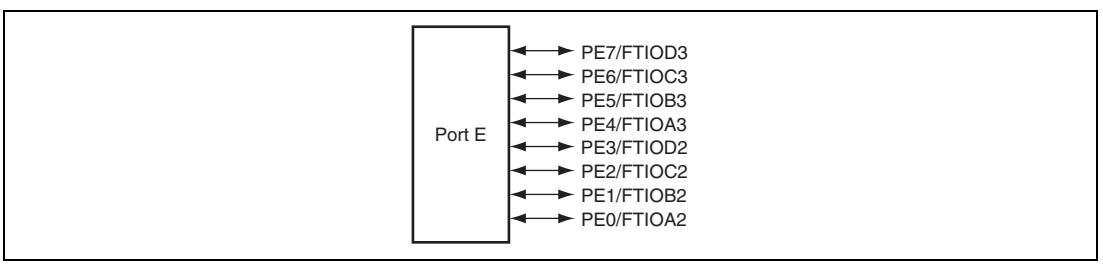



**Figure 9.9 Port E Pin Configuration**

Port E has the following registers.

- Port control register E (PCRE)
- Port data register E (PDRE)

### 9.9.1 Port Control Register E (PCRE)

PCRE selects inputs/outputs in bit units for pins to be used as general I/O ports of port E.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PCRE7 | 0 | W | When each of the port E pins functions as a general I/O port, setting a PCRE bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. |
| 6 | PCRE6 | 0 | W | |
| 5 | PCRE5 | 0 | W | |
| 4 | PCRE4 | 0 | W | |
| 3 | PCRE3 | 0 | W | |
| 2 | PCRE2 | 0 | W | |
| 1 | PCRE1 | 0 | W | |
| 0 | PCRE0 | 0 | W | |

### 9.9.2 Port Data Register E (PDRE)

PDRE is a general I/O port data register of port E.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PE7 | 0 | R/W | PDRE stores output data for port E pins. |
| 6 | PE6 | 0 | R/W | If PDRE is read while PCRE bits are set to 1, the values stored in PDRE are read. If PDRE is read while PCRE bits are cleared to 0, the pin states are read regardless of the value stored in PDRE. |
| 5 | PE5 | 0 | R/W | |
| 4 | PE4 | 0 | R/W | |
| 3 | PE3 | 0 | R/W | |
| 2 | PE2 | 0 | R/W | |
| 1 | PE1 | 0 | R/W | |
| 0 | PE0 | 0 | R/W | |

### 9.9.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- PE7/FTIOD3 pin

| Register | TRDOER1_23 | TRDFCR_23 | | TRDPMR_23 | TRDIORC_3 | PCRE | |
|---|---|---|---|---|---|---|---|
| Bit Name | ED1 | CMD1 and CMD0 | PWM3 | PWMD1 | IOD3 to IOD0 | PCRE7 | Pin Function |
| Setting Value | 1 | XX | X | X | XXXX | 0 | PE7 input/FTIOD3 input pin |
| | | | | | | 1 | PE7 output pin |
| | 0 | 00 | 0 | X | XXXX | 0 | PE7 input/FTIOD3 input pin |
| | | | | | | 1 | PE7 output pin |
| | | | 1 | 1 | XXXX | X | FTIOD3 output pin |
| | | | | 0 | 0XXX | 0 | PE7 input/FTIOD3 input pin |
| | | | | | | 1 | PE7 output pin |
| | | | | | 101X or 1001 | X | FTIOD3 output pin |
| | | | | | 11XX or 1000 | 0 | PE7 input/FTIOD3 input pin |
| | | | | | | 1 | PE7 output pin |
| | | Other than 00 | X | X | XXXX | X | FTIOD3 output pin |

[Legend] X: Don't care.

- PE6/FTIOC3 pin

| Register | TRDOER1_23 | TRDFCR_23 | | TRDPMR_23 | TRDIORC_3 | PCRE | |
|---|---|---|---|---|---|---|---|
| Bit Name | EC1 | CMD1 and CMD0 | PWM3 | PWMC1 | IOC3 to IOC0 | PCRE6 | Pin Function |
| Setting Value | 1 | XX | X | X | XXXX | 0 | PE6 input/FTIOC3 input pin |
| | | | | | | 1 | PE6 output pin |
| | 0 | 00 | 0 | X | XXXX | 0 | PE6 input/FTIOC3 input pin |
| | | | | | | 1 | PE6 output pin |
| | | | 1 | 1 | XXXX | X | FTIOC3 output pin |
| | | | | 0 | 0XXX | 0 | PE6 input/FTIOC3 input pin |
| | | | | | | 1 | PE6 output pin |
| | | | | | 101X or 1001 | X | FTIOC3 output pin |
| | | | | | 11XX or 1000 | 0 | PE6 input/FTIOC3 input pin |
| | | | | | | 1 | PE6 output pin |
| | | Other than 00 | X | X | XXXX | X | FTIOC3 output pin |

[Legend] X: Don't care.

- PE5/FTIOB3 pin

| Register | TRDOER1_23 | TRDFCR_23 | | TRDPMR_23 | TRDIORA_3 | PCRE | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **EB1** | **CMD1 and CMD0** | **PWM3** | **PWMB1** | **IOB2 to IOB0** | **PCRE5** | **Pin Function** |
| Setting Value | 1 | XX | X | X | XXX | 0 | PE5 input/FTIOB3 input pin |
| | | | | | | 1 | PE5 output pin |
| | 0 | 00 | 0 | X | XXX | 0 | PE5 input/FTIOB3 input pin |
| | | | | | | 1 | PE5 output pin |
| | | | 1 | 1 | XXX | X | FTIOB3 output pin |
| | | | | 0 | 01X or 001 | X | FTIOB3 output pin |
| | | | | | 1XX or 000 | 0 | PE5 input/FTIOB3 input pin |
| | | | | | | 1 | PE5 output pin |
| | | Other than 00 | X | X | XXX | X | FTIOB3 output pin |

[Legend] X: Don't care.

- PE4/FTIOA3 pin

| Register | TRDOER1_23 | TRDFCR_23 | | TRDIORA_3 | PCRE | |
|---|---|---|---|---|---|---|
| Bit Name | EA1 | CMD1 and CMD0 | PWM3 | IOA2 to IOA0 | PCRE4 | Pin Function |
| Setting Value | 1 | XX | X | XXX | 0 | PE4 input/FTIOA3 input pin |
| | | | | | 1 | PE4 output pin |
| | 0 | 00 | 0 | XXX | 0 | PE4 input/FTIOA3 input pin |
| | | | | | 1 | PE4 output pin |
| | | | 1 | 01X or 001 | X | FTIOA3 output pin |
| | | | | 1XX or 000 | 0 | PE4 input/FTIOA3 input pin |
| | | | | | 1 | PE4 output pin |
| | | Other than 00 | X | XXX | X | FTIOA3 output pin |

[Legend] X: Don't care.

- PE3/FTIOD2 pin

| Register | TRDOER1_23 | TRDFCR_23 | | TRDPMR_23 | TRDIORC_2 | PCRE | |
|---|---|---|---|---|---|---|---|
| Bit Name | ED0 | CMD1 and CMD0 | PWM3 | PWMD0 | IOD3 to IOD0 | PCRE3 | Pin Function |
| Setting Value | 1 | XX | X | X | XXXX | 0 | PE3 input/FTIOD2 input pin |
| | | | | | | 1 | PE3 output pin |
| | 0 | 00 | 0 | X | XXXX | 0 | PE3 input/FTIOD2 input pin |
| | | | | | | 1 | PE3 output pin |
| | | | 1 | 1 | XXXX | X | FTIOD2 output pin |
| | | | | 0 | 0XXX | 0 | PE3 input/FTIOD2 input pin |
| | | | | | | 1 | PE3 output pin |
| | | | | | 101X or 1001 | X | FTIOD2 output pin |
| | | | | | 11XX or 1000 | 0 | PE3 input/FTIOD2 input pin |
| | | | | | | 1 | PE3 output pin |
| | | Other than 00 | X | X | XXXX | X | FTIOD2 output pin |

[Legend] X: Don't care.

- PE2/FTIOC2 pin

| Register | TRDOER1_23 | TRDFCR_23 | | TRDPMR_23 | TRDIORC_2 | PCRE | |
|---|---|---|---|---|---|---|---|
| Bit Name | EC0 | CMD1 and CMD0 | PWM3 | PWMC0 | IOC3 to IOC0 | PCRE2 | Pin Function |
| Setting Value | 1 | XX | X | X | XXXX | 0 | PE2 input/FTIOC2 input pin |
| | | | | | | 1 | PE2 output pin |
| | 0 | 00 | 0 | X | XXXX | 0 | PE2 input/FTIOC2 input pin |
| | | | | | | 1 | PE2 output pin |
| | | | 1 | 1 | XXXX | X | FTIOC2 output pin |
| | | | | 0 | 0XXX | 0 | PE2 input/FTIOC2 input pin |
| | | | | | | 1 | PE2 output pin |
| | | | | | 101X or 1001 | X | FTIOC2 output pin |
| | | | | | 11XX or 1000 | 0 | PE2 input/FTIOC2 input pin |
| | | | | | | 1 | PE2 output pin |
| | | Other than 00 | X | X | XXXX | X | FTIOC2 output pin |

[Legend] X: Don't care.

- PE1/FTIOB2 pin

| Register | TRDOER1_23 | TRDFCR_23 | | TRDPMR_23 | TRDIORA_2 | PCRE | |
|---|---|---|---|---|---|---|---|
| Bit Name | EB0 | CMD1 and CMD0 | PWM3 | PWMB0 | IOB2 to IOB0 | PCRE1 | Pin Function |
| Setting Value | 1 | XX | X | X | XXX | 0 | PE1 input/FTIOB2 input pin |
| | | | | | | 1 | PE1 output pin |
| | 0 | 00 | 0 | X | XXX | X | FTIOB2 output pin |
| | | | 1 | 1 | XXX | X | FTIOB2 output pin |
| | | | | 0 | 01X or 001 | X | FTIOB2 output pin |
| | | | | | 1XX or 000 | 0 | PE1 input/FTIOB2 input pin |
| | | | | | | 1 | PE1 output pin |
| | | Other than 00 | X | X | XXX | X | FTIOB2 output pin |

[Legend] X: Don't care.

- PE0/FTIOA2 pin

| Register | TRDOER 1_23 | | TRDFCR_23 | | TRDIOR A_2 | PCRE | |
|---|---|---|---|---|---|---|---|
| Bit Name | EA0 | STCLK | CMD1 and CMD0 | PWM3 | IOA2 to IOA0 | PCRE0 | Pin Function |
| Setting Value | 1 | X | XX | X | XXX | 0 | PE0 input/FTIOA2 input pin |
| | | | | | | 1 | PE0 output pin |
| | 0 | 1 | XX | X | XXX | 0 | PE0 input/FTIOA2 input pin |
| | | | | | | 1 | PE0 output pin |
| | | 0 | 00 | 0 | XXX | X | FTIOA2 output pin |
| | | | | 1 | 01X or 001 | X | FTIOA2 output pin |
| | | | | | 1XX or 000 | 0 | PE0 input/FTIOA2 input pin |
| | | | | | | 1 | PE0 output pin |
| | | | Other than 00 | X | XXX | 0 | FTIOA2 output pin |

[Legend] X: Don't care.

## 9.10 Port F

Port F is a general input port also functioning as A/D converter analog input pins. Each pin of port F is shown in figure 9.10.

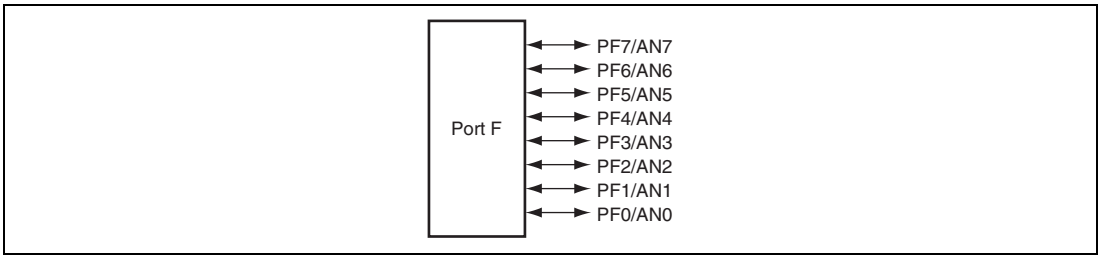



**Figure 9.10 Port F Pin Configuration**

Port F has the following registers.

- Port data register F (PDRF)
- Port mode register F (PMRF)

### 9.10.1 Port Data Register F (PDRF)

PDRF is a general input port data register of port F.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PF7 | — | R | If PDRF is read, the pin states are read. |
| 6 | PF6 | — | R | However, if a port F pin is specified as an analog input channel by ADCSR in the A/D converter, the bit is read as 0. |
| 5 | PF5 | — | R | |
| 4 | PF4 | — | R | |
| 3 | PF3 | — | R | |
| 2 | PF2 | — | R | |
| 1 | PF1 | — | R | |
| 0 | PF0 | — | R | |

### 9.10.2 Port Mode Register F (PMRF)

PMRF switches functions of pins in port F.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 0 | PMRF0 | 0 | R/W | This bit selects the function of pin PF0/AN0.<br>0: AN0 input pin<br>1: General input port |

### 9.10.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- PF7/AN7 pin

| Register | ADCR | ADCSR | | | | |
|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **Pin Function** |
| Setting Value | 0 | X | 1 | 1 | 1 | AN7 input pin |
| | Other than above | | | | | PF7 input pin |

[Legend] X: Don't care.

- PF6/AN6 pin

| Register | ADCR | ADCSR | | | | |
|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **Pin Function** |
| Setting Value | 0 | 0 | 1 | 1 | 0 | AN6 input pin |
| | | 1 | 1 | 1 | X | |
| | Other than above | | | | | PF6 input pin |

[Legend] X: Don't care.

- PF5/AN5 pin

| Register | ADCR | ADCSR | | | | |
|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **Pin Function** |
| Setting Value | 0 | 0 | 1 | 0 | 1 | AN5 input pin |
| | | 1 | 1 | 1 | X | |
| | | | | 0 | 1 | |
| | Other than above | | | | | PF5 input pin |

[Legend] X: Don't care.

- PF4/AN4 pin

| Register | ADCR | ADCSR | | | | |
|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **Pin Function** |
| Setting Value | 0 | 0 | 1 | 0 | 0 | AN4 input pin |
| | | 1 | 1 | X | X | |
| | Other than above | | | | | PF4 input pin |

[Legend] X: Don't care.

- PF3/AN3 pin

| Register | ADCR | ADCSR | | | | |
|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **Pin Function** |
| Setting Value | 0 | X | 0 | 1 | 1 | AN3 input pin |
| | Other than above | | | | | PF3 input pin |

[Legend] X: Don't care.

- PF2/AN2 pin

| Register | ADCR | ADCSR | | | | |
|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **Pin Function** |
| Setting Value | 0 | 0 | 0 | 1 | 0 | AN2 input pin |
| | | 1 | 0 | 1 | X | |
| | Other than above | | | | | PF2 input pin |

[Legend] X: Don't care.

- PF1/AN1 pin

| Register | ADCR | ADCSR | | | | |
|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **Pin Function** |
| Setting Value | 0 | 0 | 0 | 0 | 1 | AN1 input pin |
| | | 1 | 0 | 1 | X | |
| | | | | 0 | 1 | |
| | Other than above | | | | | PF1 input pin |

[Legend] X: Don't care.

- PF0/AN0 pin

| Register | PMRF | ADCR | ADCSR | | | | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **PF0** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **Pin Function** |
| Setting Value | 0 | 0 | 0 | 0 | 0 | 0 | AN0 input pin |
| | | | 1 | 0 | X | X | |
| | Other than above | | | | | | PF0 input pin |

[Legend] X: Don't care.

## 9.11 Port G

Port G is a general input port also functioning as A/D converter analog input pins, timer RC input pins, and timer RD input pins. Each pin of port G is shown in figure 9.11.

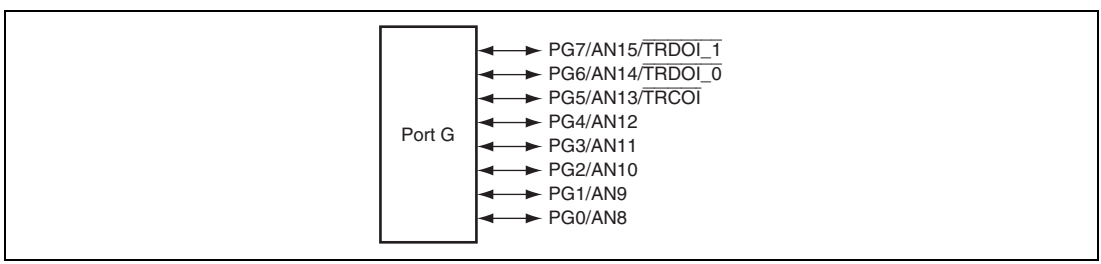



**Figure 9.11 Port G Pin Configuration**

Port G has the following registers.

- Port control register G (PCRG)
- Port data register G (PDRG)
- Port mode register G (PMRG)

### 9.11.1 Port Control Register G (PCRG)

PCRG selects inputs/outputs in bit units for pins to be used as general I/O ports of port G.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PCRG7 | 0 | W | When each of the port G pins functions as a general I/O port, setting a PCRG bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. |
| 6 | PCRG6 | 0 | W | |
| 5 | PCRG5 | 0 | W | |
| 4 | PCRG4 | 0 | W | |
| 3 | PCRG3 | 0 | W | |
| 2 | PCRG2 | 0 | W | |
| 1 | PCRG1 | 0 | W | |
| 0 | PCRG0 | 0 | W | |

### 9.11.2 Port Data Register G (PDRG)

PDRG is a general I/O port data register of port G.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PG7 | 0 | R/W | PDRG stores output data for port G pins. |
| 6 | PG6 | 0 | R/W | If PDRG is read while PCRG bits are set to 1, the values stored in PDRG are read. If PDRG is read while PCRG bits are cleared to 0, the pin states are read regardless of the value stored in PDRG. However, if a port G pin is specified as an analog input channel by ADCSR or ADCR in the A/D converter, the bit is read as 0 even when the PGRG bit is cleared. |
| 5 | PG5 | 0 | R/W | |
| 4 | PG4 | 0 | R/W | |
| 3 | PG3 | 1 | R/W | |
| 2 | PG2 | 0 | R/W | |
| 1 | PG1 | 0 | R/W | |
| 0 | PG0 | 0 | R/W | |

### 9.11.3 Port Mode Register G (PMRG)

PMRG switches functions of pins in port G.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PMRG7 | 0 | R/W | This bit selects the function of pin PG7/AN15/$\overline{\text{TRDOI_1}}$.<br>0: General I/O port<br>1: AN15/$\overline{\text{TRDOI_1}}$ input pin |
| 6 | PMRG6 | 0 | R/W | This bit selects the function of pin PG6/AN14/$\overline{\text{TRDOI_0}}$.<br>0: General I/O port<br>1: AN14/$\overline{\text{TRDOI_0}}$ input pin |
| 5 | PMRG5 | 0 | R/W | This bit selects the function of pin PG5/AN13/$\overline{\text{TRCOI}}$.<br>0: General I/O port<br>1: AN14/$\overline{\text{TRCOI}}$ input pin |
| 4 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 3 | PMRG3 | 0 | R/W | These bits select the trigger source of the A/D converter. |
| 2 | PMRG2 | 0 | R/W | 00: A/D converter is activated by the $\overline{\text{ADTRG}}$ signal<br>01: A/D converter is activated by timer RD_0<br>10: A/D converter is activated by timer RD_1<br>11: Reserved |
| 1 | PMRG1 | 0 | R/W | Selects the edge of the $\overline{\text{ADTRG}}$ signal.<br>0: Falling edge<br>1: Rising edge |
| 0 | PMRG0 | 0 | R/W | This bit selects the function of pin PH0/SCK3_3/$\overline{\text{ADTRG}}$.<br>0: General I/O port<br>1: $\overline{\text{ADTRG}}$ input pin |

### 9.11.4 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- PG7/AN15/$\overline{\text{TRDOI_1}}$ pin

| Register | ADCR | ADCSR | | | | PMRG | PCRG | |
|---|---|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **PMRG7** | **PCRG7** | **Pin Function** |
| Setting Value | 1 | X | 1 | 1 | 1 | X | X | AN15 input pin |
| | Other than above | | | | | 1 | X | $\overline{\text{TRDOI_1}}$ input pin |
| | | | | | | 0 | 0 | PG7 input pin |
| | | | | | | | 1 | PG7 output pin |

[Legend] X: Don't care.

- PG6/AN14/$\overline{\text{TRDOI_0}}$ pin

| Register | ADCR | ADCSR | | | | PMRG | PCRG | |
|---|---|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **PMRG6** | **PCRG6** | **Pin Function** |
| Setting Value | 1 | X | 1 | 1 | 0 | X | X | AN14 input pin |
| | 1 | 1 | 1 | 1 | 1 | X | X | |
| | Other than above | | | | | 1 | X | $\overline{\text{TRDOI_0}}$ input pin |
| | | | | | | 0 | 0 | PG6 input pin |
| | | | | | | | 1 | PG6 output pin |

[Legend] X: Don't care.

- PG5/AN13/$\overline{\text{TRCOI}}$ pin

| Register | ADCR | ADCSR | | | | PMRG | PCRG | |
|---|---|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **PMRG5** | **PCRG5** | **Pin Function** |
| Setting Value | 1 | X | 1 | 0 | 1 | X | X | AN13 input pin |
| | 1 | 1 | 1 | 1 | 0 | X | X | |
| | 1 | 1 | 1 | 1 | 1 | X | X | |
| | Other than above | | | | | 1 | X | $\overline{\text{TRCOI}}$ input pin |
| | | | | | | 0 | 0 | PG5 input pin |
| | | | | | | | 1 | PG5 output pin |

[Legend] X: Don't care.

- PG4/AN12 pin

| Register | ADCR | ADCSR | | | | PCRG | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **PCRG4** | **Pin Function** |
| Setting Value | 1 | X | 1 | 0 | 0 | X | AN12 input pin |
| | 1 | 1 | 1 | X | X | X | |
| | Other than above | | | | | 0 | PG4 input pin |
| | | | | | | 1 | PG4 output pin |

[Legend] X: Don't care.

- PG3/AN11 pin

| Register | ADCR | ADCSR | | | | PCRG | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **PCRG3** | **Pin Function** |
| Setting Value | 1 | X | 0 | 1 | 1 | X | AN11 input pin |
| | Other than above | | | | | 0 | PG3 input pin |
| | | | | | | 1 | PG3 output pin |

[Legend] X: Don't care.

- PG2/AN10 pin

| Register | ADCR | | ADCSR | | | PCRG | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **PCRG2** | **Pin Function** |
| Setting Value | 1 | X | 0 | 1 | 0 | X | AN10 input pin |
| | 1 | 1 | 0 | 1 | 1 | X | |
| | Other than above | | | | | 0 | PG2 input pin |
| | | | | | | 1 | PG2 output pin |

[Legend] X: Don't care.

- PG1/AN9 pin

| Register | ADCR | | ADCSR | | | PCRG | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **PCRG1** | **Pin Function** |
| Setting Value | 1 | X | 0 | 0 | 1 | X | AN9 input pin |
| | 1 | 1 | 0 | 1 | 0 | X | |
| | 1 | 1 | 0 | 1 | 1 | X | |
| | Other than above | | | | | 0 | PG1 input pin |
| | | | | | | 1 | PG1 output pin |

[Legend] X: Don't care.

- PG0/AN8 pin

| Register | ADCR | | ADCSR | | | PCRG | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **CH3** | **SCAN** | **CH2** | **CH1** | **CH0** | **PCRG0** | **Pin Function** |
| Setting Value | 1 | X | 0 | 0 | 0 | X | AN8 input pin |
| | 1 | 1 | 0 | X | X | X | |
| | Other than above | | | | | 0 | PG0 input pin |
| | | | | | | 1 | PG0 output pin |

[Legend] X: Don't care.

## 9.12 Port H

Port H is a general I/O port also functioning as SCI3_3 I/O pins, timer RC input pins, and A/D converter input pins. Each pin of port H is shown in figure 9.12. The settings for the SCI3_3 and timer RC functions have priority over those for other functions.

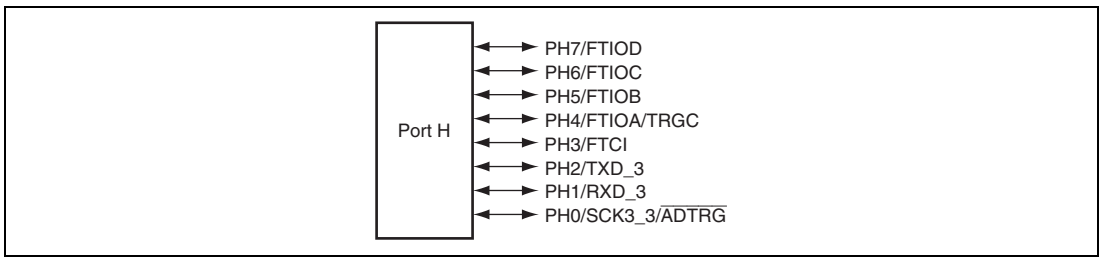



**Figure 9.12 Port H Pin Configuration**

Port H has the following registers.

- Port control register H (PCRH)
- Port data register H (PDRH)

### 9.12.1 Port Control Register H (PCRH)

PCRH selects inputs/outputs in bit units for pins to be used as general I/O ports of port H.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PCRH7 | 0 | W | When each of the port H pins PH7 to PH0 functions as a general I/O port, setting a PCRH bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. |
| 6 | PCRH6 | 0 | W | |
| 5 | PCRH5 | 0 | W | |
| 4 | PCRH4 | 0 | W | |
| 3 | PCRH3 | 0 | W | |
| 2 | PCRH2 | 0 | W | |
| 1 | PCRH1 | 0 | W | |
| 0 | PCRH0 | 0 | W | |

### 9.12.2 Port Data Register H (PDRH)

PDRH is a general I/O port data register of port H.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PH7 | 0 | R/W | PDRH stores output data for port H pins. |
| 6 | PH6 | 0 | R/W | If PDRH is read while PCRH bits are set to 1, the values stored in PDRH are read. If PDRH is read while PCRH bits are cleared to 0, the pin states are read regardless of the value stored in PDRH. |
| 5 | PH5 | 0 | R/W | |
| 4 | PH4 | 0 | R/W | |
| 3 | PH3 | 0 | R/W | |
| 2 | PH2 | 0 | R/W | |
| 1 | PH1 | 0 | R/W | |
| 0 | PH0 | 0 | R/W | |

### 9.12.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- PH7/FTIOD pin

| Register | TRCOER | TRCMR | | TRCIOR1 | | | PCRH | |
|---|---|---|---|---|---|---|---|---|
| Bit Name | ED | PWM2 | PWMD | IOD2 | IOD1 | IOD0 | PCRH7 | Pin Function |
| Setting Value | 1 | X | X | X | X | X | 0 | PH7 input/FTIOD input pin |
| | | | | | | | 1 | PH7 output pin |
| | 0 | 0 | X | X | X | X | 0 | PH7 input/FTIOD input pin |
| | | | | | | | 1 | PH7 output pin |
| | | 1 | 1 | X | X | X | X | FTIOD (PWM) output pin |
| | | | 0 | 0 | 1 | X | X | FTIOD output pin |
| | | | | | 0 | 1 | X | FTIOD output pin |
| | | | | | | 0 | 0 | PH7 input/FTIOD input pin |
| | | | | | | | 1 | PH7 output pin |
| | | | | 1 | X | X | 0 | PH7 input/FTIOD input pin |
| | | | | | | | 1 | PH7 output pin |

[Legend] X: Don't care.

- PH6/FTIOC pin

| Register | TRCOER | TRCMR | | TRCIOR1 | | | PCRH | |
|---|---|---|---|---|---|---|---|---|
| **Bit Name** | **EC** | **PWM2** | **PWMC** | **IOC2** | **IOC1** | **IOC0** | **PCRH6** | **Pin Function** |
| Setting Value | 1 | X | X | X | X | X | 0 | PH6 input/FTIOC input pin |
| | | | | | | | 1 | PH6 output pin |
| | 0 | 0 | X | X | X | X | 0 | PH6 input/FTIOC input pin |
| | | | | | | | 1 | PH6 output pin |
| | | 1 | 1 | X | X | X | X | FTIOC (PWM) output pin |
| | | | 0 | 0 | 1 | X | X | FTIOC output pin |
| | | | | | 0 | 1 | X | FTIOC output pin |
| | | | | | | 0 | 0 | PH6 input/FTIOC input pin |
| | | | | | | | 1 | PH6 output pin |
| | | | | 1 | X | X | 0 | PH6 input/FTIOC input pin |
| | | | | | | | 1 | PH6 output pin |

[Legend] X: Don't care.

- PH5/FTIOB pin

| Register | TRCOER | TRCMR | | TRCIOR0 | | | PCRH | |
|---|---|---|---|---|---|---|---|---|
| **Bit Name** | **EB** | **PWM2** | **PWMB** | **IOB2** | **IOB1** | **IOB0** | **PCRH5** | **Pin Function** |
| Setting Value | 1 | X | X | X | X | X | 0 | PH5 input/FTIOB input pin |
| | | | | | | | 1 | PH5 output |
| | 0 | 0 | X | X | X | X | X | FTIOB (PWM2) output pin |
| | | 1 | 1 | X | X | X | X | FTIOB (PWM) output pin |
| | | | 0 | 0 | 1 | X | X | FTIOB output pin |
| | | | | | 0 | 1 | X | FTIOB output pin |
| | | | | | | 0 | 0 | PH5 input/FTIOB input pin |
| | | | | | | | 1 | PH5 output |
| | | | | 1 | X | X | 0 | PH5 input/FTIOB input pin |
| | | | | | | | 1 | PH5 output |

[Legend] X: Don't care.

- PH4/FTIOA/TRGC pin

| Register | TRCOER | TRCMR | | TRCIOR0 | | PCRH | |
|---|---|---|---|---|---|---|---|
| **Bit Name** | **EA** | **PWM2** | **IOA2** | **IOA1** | **IOA0** | **PCRH4** | **Pin Function** |
| Setting Value | 1 | X | X | X | X | 0 | PH4 input/FTIOA input /TRGC input pin |
| | | | | | | 1 | PH4 output pin |
| | 0 | 0 | X | X | X | 0 | PH4 input/FTIOA input /TRGC input pin |
| | | | | | | 1 | PH4 output pin |
| | | 1 | 0 | 1 | X | X | FTIOA output pin |
| | | | | 0 | 1 | X | FTIOA output pin |
| | | | | | 0 | 0 | PH4 input/FTIOA input /TRGC input pin |
| | | | | | | 1 | PH4 output pin |
| | | | 1 | X | X | 0 | PH4 input/FTIOA input /TRGC input pin |
| | | | | | | 1 | PH4 output pin |

[Legend] X: Don't care.

- PH3/FTCI pin

| Register | PCRH | |
|---|---|---|
| **Bit Name** | **PCRH3** | **Pin Function** |
| Setting Value | 0 | PH3 input/FTCI input pin |
| | 1 | PH3 output/FTCI input pin |

[Legend] X: Don't care.

- PH2/TXD_3 pin

| Register | SMCR | PCRH | |
|---|---|---|---|
| **Bit Name** | **TXD_3** | **PCRH2** | **Pin Function** |
| Setting Value | 0 | 0 | PH2 input pin |
| | | 1 | PH2 output pin |
| | 1 | X | TXD_3 output pin |

[Legend] X: Don't care.

- PH1/RXD_3 pin

| Register | SCR3_3 | PCRH | |
|---|---|---|---|
| **Bit Name** | **RE** | **PCRH1** | **Pin Function** |
| Setting Value | 0 | 0 | PH1 input pin |
| | | 1 | PH1 output pin |
| | 1 | X | RXD_3 input pin |

[Legend] X: Don't care.

- PH0/SCK3_3/$\overline{\text{ADTRG}}$ pin

| Register | SCR3_3 | | SMR3_3 | PMRG | PCRH | |
|---|---|---|---|---|---|---|
| **Bit Name** | **CKE1** | **CKE0** | **COM** | **PMRG0** | **PCRH0** | **Pin Function** |
| Setting Value | 0 | 0 | 0 | 0 | 0 | PH0 input pin |
| | | | | | 1 | PH0 output pin |
| | | | | 1 | X | $\overline{\text{ADTRG}}$ input pin |
| | 0 | 0 | 1 | X | X | SCK3_3 output pin |
| | 0 | 1 | X | X | X | SCK3_3 output pin |
| | 1 | X | X | X | X | SCK3_3 input pin |

[Legend] X: Don't care.

## 9.13 Port J

Port J is a general I/O port also functioning as external oscillation pins and a clock output pin. Each pin of port J is shown in figure 9.13. The setting of CKCSR has priority over those for other functions.

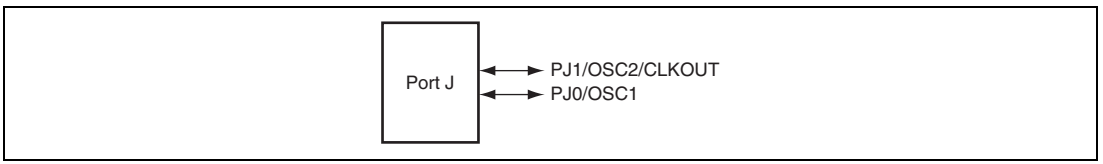



**Figure 9.13 Port J Pin Configuration**

Port J has the following registers.

- Port control register J (PCRJ)
- Port data register J (PDRJ)

### 9.13.1 Port Control Register J (PCRJ)

PCRJ selects inputs/outputs in bit units for pins to be used as general I/O ports of port J.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 2 | — | — | — | Reserved |
| 1 | PCRJ1 | 0 | W | When each of the port J pins PJ1 to PJ0 functions as a general I/O port, setting a PCRJ bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port. |
| 0 | PCRJ0 | 0 | W | |

### 9.13.2 Port Data Register J (PDRJ)

PDRJ is a general I/O port data register of port J.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 2 | — | — | — | Reserved |
| 1 | PJ1 | 0 | R/W | PDRJ stores output data for port J pins. |
| 0 | PJ0 | 0 | R/W | If PDRJ is read while PCRJ bits are set to 1, the values stored in PDRJ are read. If PDRJ is read while PCRJ bits are cleared to 0, the pin states are read regardless of the value stored in PDRJ. |

### 9.13.3 Pin Functions

The correspondence between the register specification and the port functions is shown below.

- PJ1/OSC2/CLKOUT pin

| Register | CKCSR | | PCRJ | |
|---|---|---|---|---|
| **Bit Name** | **PMRJ1** | **PMRJ0** | **PCRJ1** | **Pin Function** |
| Setting Value | 0 | 0 | 0 | PJ1 input pin |
| | | | 1 | PJ1 output pin |
| | | 1 | X | Open |
| | 1 | 0 | X | CLKOUT output pin |
| | | 1 | X | OSC2 output pin |

[Legend] X: Don't care.

- PJ0/OSC1 pin

| Register | CKCSR | | PCRJ | |
|---|---|---|---|---|
| **Bit Name** | **PMRJ1** | **PMRJ0** | **PCRJ0** | **Pin Function** |
| Setting Value | X | 0 | 0 | PJ0 input pin |
| | | | 1 | PJ0 output pin |
| | 0 | 1 | X | OSC1 (clock input) |
| | 1 | 1 | X | OSC1 oscillation pin |

[Legend] X: Don't care.

Note: For details on external clock input, also see 5.2.4, Clock Control/Status Register (CKCSR).

# Section 10 Realtime Clock (RTC)

The realtime clock (RTC) is a timer used to count time ranging from a second to a week. Figure 10.1 shows the block diagram of the RTC.

## 10.1 Features

- Counts seconds, minutes, hours, and day-of-week
- Start/stop function
- Reset function
- Readable/writable counter of seconds, minutes, hours, and day-of-week with BCD codes
- Periodic (seconds, minutes, hours, days, and weeks) interrupts
- 8-bit free running counter
- Selection of clock source

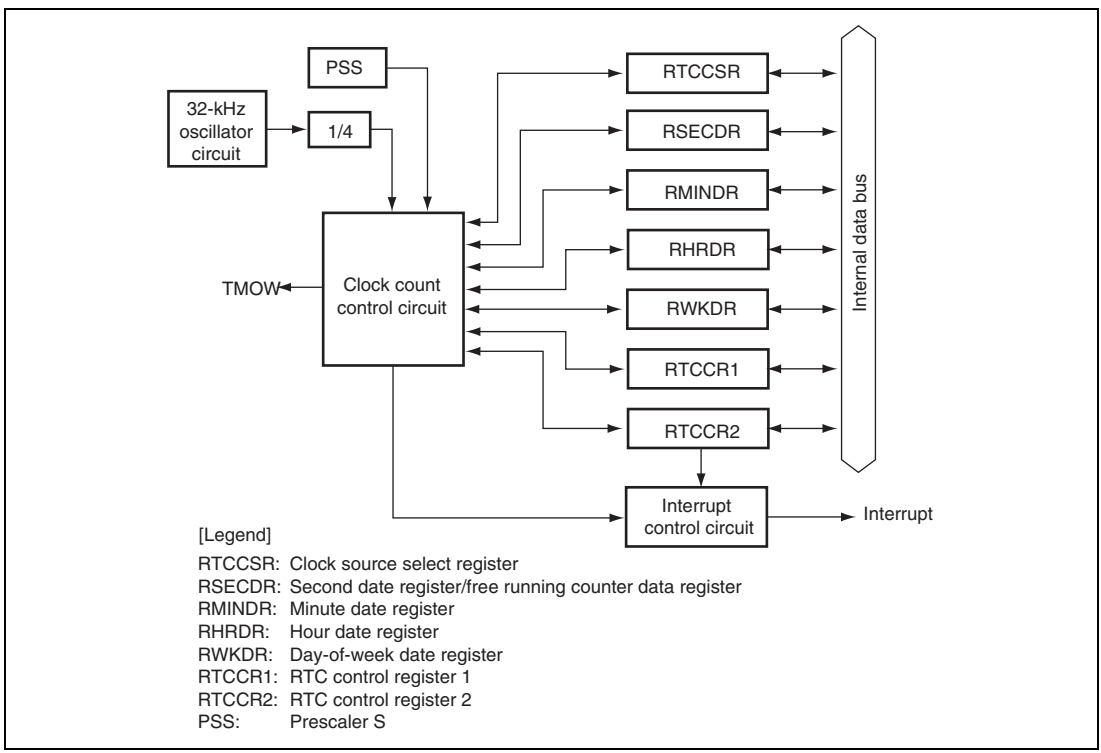



**Figure 10.1 Block Diagram of RTC**

## 10.2 Input/Output Pin

Table 10.1 shows the RTC input/output pin.

**Table 10.1 Pin Configuration**

| Name | Abbreviation | I/O | Function |
|---|---|---|---|
| Clock output | TMOW | Output | RTC divided clock output |

## 10.3 Register Descriptions

The RTC has the following registers.

- Second data register/free running counter data register (RSECDR)
- Minute data register (RMINDR)
- Hour data register (RHRDR)
- Day-of-week data register (RWKDR)
- RTC control register 1 (RTCCR1)
- RTC control register 2 (RTCCR2)
- Clock source select register (RTCCSR)

### 10.3.1 Second Data Register/Free Running Counter Data Register (RSECDR)

RSECDR counts the BCD-coded second value. The setting range is decimal 00 to 59. It is an 8-bit read register used as a counter, when it operates as a free running counter. For more information on reading seconds, minutes, hours, and day-of-week, see section 10.4.3, Data Reading Procedure.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | BSY | — | R | RTC Busy<br>This bit is set to 1 when the RTC is updating (operating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | SC12 | — | R/W | Counting Ten's Position of Seconds |
| 5 | SC11 | — | R/W | Counts on 0 to 5 for 60-second counting. |
| 4 | SC10 | — | R/W | |
| 3 | SC03 | — | R/W | Counting One's Position of Seconds |
| 2 | SC02 | — | R/W | Counts on 0 to 9 once per second. When a carry is generated, 1 is added to the ten's position. |
| 1 | SC01 | — | R/W | |
| 0 | SC00 | — | R/W | |

### 10.3.2 Minute Data Register (RMINDR)

RMINDR counts the BCD-coded minute value on the carry generated once per minute by the RSECDR counting. The setting range is decimal 00 to 59.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | BSY | — | R | RTC Busy<br>This bit is set to 1 when the RTC is updating (operating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted. |
| 6 | MN12 | — | R/W | Counting Ten's Position of Minutes |
| 5 | MN11 | — | R/W | Counts on 0 to 5 for 60-minute counting. |
| 4 | MN10 | — | R/W | |
| 3 | MN03 | — | R/W | Counting One's Position of Minutes |
| 2 | MN02 | — | R/W | Counts on 0 to 9 once per minute. When a carry is generated, 1 is added to the ten's position. |
| 1 | MN01 | — | R/W | |
| 0 | MN00 | — | R/W | |

### 10.3.3 Hour Data Register (RHRDR)

RHRDR counts the BCD-coded hour value on the carry generated once per hour by RMINDR. The setting range is either decimal 00 to 11 or 00 to 23 by the selection of the 12/24 bit in RTCCR1.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | BSY | — | R | RTC Busy<br>This bit is set to 1 when the RTC is updating (operating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted. |
| 6 | — | 0 | — | Reserved<br>This bit is always read as 0. |
| 5 | HR11 | — | R/W | Counting Ten's Position of Hours |
| 4 | HR10 | — | R/W | Counts on 0 to 2 for ten's position of hours. |
| 3 | HR03 | — | R/W | Counting One's Position of Hours |
| 2 | HR02 | — | R/W | Counts on 0 to 9 once per hour. When a carry is generated, 1 is added to the ten's position. |
| 1 | HR01 | — | R/W | |
| 0 | HR00 | — | R/W | |

### 10.3.4 Day-of-Week Data Register (RWKDR)

RWKDR counts the BCD-coded day-of-week value on the carry generated once per day by RHRDR. The setting range is decimal 0 to 6 using bits WK2 to WK0.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | BSY | — | R | RTC Busy<br>This bit is set to 1 when the RTC is updating (operating) the values of second, minute, hour, and day-of-week data registers. When this bit is 0, the values of second, minute, hour, and day-of-week data registers must be adopted. |
| 6 to 3 | — | All 0 | — | Reserved<br>These bits are always read as 0. |
| 2 | WK2 | — | R/W | Day-of-Week Counting |
| 1 | WK1 | — | R/W | Day-of-week is indicated with a binary code |
| 0 | WK0 | — | R/W | 000: Sunday<br>001: Monday<br>010: Tuesday<br>011: Wednesday<br>100: Thursday<br>101: Friday<br>110: Saturday<br>111: Reserved (setting prohibited) |

### 10.3.5 RTC Control Register 1 (RTCCR1)

RTCCR1 controls start/stop and reset of the clock timer. For the definition of time expression, see figure 10.2.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | RUN | — | R/W | RTC Operation Start<br>0: Stops RTC operation<br>1: Starts RTC operation |
| 6 | 12/24 | — | R/W | Operating Mode<br>0: RTC operates in 12-hour mode. RHRDR counts on 0 to 11.<br>1: RTC operates in 24-hour mode. RHRDR counts on 0 to 23. |
| 5 | PM | — | R/W | a.m./p.m.<br>0: Indicates a.m. when RTC is in the 12-hour mode.<br>1: Indicates p.m. when RTC is in the 12-hour mode. |
| 4 | RST | 0 | R/W | Reset<br>0: Normal operation<br>1: Resets registers and control circuits except RTCCSR and this bit. Clear this bit to 0 after having been set to 1. |
| 3 | INT | — | R/W | Interrupt Generation Timing<br>0: Generates a second, minute, hour, or day-of-week periodic interrupt during RTC busy period.<br>1: Generates a second, minute, hour, or day-of-week periodic interrupt immediately after completing RTC busy period. |
| 2 to 0 | — | All 0 | — | Reserved<br>These bits are always read as 0. |

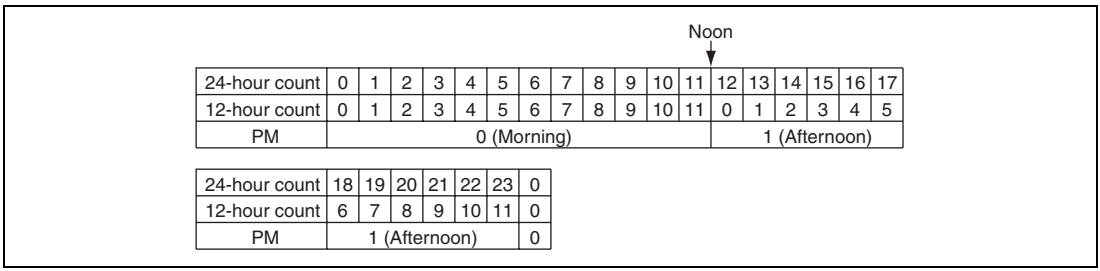



**Figure 10.2 Definition of Time Expression**

### 10.3.6 RTC Control Register 2 (RTCCR2)

RTCCR2 controls RTC periodic interrupts of weeks, days, hours, minutes, and seconds. Enabling interrupts of weeks, days, hours, minutes, and seconds sets the IRRTA flag to 1 in the interrupt flag register 1 (IRR1) when an interrupt occurs. It also controls an overflow interrupt of a free running counter when RTC operates as a free running counter.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | — | All 0 | — | Reserved<br>These bits are always read as 0. |
| 5 | FOIE | — | R/W | Free Running Counter Overflow Interrupt Enable<br>0: Disables an overflow interrupt<br>1: Enables an overflow interrupt |
| 4 | WKIE | — | R/W | Week Periodic Interrupt Enable<br>0: Disables a week periodic interrupt<br>1: Enables a week periodic interrupt |
| 3 | DYIE | — | R/W | Day Periodic Interrupt Enable<br>0: Disables a day periodic interrupt<br>1: Enables a day periodic interrupt |
| 2 | HRIE | — | R/W | Hour Periodic Interrupt Enable<br>0: Disables an hour periodic interrupt<br>1: Enables an hour periodic interrupt |
| 1 | MNIE | — | R/W | Minute Periodic Interrupt Enable<br>0: Disables a minute periodic interrupt<br>1: Enables a minute periodic interrupt |
| 0 | SEIE | — | R/W | Second Periodic Interrupt Enable<br>0: Disables a second periodic interrupt<br>1: Enables a second periodic interrupt |

### 10.3.7 Clock Source Select Register (RTCCSR)

RTCCSR selects clock source. A free running counter controls start/stop of counter operation by the RUN bit in RTCCR1. When a clock other than 32.768 kHz is selected, the RTC is disabled and operates as an 8-bit free running counter. When the RTC operates as an 8-bit free running counter, RSECDR enables counter values to be read. An interrupt can be generated by setting 1 to the FOIE bit in RTCCR2 and enabling an overflow interrupt of the free running counter. A clock in which the system clock is divided by 32, 16, 8, or 4 is output in active or sleep mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | — | Reserved<br>This bit is always read as 0. |
| 6 | RCS6 | 0 | R/W | Clock Output Selection |
| 5 | RCS5 | 0 | R/W | Selects a clock output from the TMOW pin when setting TMOW in PMR1 to 1.<br>00: ϕ/4<br>01: ϕ/8<br>10: ϕ/16<br>11: ϕ/32 |
| 4 | — | 0 | — | Reserved<br>This bit is always read as 0. |
| 3 | RCS3 | 1 | R/W | Clock Source Selection |
| 2 | RCS2 | 0 | R/W | 0000: ϕ/8····· Free running counter operation |
| 1 | RCS1 | 0 | R/W | 0001: ϕ/32····· Free running counter operation |
| 0 | RCS0 | 0 | R/W | 0010: ϕ/128····· Free running counter operation<br>0011: ϕ/256····· Free running counter operation<br>0100: ϕ/512····· Free running counter operation<br>0101: ϕ/2048····· Free running counter operation<br>0110: ϕ/4096····· Free running counter operation<br>0111: ϕ/8192····· Free running counter operation<br>1XXX: 32.768 kHz···RTC operation |

[Legend]
X: Don't care

## 10.4 Operation

### 10.4.1 Initial Settings of Registers after Power-On

The RTC registers that store second, minute, hour, and day-of week data are not reset by a $\overline{\text{RES}}$ input. Therefore, all registers must be set to their initial values after power-on. Once the register settings are made, the RTC provides an accurate time as long as power is supplied regardless of a $\overline{\text{RES}}$ input.

### 10.4.2 Initial Setting Procedure

Figure 10.3 shows the procedure for the initial setting of the RTC. To set the RTC again, also follow this procedure.

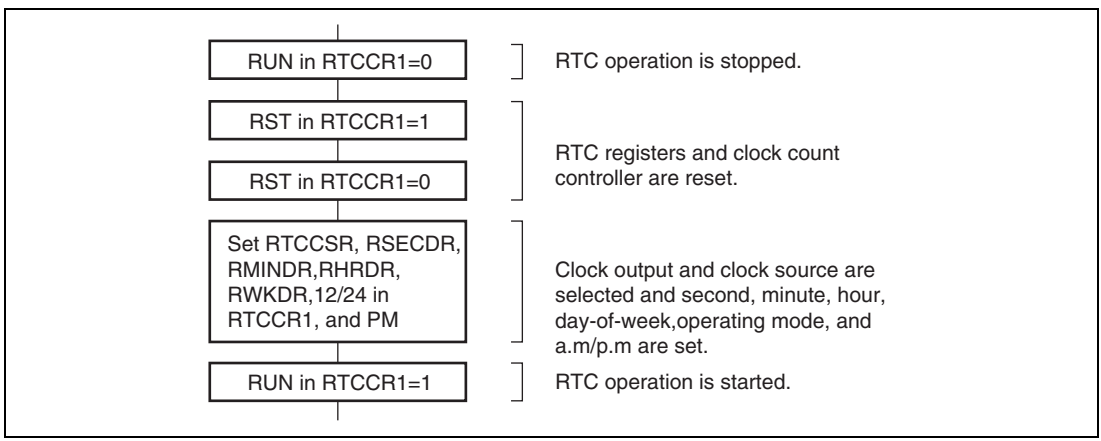



**Figure 10.3 Initial Setting Procedure**

### 10.4.3 Data Reading Procedure

When the seconds, minutes, hours, or day-of-week datum is updated while time data is being read, the data obtained may not be correct, and so the time data must be read again. Figure 10.4 shows an example in which correct data is not obtained. In this example, since only RSECDR is read after data update, about 1-minute inconsistency occurs.

To avoid reading in this timing, the following processing must be performed.

1. Check the setting of the BSY bit, and when the BSY bit changes from 1 to 0, read from the second, minute, hour, and day-of-week registers. When about 62.5 ms is passed after the BSY bit is set to 1, the registers are updated, and the BSY bit is cleared to 0.
2. Making use of interrupts, read from the second, minute, hour, and day-of week registers after the IRRTA flag in IRR1 is set to 1 and the BSY bit is confirmed to be 0.
3. Read from the second, minute, hour, and day-of week registers twice in a row, and if there is no change in the read data, the read data is used.

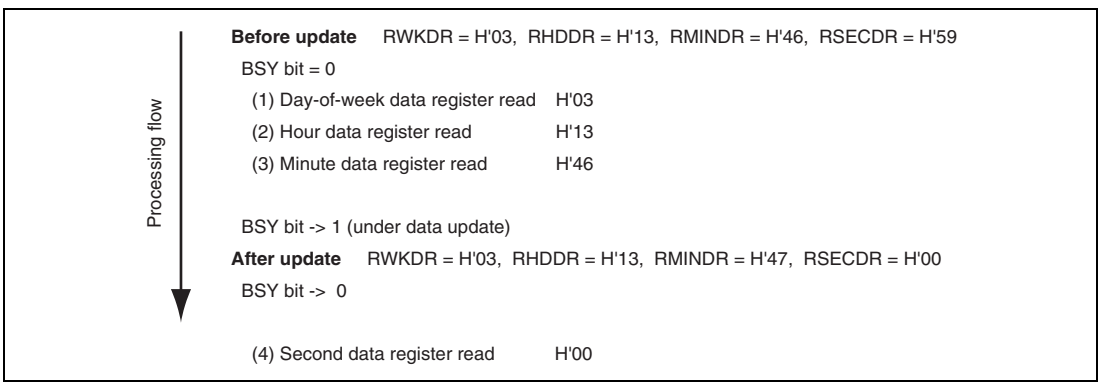



**Figure 10.4 Example: Reading of Inaccurate Time Data**

## 10.5 Interrupt Sources

There are five kinds of RTC interrupts: week interrupts, day interrupts, hour interrupts, minute interrupts, and second interrupts.

When using an interrupt, initiate the RTC last after other registers are set. Do not set multiple interrupt enable bits in RTCCR2 simultaneously to 1.

When an interrupt request of the RTC occurs, the IRRTA flag in IRR1 is set to 1. When clearing the flag, write 0.

**Table 10.2 Interrupt Sources**

| Interrupt Name | Interrupt Source | Interrupt Enable Bit |
|---|---|---|
| Overflow interrupt | Occurs when the free running counter is overflowed. | FOIE |
| Week periodic interrupt | Occurs every week when the day-of-week date register value becomes 0. | WKIE |
| Day periodic interrupt | Occurs every day when the day-of-week date register is counted. | DYIE |
| Hour periodic interrupt | Occurs every hour when the hour date register is counted. | HRIE |
| Minute periodic interrupt | Occurs every minute when the minute date register is counted. | MNIE |
| Second periodic interrupt | Occurs every second when the second date register is counted. | SCIE |

# Section 11 Timer B1

Timer B1 is an 8-bit timer that increments each time a clock pulse is input. This timer has two operating modes, interval and auto reload. Figure 11.1 shows a block diagram of timer B1.

## 11.1 Features

- Selection of seven internal clock sources (ϕ/8192, ϕ/2048, ϕ/512, ϕ/256, ϕ/64, ϕ/16, and ϕ/4) or an external clock (can be used to count external events).
- An interrupt is generated when the counter overflows.

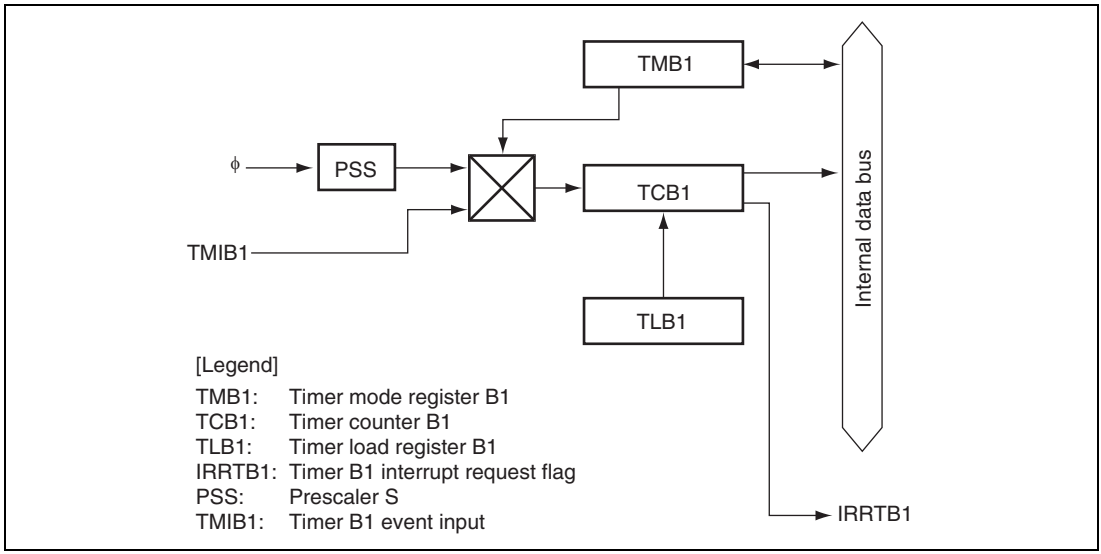



**Figure 11.1 Block Diagram of Timer B1**

## 11.2 Input/Output Pin

Table 11.1 shows the timer B1 pin configuration.

**Table 11.1 Pin Configuration**

| Name | Abbreviation | I/O | Function |
|---|---|---|---|
| Timer B1 event input | TMIB1 | Input | Event input to TCB1 |

## 11.3 Register Descriptions

The timer B1 has the following registers.

- Timer mode register B1 (TMB1)
- Timer counter B1 (TCB1)
- Timer load register B1 (TLB1)

### 11.3.1 Timer Mode Register B1 (TMB1)

TMB1 selects the auto-reload function and input clock.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TMB17 | 0 | R/W | Auto-Reload Function Select<br>0: Interval timer function selected<br>1: Auto-reload function selected |
| 6 to 3 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 2 | TMB12 | 0 | R/W | Clock Select |
| 1 | TMB11 | 0 | R/W | 000: Internal clock: $\phi$/8192 |
| 0 | TMB10 | 0 | R/W | 001: Internal clock: $\phi$/2048<br>010: Internal clock: $\phi$/512<br>011: Internal clock: $\phi$/256<br>100: Internal clock: $\phi$/64<br>101: Internal clock: $\phi$/16<br>110: Internal clock: $\phi$/4<br>111: External event (TMIB1): rising or falling edge*<br>Note: * The edge of the external event signal is selected by bit IEG1 in the interrupt edge select register 1 (IEGR1). See section 3.2.1, Interrupt Edge Select Register 1 (IEGR1), for details. Before setting TMB12 to TMB10 to 1, IRQ1 in the port mode register 1 (PMR1) should be set to 1. |

### 11.3.2 Timer Counter B1 (TCB1)

TCB1 is an 8-bit read-only up-counter, which is incremented by internal clock input. The clock source for input to this counter is selected by bits TMB12 to TMB10 in TMB1. TCB1 values can be read by the CPU at any time. When TCB1 overflows from H'FF to H'00 or to the value set in TLB1, the IRRTB1 flag in IRR2 is set to 1. TCB1 is allocated to the same address as TLB1. TCB1 is initialized to H'00.

### 11.3.3 Timer Load Register B1 (TLB1)

TLB1 is an 8-bit write-only register for setting the reload value of TCB1. When a reload value is set in TLB1, the same value is loaded into TCB1 as well, and TCB1 starts counting up from that value. When TCB1 overflows during operation in auto-reload mode, the TLB1 value is loaded into TCB1. Accordingly, overflow periods can be set within the range of 1 to 256 input clocks. TLB1 is allocated to the same address as TCB1. TLB1 is initialized to H'00.

## 11.4 Operation

### 11.4.1 Interval Timer Operation

When bit TMB17 in TMB1 is cleared to 0, timer B1 functions as an 8-bit interval timer. Upon reset, TCB1 is cleared to H'00 and bit TMB17 is cleared to 0, so up-counting and interval timing resume immediately. The operating clock of timer B1 is selected from seven internal clock signals output by prescaler S, or an external clock input at pin TMB1. The selection is made by bits TMB12 to TMB10 in TMB1.

After the count value in TMB1 reaches H'FF, the next clock signal input causes timer B1 to overflow, setting flag IRRTB1 in IRR2 to 1. If IENTB1 in IENR2 is 1, an interrupt is requested to the CPU.

At overflow, TCB1 returns to H'00 and starts counting up again. During interval timer operation (TMB17 = 0), when a value is set in TLB1, the same value is set in TCB1.

### 11.4.2 Auto-Reload Timer Operation

Setting bit TMB17 in TMB1 to 1 causes timer B1 to function as an 8-bit auto-reload timer. When a reload value is set in TLB1, the same value is loaded into TCB1, becoming the value from which TCB1 starts its count. After the count value in TCB1 reaches H'FF, the next clock signal input causes timer B1 to overflow. The TLB1 value is then loaded into TCB1, and the count continues from that value. The overflow period can be set within a range from 1 to 256 input clocks, depending on the TLB1 value.

The clock sources and interrupts in auto-reload mode are the same as in interval mode. In auto-reload mode (TMB17 = 1), when a new value is set in TLB1, the TLB1 value is also loaded into TCB1.

### 11.4.3 Event Counter Operation

Timer B1 can operate as an event counter in which TMIB1 is set to an event input pin. External event counting is selected by setting bits TMB12 to TMB10 in TMB1 to 1. TCB1 counts up at rising or falling edge of an external event signal input at pin TMB1.

When timer B1 is used to count external event input, bit IRQ1 in PMR1 should be set to 1 and IEN1 in IENR1 should be cleared to 0 to disable IRQ1 interrupt requests.

## 11.5 Timer B1 Operating Modes

Table 11.2 shows the timer B1 operating modes.

**Table 11.2 Timer B1 Operating Modes**

| Operating Mode | | Reset | Active | Sleep | Subactive | Subsleep | Standby |
|---|---|---|---|---|---|---|---|
| TCB1 | Interval | Reset | Functions | Functions | Halted | Halted | Halted |
| | Auto-reload | Reset | Functions | Functions | Halted | Halted | Halted |
| TMB1 | | Reset | Functions | Retained | Retained | Retained | Retained |

# Section 12 Timer V

Timer V is an 8-bit timer based on an 8-bit counter. Timer V counts external events. Compare-match signals with two registers can also be used to reset the counter, request an interrupt, or output a pulse signal with an arbitrary duty cycle. Counting can be initiated by a trigger input at the TRGV pin, enabling pulse output control to be synchronized to the trigger, with an arbitrary delay from the trigger input. Figure 12.1 shows a block diagram of timer V.

## 12.1 Features

- Choice of seven clock signals is available.
  Choice of six internal clock sources (ϕ/128, ϕ/64, ϕ/32, ϕ/16, ϕ/8, ϕ/4) or an external clock.
- Counter can be cleared by compare match A or B, or by an external reset signal. If the count stop function is selected, the counter can be halted when cleared.
- Timer output is controlled by two independent compare match signals, enabling pulse output with an arbitrary duty cycle, PWM output, and other applications.
- Three interrupt sources: compare match A, compare match B, timer overflow
- Counting can be initiated by trigger input at the TRGV pin. The rising edge, falling edge, or both edges of the TRGV input can be selected.

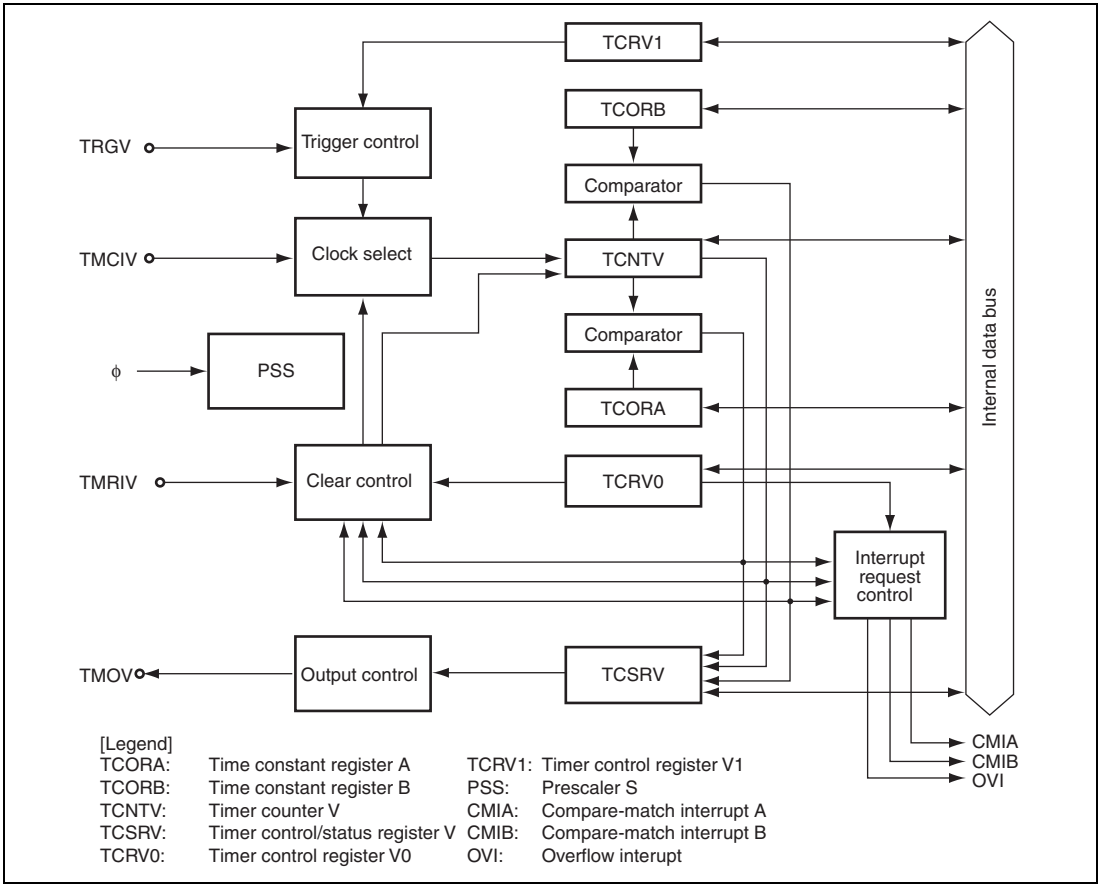



**Figure 12.1 Block Diagram of Timer V**

## 12.2 Input/Output Pins

Table 12.1 shows the timer V pin configuration.

**Table 12.1 Pin Configuration**

| Name | Abbreviation | I/O | Function |
|---|---|---|---|
| Timer V output | TMOV | Output | Timer V waveform output |
| Timer V clock input | TMCIV | Input | Clock input to TCNTV |
| Timer V reset input | TMRIV | Input | External input to reset TCNTV |
| Trigger input | TRGV | Input | Trigger input to initiate counting |

## 12.3 Register Descriptions

Time V has the following registers.

- Timer counter V (TCNTV)
- Timer constant register A (TCORA)
- Timer constant register B (TCORB)
- Timer control register V0 (TCRV0)
- Timer control/status register V (TCSRV)
- Timer control register V1 (TCRV1)

### 12.3.1 Timer Counter V (TCNTV)

TCNTV is an 8-bit up-counter. The clock source is selected by bits CKS2 to CKS0 in timer control register V0 (TCRV0). The TCNTV value can be read and written by the CPU at any time. TCNTV can be cleared by an external reset input signal, or by compare match A or B. The clearing signal is selected by bits CCLR1 and CCLR0 in TCRV0.

When TCNTV overflows, OVF is set to 1 in timer control/status register V (TCSRV).

TCNTV is initialized to H'00.

### 12.3.2 Time Constant Registers A, B (TCORA, TCORB)

TCORA and TCORB have the same function.

TCORA and TCORB are 8-bit readable/writable registers.

TCORA and TCNTV are compared at all times. When the TCORA and TCNTV contents match, CMFA is set to 1 in TCSRV. If CMIEA is also set to 1 in TCRV0, a CPU interrupt is requested. Note that they must not be compared during the T3 state of a TCORA write cycle.

Timer output from the TMOV pin can be controlled by the identifying signal (compare match A) and the settings of bits OS3 to OS0 in TCSRV.

TCORA and TCORB are initialized to H'FF.

### 12.3.3 Timer Control Register V0 (TCRV0)

TCRV0 selects the input clock signals of TCNTV, specifies the clearing conditions of TCNTV, and controls each interrupt request.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | CMIEB | 0 | R/W | Compare Match Interrupt Enable B<br>When this bit is set to 1, interrupt request from the CMFB bit in TCSRV is enabled. |
| 6 | CMIEA | 0 | R/W | Compare Match Interrupt Enable A<br>When this bit is set to 1, interrupt request from the CMFA bit in TCSRV is enabled. |
| 5 | OVIE | 0 | R/W | Timer Overflow Interrupt Enable<br>When this bit is set to 1, interrupt request from the OVF bit in TCSRV is enabled. |
| 4 | CCLR1 | 0 | R/W | Counter Clear 1 and 0 |
| 3 | CCLR0 | 0 | R/W | These bits specify the clearing conditions of TCNTV.<br>00: Clearing is disabled<br>01: Cleared by compare match A<br>10: Cleared by compare match B<br>11: Cleared on the rising edge of the TMRIV pin. The operation of TCNTV after clearing depends on TRGE in TCRV1. |
| 2 | CKS2 | 0 | R/W | Clock Select 2 to 0 |
| 1 | CKS1 | 0 | R/W | These bits select clock signals to input to TCNTV and the counting condition in combination with ICKS0 in TCRV1.<br>Refer to table 12.2. |
| 0 | CKS0 | 0 | R/W | |

**Table 12.2 Clock Signals to Input to TCNTV and Counting Conditions**

| TCRV0 | | | TCRV1 | |
|---|---|---|---|---|
| Bit 2 | Bit 1 | Bit 0 | Bit 0 | |
| CKS2 | CKS1 | CKS0 | ICKS0 | Description |
| 0 | 0 | 0 | — | Clock input prohibited |
| | | 1 | 0 | Internal clock: counts on $\phi$/4, falling edge |
| | | | 1 | Internal clock: counts on $\phi$/8, falling edge |
| | 1 | 0 | 0 | Internal clock: counts on $\phi$/16, falling edge |
| | | | 1 | Internal clock: counts on $\phi$/32, falling edge |
| | | 1 | 0 | Internal clock: counts on $\phi$/64, falling edge |
| | | | 1 | Internal clock: counts on $\phi$/128, falling edge |
| 1 | 0 | 0 | — | Clock input prohibited |
| | | 1 | — | External clock: counts on rising edge |
| | 1 | 0 | — | External clock: counts on falling edge |
| | | 1 | — | External clock: counts on rising and falling edge |

### 12.3.4 Timer Control/Status Register V (TCSRV)

TCSRV indicates the status flag and controls outputs by using a compare match.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | CMFB | 0 | R/W | Compare Match Flag B<br>Setting condition:<br>When the TCNTV value matches the TCORB value<br>Clearing condition:<br>After reading CMFB = 1, cleared by writing 0 to CMFB |
| 6 | CMFA | 0 | R/W | Compare Match Flag A<br>Setting condition:<br>When the TCNTV value matches the TCORA value<br>Clearing condition:<br>After reading CMFA = 1, cleared by writing 0 to CMFA |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | OVF | 0 | R/W | Timer Overflow Flag<br>Setting condition:<br>When TCNTV overflows from H'FF to H'00<br>Clearing condition:<br>After reading OVF = 1, cleared by writing 0 to OVF |
| 4 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 3 | OS3 | 0 | R/W | Output Select 3 and 2<br>These bits select an output method for the TOMV pin by the compare match of TCORB and TCNTV.<br>00: No change<br>01: 0 output<br>10: 1 output<br>11: Output toggles |
| 2 | OS2 | 0 | R/W | |
| 1 | OS1 | 0 | R/W | Output Select 1 and 0<br>These bits select an output method for the TOMV pin by the compare match of TCORA and TCNTV.<br>00: No change<br>01: 0 output<br>10: 1 output<br>11: Output toggles |
| 0 | OS0 | 0 | R/W | |

OS3 and OS2 select the output level for compare match B. OS1 and OS0 select the output level for compare match A. The two output levels can be controlled independently. After a reset, the timer output is 0 until the first compare match.

### 12.3.5 Timer Control Register V1 (TCRV1)

TCRV1 selects the edge at the TRGV pin, enables TRGV input, and selects the clock input to TCNTV.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 5 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 4 | TVEG1 | 0 | R/W | TRGV Input Edge Select |
| 3 | TVEG0 | 0 | R/W | These bits select the TRGV input edge.<br>00: TRGV trigger input is prohibited<br>01: Rising edge is selected<br>10: Falling edge is selected<br>11: Rising and falling edges are both selected |
| 2 | TRGE | 0 | R/W | TCNT starts counting up by the input of the edge which is selected by TVEG1 and TVEG0.<br>0: Disables starting counting-up TCNTV by the input of the TRGV pin and halting counting-up TCNTV when TCNTV is cleared by a compare match.<br>1: Enables starting counting-up TCNTV by the input of the TRGV pin and halting counting-up TCNTV when TCNTV is cleared by a compare match. |
| 1 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 0 | ICKS0 | 0 | R/W | Internal Clock Select 0<br>This bit selects clock signals to input to TCNTV in combination with CKS2 to CKS0 in TCRV0.<br>Refer to table 12.2. |

## 12.4 Operation

### 12.4.1 Timer V Operation

1. According to table 12.2, six internal/external clock signals output by prescaler S can be selected as the timer V operating clock signals. When the operating clock signal is selected, TCNTV starts counting-up. Figure 12.2 shows the count timing with an internal clock signal selected, and figure 12.3 shows the count timing with both edges of an external clock signal selected.
2. When TCNTV overflows (changes from H'FF to H'00), the overflow flag (OVF) in TCRV0 will be set. The timing at this time is shown in figure 12.4. An interrupt request is sent to the CPU when OVIE in TCRV0 is 1.
3. TCNTV is constantly compared with TCORA and TCORB. Compare match flag A or B (CMFA or CMFB) is set to 1 when TCNTV matches TCORA or TCORB, respectively. The compare-match signal is generated in the last state in which the values match. Figure 12.5 shows the timing. An interrupt request is generated for the CPU when CMIEA or CMIEB in TCRV0 is 1.
4. When a compare match A or B is generated, the TMOV responds with the output value selected by bits OS3 to OS0 in TCSRV. Figure 12.6 shows the timing when the output is toggled by compare match A.
5. When CCLR1 or CCLR0 in TCRV0 is 01 or 10, TCNTV can be cleared by the corresponding compare match. Figure 12.7 shows the timing.
6. When CCLR1 or CCLR0 in TCRV0 is 11, TCNTV can be cleared by the rising edge of the input of TMRIV pin. A TMRIV input pulse-width of at least 1.5 system clocks is necessary. Figure 12.8 shows the timing.
7. When a counter-clearing source is generated with TRGE in TCRV1 set to 1, the counting-up is halted as soon as TCNTV is cleared. TCNTV resumes counting-up when the edge selected by TVEG1 or TVEG0 in TCRV1 is input from the TGRV pin.

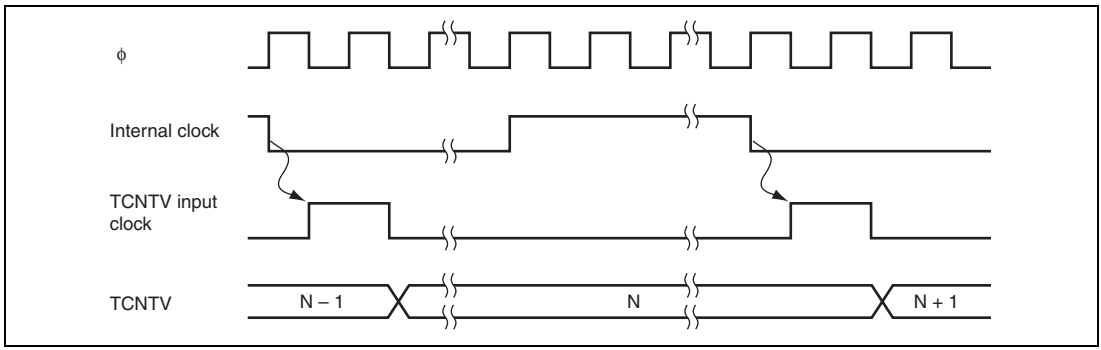



**Figure 12.2 Increment Timing with Internal Clock**

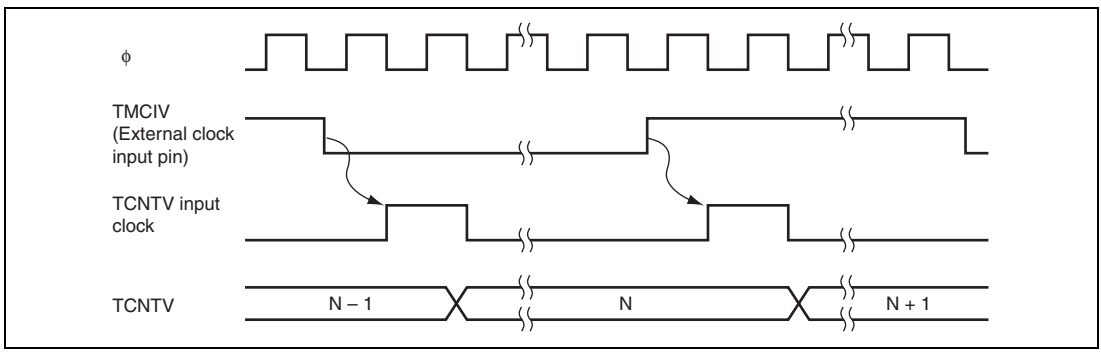



**Figure 12.3 Increment Timing with External Clock**

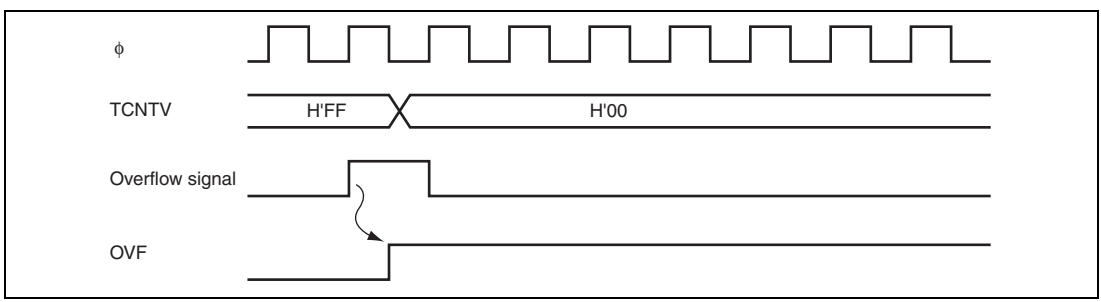



**Figure 12.4 OVF Set Timing**

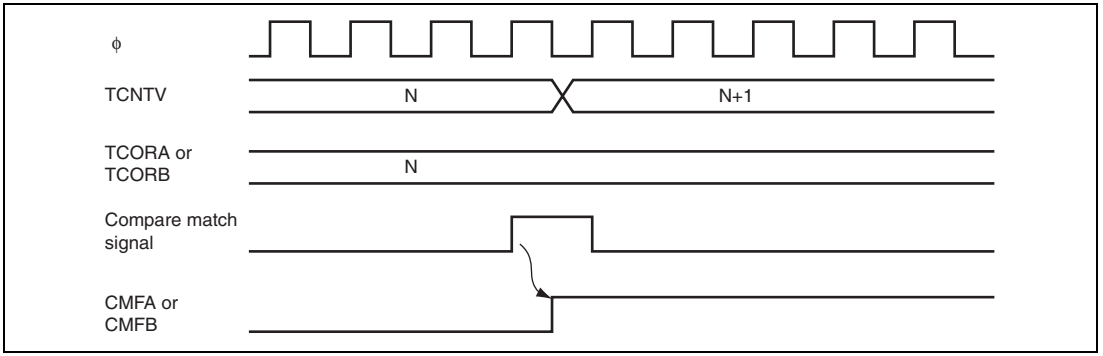



**Figure 12.5 CMFA and CMFB Set Timing**

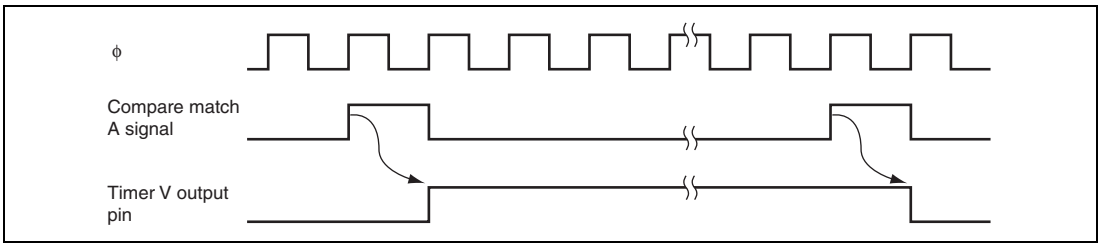



**Figure 12.6 TMOV Output Timing**

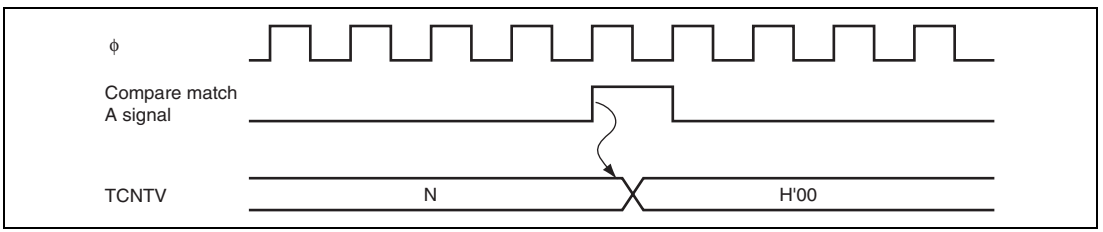



**Figure 12.7 Clear Timing by Compare Match**

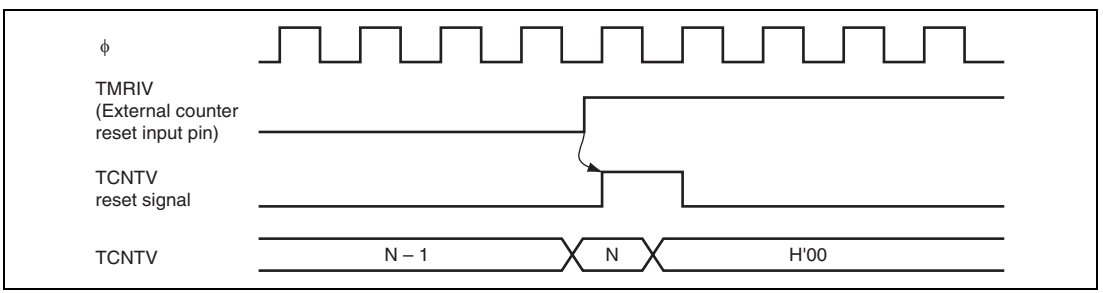



**Figure 12.8 Clear Timing by TMRIV Input**

## 12.5 Timer V Application Examples

### 12.5.1 Pulse Output with Arbitrary Duty Cycle

Figure 12.9 shows an example of output of pulses with an arbitrary duty cycle.

1. Set bits CCLR1 and CCLR0 in TCRV0 so that TCNTV will be cleared by compare match with TCORA.
2. Set bits OS3 to OS0 in TCSRV so that the output will go to 1 at compare match with TCORA and to 0 at compare match with TCORB.
3. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.
4. With these settings, a waveform is output without further software intervention, with a period determined by TCORA and a pulse width determined by TCORB.

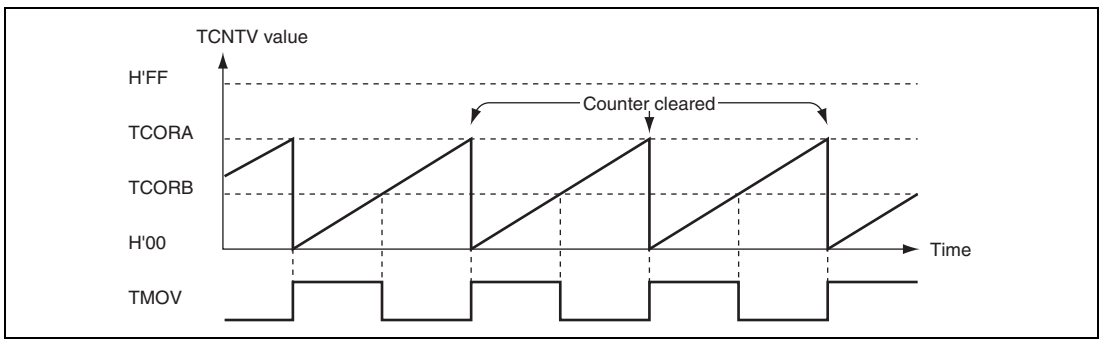



**Figure 12.9 Pulse Output Example**

### 12.5.2 Pulse Output with Arbitrary Pulse Width and Delay from TRGV Input

The trigger function can be used to output a pulse with an arbitrary pulse width at an arbitrary delay from the TRGV input, as shown in figure 12.10. To set up this output:

1. Set bits CCLR1 and CCLR0 in TCRV0 so that TCNTV will be cleared by compare match with TCORB.
2. Set bits OS3 to OS0 in TCSRV so that the output will go to 1 at compare match with TCORA and to 0 at compare match with TCORB.
3. Set bits TVEG1 and TVEG0 in TCRV1 and set TRGE to select the falling edge of the TRGV input.
4. Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.
5. With these settings, a pulse waveform will be output without further software intervention, with a delay determined by TCORA from the TRGV input, and a pulse width determined by (TCORB – TCORA).

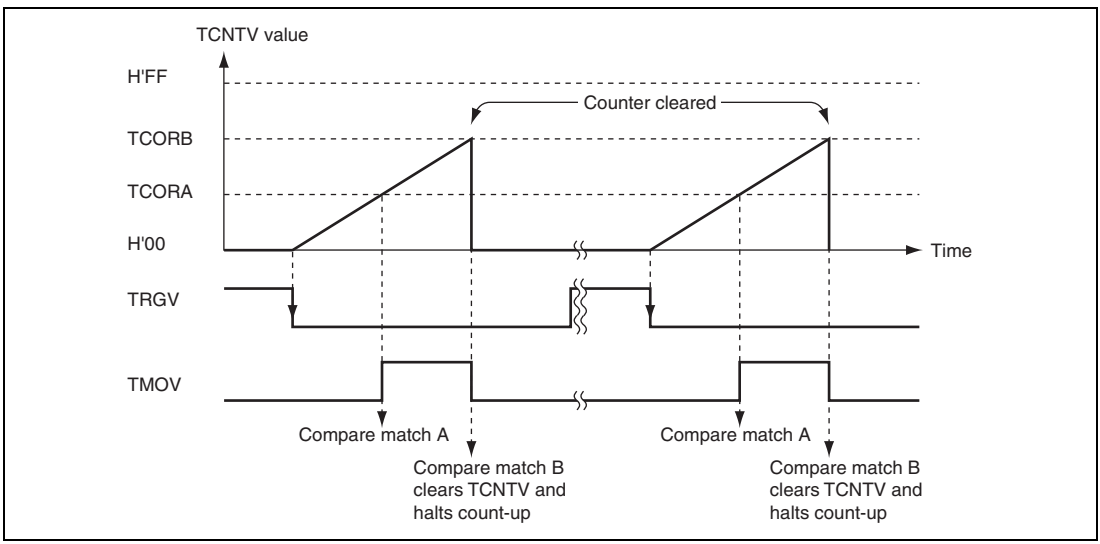



**Figure 12.10 Example of Pulse Output Synchronized to TRGV Input**

## 12.6 Usage Notes

The following types of contention or operation can occur in timer V operation.

1. Writing to registers is performed in the T3 state of a TCNTV write cycle. If a TCNTV clear signal is generated in the T3 state of a TCNTV write cycle, as shown in figure 12.11, clearing takes precedence and the write to the counter is not carried out. If counting-up is generated in the T3 state of a TCNTV write cycle, writing takes precedence.
2. If a compare match is generated in the T3 state of a TCORA or TCORB write cycle, the write to TCORA or TCORB takes precedence and the compare match signal is inhibited. Figure 12.12 shows the timing.
3. If compare matches A and B occur simultaneously, any conflict between the output selections for compare match A and compare match B is resolved by the following priority: toggle output > output 1 > output 0.
4. Depending on the timing, TCNTV may be incremented by a switch between different internal clock sources. When TCNTV is internally clocked, an increment pulse is generated from the falling edge of an internal clock signal, that is divided system clock (ϕ). Therefore, as shown in figure 12.3 the switch is from a high clock signal to a low clock signal, the switchover is seen as a falling edge, causing TCNTV to increment. TCNTV can also be incremented by a switch between internal and external clocks.

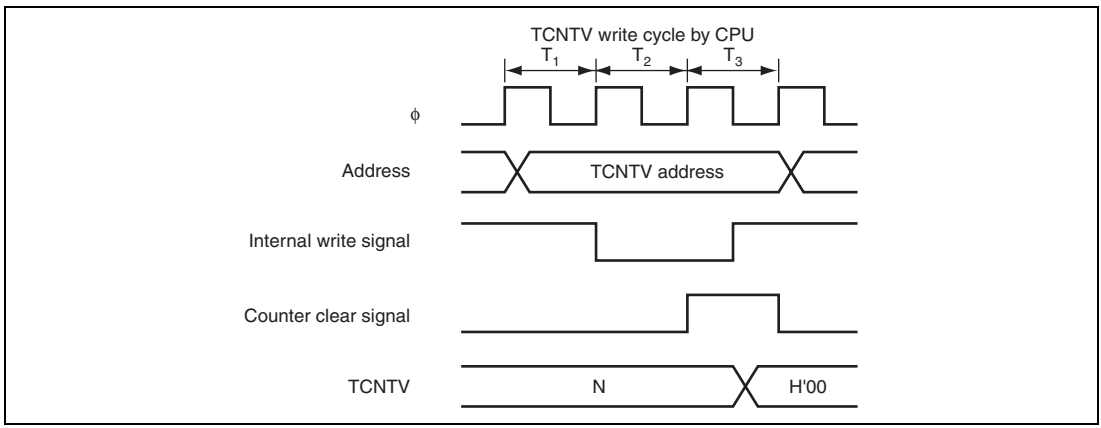



**Figure 12.11 Contention between TCNTV Write and Clear**

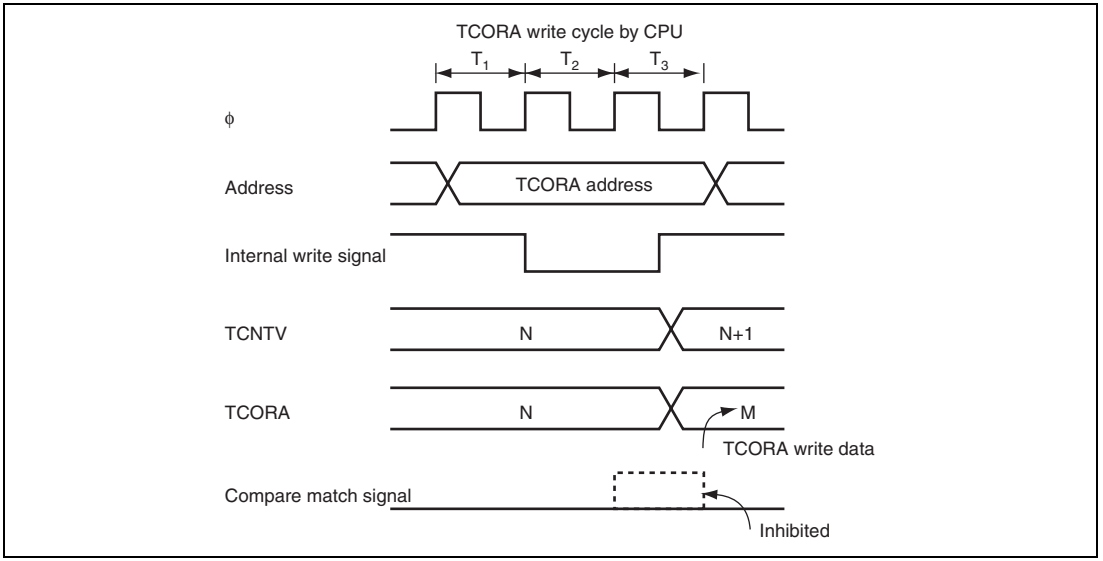



**Figure 12.12 Contention between TCORA Write and Compare Match**

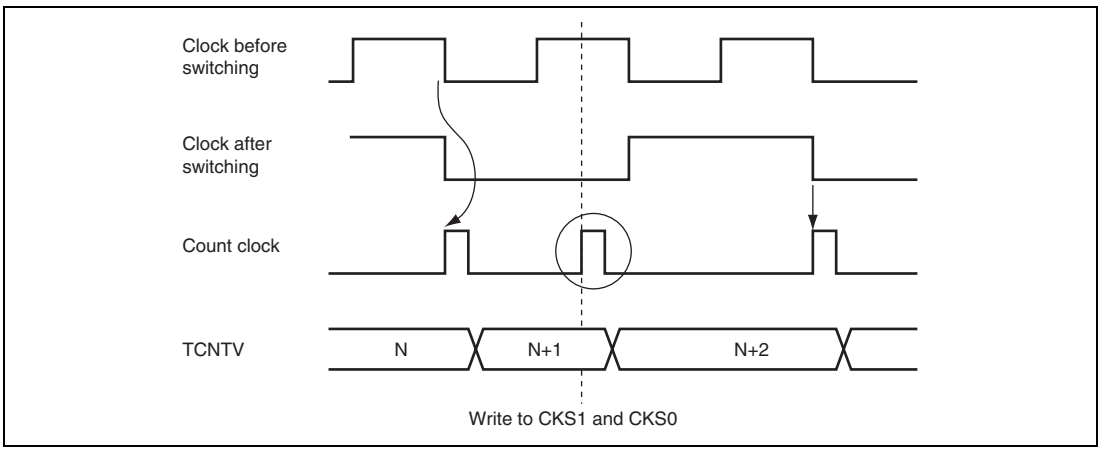



**Figure 12.13 Internal Clock Switching and TCNTV Operation**

# Section 13 Timer RC

Timer RC is a 16-bit timer having output compare and input capture functions. Timer RC can count external events and output pulses with a desired duty cycle using the compare match function between the timer counter and four general registers. Thus, it can be applied to various systems.

## 13.1 Features

- Selection of seven counter clock sources
  Six internal clocks (ϕ, ϕ/2, ϕ/4, ϕ/8, ϕ/32, and ϕ40M which is a 40-MHz/32-MHz clock derived from the on-chip oscillator) and an external clock (for counting external events)
- Capability to process up to four pulse outputs or four pulse inputs
- Four general registers
  — Can be used as output compare or input capture registers independently
  — Can be used as buffer registers for the output compare or input capture registers
- Timer inputs and outputs
  — Timer mode
    Waveform output by compare match (Selection of 0 output, 1 output, or toggle output)
    Input capture function (Rising edge, falling edge, or both edges)
    Counter clearing function (Counters can be cleared by compare match)
  — PWM mode
    Generates up to three-phase PWM output with desired duty cycles.
  — PWM2 mode
    Generates pulses with a desired period and duty cycle.
- Any initial timer output value can be set
- Five interrupt sources
  Four compare match/input capture interrupts and an overflow interrupt.

Table 13.1 summarizes the timer RC functions, and figure 13.1 shows a block diagram of timer RC.

**Table 13.1 Timer RC Functions**

| Item | | Counter | Input/Output Pins | | | |
|---|---|---|---|---|---|---|
| | | | FTIOA | FTIOB | FTIOC | FTIOD |
| Count clock | | Internal clocks: φ, φ/2, φ/4, φ/8, φ/32, and φ40M<br>External clock: FTCI | | | | |
| General registers (output compare/input capture registers) | | Period specified in GRA | GRA | GRB | GRC (buffer register for GRA in buffer mode) | GRD (buffer register for GRB in buffer mode) |
| Counter clearing function | | GRA compare match | GRA compare match | — | — | — |
| | | TGRC input | — | — | — | — |
| Initial output value setting function | | — | Yes | Yes | Yes | Yes |
| Buffer function | | — | Yes | Yes | — | — |
| Compare match output | 0 | — | Yes | Yes | Yes | Yes |
| | 1 | — | Yes | Yes | Yes | Yes |
| | Toggle | — | Yes | Yes | Yes | Yes |
| Input capture function | | — | Yes | Yes | Yes | Yes |
| PWM mode | | — | — | Yes | Yes | Yes |
| PWM2 mode | | — | — | Yes | — | — |
| Interrupt sources | | Overflow | Compare match/input capture | Compare match/input capture | Compare match/input capture | Compare match/input capture |

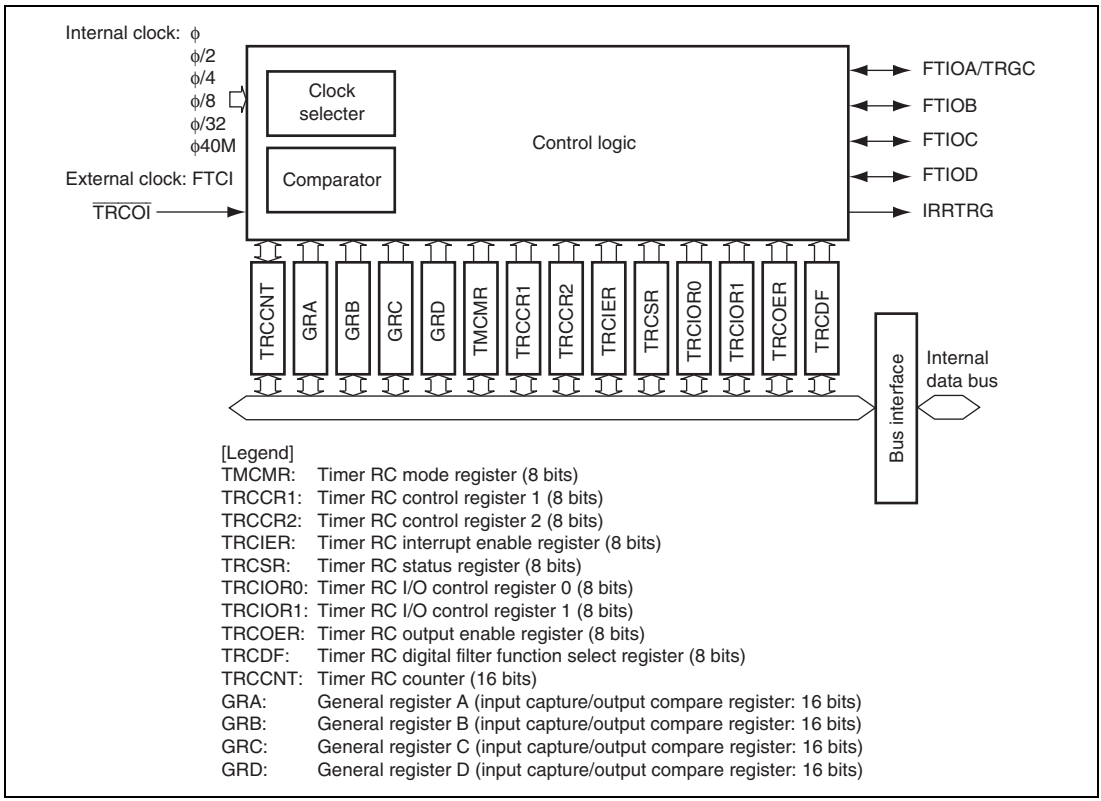



**Figure 13.1 Timer RC Block Diagram**

## 13.2 Input/Output Pins

Table 13.2 summarizes the timer RC pins.

**Table 13.2 Pin Configuration**

| Name | Symbol | Input/ Output | Function |
|---|---|---|---|
| External clock input | FTCI | Input | External clock input pin |
| Input capture/ output compare A | FTIOA/TRGC | I/O | Output pin for GRA output compare/ input pin for GRA input capture/ external trigger input pin (TRGC) |
| Input capture/ output compare B | FTIOB | I/O | Output pin for GRB output compare/ input pin for GRB input capture/ PWM output pin in PWM mode |
| Input capture/ output compare C | FTIOC | I/O | Output pin for GRC output compare/ input pin for GRC input capture/ or PWM output pin in PWM mode |
| Input capture/ output compare D | FTIOD | I/O | Output pin for GRD output compare/ input pin for GRD input capture/ or PWM output pin in PWM mode |
| Timer output control input | $\overline{\text{TRCOI}}$ | Input | Input pin for timer output disabling signal |

## 13.3 Register Descriptions

Timer RC has the following registers.

- Timer RC mode register (TRCMR)
- Timer RC control register 1 (TRCCR1)
- Timer RC control register 2 (TRCCR2)
- Timer RC interrupt enable register (TRCIER)
- Timer RC status register (TRCSR)
- Timer RC I/O control register 0 (TRCIOR0)
- Timer RC I/O control register 1 (TRCIOR1)
- Timer RC output enable register (TRCOER)
- Timer RC digital filtering function select register (TRCDF)
- Timer RC counter (TRCCNT)
- General Registers A to D (GRA to GRD)
- General register A (GRA)
- General register B (GRB)
- General register C (GRC)
- General register D (GRD)

### 13.3.1 Timer RC Mode Register (TRCMR)

TRCMR selects the general register functions and the timer output mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | CTS | 0 | R/W | Counter Start<br>TRCCNT stops counting when this bit is 0, while it performs counting when this bit is 1.<br>[Setting condition]<br>• When 1 is written in CTS<br>[Clearing conditions]<br>• When 0 is written in CTS<br>• In PWM2 mode, when the CSTP bit in TRCCR2 is set to 1 and a compare match signal is generated |
| 6 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 5 | BUFEB | 0 | R/W | Buffer Operation B<br>Selects the GRD function.<br>0: GRD functions as an input capture/output compare register<br>1: GRD functions as the buffer register for GRB |
| 4 | BUFEA | 0 | R/W | Buffer Operation A<br>Selects the GRC function.<br>0: GRC functions as an input capture/output compare register<br>1: GRC functions as the buffer register for GRA |
| 3 | PWM2 | 1 | R/W | PWM2 Mode<br>Selects the output mode of the FTIOB pin.<br>0: Functions in PWM2 mode.<br>The following settings are invalid: TRCIOR0, TRCIOR1, and the PWMB, PWMC, and PWMD bits in TRCMR.<br>1: Functions in timer mode or PWM mode.<br>The following settings are valid: TRCIOR0, TRCIOR1, and the PWMB, PWMC, and PWMD bits in TRCMR. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | PWMD | 0 | R/W | PWM Mode D<br>Selects the output mode of the FTIOD pin.<br>0: Functions in timer mode<br>1: Functions in PWM mode |
| 1 | PWMC | 0 | R/W | PWM Mode C<br>Selects the output mode of the FTIOC pin.<br>0: Functions in timer mode<br>1: Functions in PWM mode |
| 0 | PWMB | 0 | R/W | PWM Mode B<br>Selects the output mode of the FTIOB pin.<br>0: Functions in timer mode<br>1: Functions in PWM mode |

### 13.3.2 Timer RC Control Register 1 (TRCCR1)

TRCCR1 specifies the source of the counter clock, clearing conditions, and initial output levels of TRCCNT.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | CCLR | 0 | R/W | Counter Clear<br>The TRCCNT value is cleared by compare match A when this bit is 1. When it is 0, TRCCNT functions as a free-running counter. |
| 6 | CKS2 | 0 | R/W | Clock Select 2 to 0 |
| 5 | CKS1 | 0 | R/W | Select the source of the clock input to TRCCNT. |
| 4 | CKS0 | 0 | R/W | 000: TRCCNT counts the internal clock $\phi$<br>001: TRCCNT counts the internal clock $\phi/2$<br>010: TRCCNT counts the internal clock $\phi/4$<br>011: TRCCNT counts the internal clock $\phi/8$<br>100: TRCCNT counts the internal clock $\phi/32$<br>101: TRCCNT counts the rising edge of the external event (FTCI)<br>110: TRCCNT counts the internal clock $\phi$40M<br>111: Reserved (setting prohibited)<br>When the internal clock ($\phi$) is selected, TRCCNT counts the subclock in subactive or subsleep mode. *<br>Note: * When selecting the internal clock $\phi$40M, the on-chip oscillator should be in operation. When switching the clock, the counter should be halted. |
| 3 | TOD | 0 | R/W | Timer Output Level Setting D<br>Sets the output value of the FTIOD pin until the first compare match D is generated. In PWM mode, controls the output polarity of the FTIOD pin.<br>0: Output value is 0*<br>1: Output value is 1* |
| 2 | TOC | 0 | R/W | Timer Output Level Setting C<br>Sets the output value of the FTIOC pin until the first compare match C is generated. In PWM mode, controls the output polarity of the FTIOC pin.<br>0: Output value is 0*<br>1: Output value is 1* |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | TOB | 0 | R/W | Timer Output Level Setting B<br>Sets the output value of the FTIOB pin until the first compare match B is generated. In PWM mode, controls the output polarity of the FTIOB pin.<br>0: Output value is 0*<br>1: Output value is 1* |
| 0 | TOA | 0 | R/W | Timer Output Level Setting A<br>Sets the output value of the FTIOA pin until the first compare match A is generated.<br>0: Output value is 0*<br>1: Output value is 1* |

[Legend]

X: Don't care.

Note: * The change of the setting is immediately reflected in the output value.

### 13.3.3 Timer RC Control Register 2 (TRCCR2)

TRCCR2 specifies the edge of the TRGC signal and an input enable.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TCEG1 | 0 | R/W | TRGC Input Edge Select |
| 6 | TCEG0 | 0 | R/W | These bits select the input edge of the TRGC signal. This function is only enabled when the PWM2 bit in TRCMR is set to 0.<br>00: A trigger input on TRGC is disabled<br>01: The rising edge is selected<br>10: The falling edge is selected<br>11: Both edges are selected |
| 5 | CSTP | 0 | R/W | Specifies whether TRCCNT counting up is halted or continued by the compare match A signal. This function is only enabled when the PWM2 bit in TRCMR is set to 0.<br>0: TRCCNT counting up is continued<br>1: TRCCNT counting up is halted |
| 4 to 0 | — | All 1 | — | Reserved<br>These bits are always read as 1. |

### 13.3.4 Timer RC Interrupt Enable Register (TRCIER)

TRCIER controls the timer RC interrupt request.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | OVIE | 0 | R/W | Timer Overflow Interrupt Enable<br>When this bit is set to 1, an FOVI interrupt requested by the OVF flag in TRCSR is enabled. |
| 6 to 4 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 3 | IMIED | 0 | R/W | Input Capture/Compare Match Interrupt Enable D<br>When this bit is set to 1, an IMID interrupt requested by the IMFD flag in TRCSR is enabled. |
| 2 | IMIEC | 0 | R/W | Input Capture/Compare Match Interrupt Enable C<br>When this bit is set to 1, an IMIC interrupt requested by the IMFC flag in TRCSR is enabled. |
| 1 | IMIEB | 0 | R/W | Input Capture/Compare Match Interrupt Enable B<br>When this bit is set to 1, an IMIB interrupt requested by the IMFB flag in TRCSR is enabled. |
| 0 | IMIEA | 0 | R/W | Input Capture/Compare Match Interrupt Enable A<br>When this bit is set to 1, an IMIA interrupt requested by the IMFA flag in TRCSR is enabled. |

### 13.3.5 Timer RC Status Register (TRCSR)

TRCSR shows the status of interrupt requests.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | OVF | 0 | R/W | Timer Overflow Flag<br>[Setting condition]<br>When TRCCNT overflows from H'FFFF to H'0000<br>[Clearing condition]<br>Read OVF when OVF = 1, then write 0 in OVF |
| 6 to 4 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 3 | IMFD | 0 | R/W | Input Capture/Compare Match Flag D<br>[Setting conditions]<br>• TRCCNT = GRD when GRD functions as an output compare register<br>• The TRCCNT value is transferred to GRD by an input capture signal when GRD functions as an input capture register<br>• TRCCNT = GRD when the PWMD bit is set to 1 or the PWM2 bit to 0 in TRCMR<br>[Clearing condition]<br>Read IMFD when IMFD = 1, then write 0 in IMFD |
| 2 | IMFC | 0 | R/W | Input Capture/Compare Match Flag C<br>[Setting conditions]<br>• TRCCNT = GRC when GRC functions as an output compare register<br>• The TRCCNT value is transferred to GRC by an input capture signal when GRC functions as an input capture register<br>• TRCCNT = GRC when the PWMC bit is set to 1 or the PWM2 bit to 0 in TRCMR<br>[Clearing condition]<br>Read IMFC when IMFC = 1, then write 0 in IMFC |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | IMFB | 0 | R/W | Input Capture/Compare Match Flag B<br>[Setting conditions]<br>• TRCCNT = GRB when GRB functions as an output compare register<br>• The TRCCNT value is transferred to GRB by an input capture signal when GRB functions as an input capture register<br>• TRCCNT = GRB when the PWMB bit is set to 1 or the PWM2 bit to 0 in TRCMR<br>[Clearing condition]<br>Read IMFB when IMFB = 1, then write 0 in IMFB |
| 0 | IMFA | 0 | R/W | Input Capture/Compare Match Flag A<br>[Setting conditions]<br>• TRCCNT = GRA when GRA functions as an output compare register<br>• The TRCCNT value is transferred to GRA by an input capture signal when GRA functions as an input capture register<br>• TRCCNT = GRA when the PWMD, PWMC, or PWMB bit is set to 1 or the PWM2 bit to 0 in TRCMR<br>[Clearing condition]<br>Read IMFA when IMFA = 1, then write 0 in IMFA |

### 13.3.6 Timer RC I/O Control Register 0 (TRCIOR0)

TRCIOR0 selects the functions of GRA and GRB, and specifies the functions of the FTIOA and FTIOB pins.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 6 | IOB2 | 0 | R/W | I/O Control B2<br>Selects the GRB function.<br>0: GRB functions as an output compare register<br>1: GRB functions as an input capture register |
| 5 | IOB1 | 0 | R/W | I/O Control B1 and B0 |
| 4 | IOB0 | 0 | R/W | When IOB2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOB pin at GRB compare match<br>10: 1 output to the FTIOB pin at GRB compare match<br>11: Output toggles to the FTIOB pin at GRB compare match<br>When IOB2 = 1,<br>00: Input capture at rising edge at the FTIOB pin<br>01: Input capture at falling edge at the FTIOB pin<br>1X: Input capture at rising and falling edges of the FTIOB pin |
| 3 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 2 | IOA2 | 0 | R/W | I/O Control A2<br>Selects the GRA function.<br>0: GRA functions as an output compare register<br>1: GRA functions as an input capture register |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | IOA1 | 0 | R/W | I/O Control A1 and A0 |
| 0 | IOA0 | 0 | R/W | When IOA2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOA pin at GRA compare match<br>10: 1 output to the FTIOA pin at GRA compare match<br>11: Output toggles to the FTIOA pin at GRA compare match<br>When IOA2 = 1,<br>00: Input capture at rising edge of the FTIOA pin<br>01: Input capture at falling edge of the FTIOA pin<br>1X: Input capture at rising and falling edges of the FTIOA pin |

[Legend]

X: Don't care.

Note: When a GR register functions as a buffer register for a paired GR register, the settings in the IOA2 and IOB2 bits in TRCIOR0 and the IOC2 and IOD2 bits in TRCIOR1 of both registers should be the same.

### 13.3.7 Timer RC I/O Control Register 1 (TRCIOR1)

TRCIOR1 selects the functions of GRC and GRD, and specifies the functions of the FTIOC and FTIOD pins.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 6 | IOD2 | 0 | R/W | I/O Control D2<br>Selects the GRD function.<br>0: GRD functions as an output compare register<br>1: GRD functions as an input capture register |
| 5 | IOD1 | 0 | R/W | I/O Control D1 and D0 |
| 4 | IOD0 | 0 | R/W | When IOD2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOD pin at GRD compare match<br>10: 1 output to the FTIOD pin at GRD compare match<br>11: Output toggles to the FTIOD pin at GRD compare match<br>When IOD2 = 1,<br>00: Input capture at rising edge at the FTIOD pin<br>01: Input capture at falling edge at the FTIOD pin<br>1X: Input capture at rising and falling edges at the FTIOD pin |
| 3 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 2 | IOC2 | 0 | R/W | I/O Control C2<br>Selects the GRC function.<br>0: GRC functions as an output compare register<br>1: GRC functions as an input capture register |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | IOC1 | 0 | R/W | I/O Control C1 and C0 |
| 0 | IOC0 | 0 | R/W | When IOC2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOC pin at GRC compare match<br>10: 1 output to the FTIOC pin at GRC compare match<br>11: Output toggles to the FTIOC pin at GRC compare match<br>When IOC2 = 1,<br>00: Input capture to GRC at rising edge of the FTIOC pin<br>01: Input capture to GRC at falling edge of the FTIOC pin<br>1X: Input capture to GRC at rising and falling edges of the FTIOC pin |

[Legend]

X: Don't care.

Note: When a GR register functions as a buffer register for a paired GR register, the settings in the IOA2 and IOB2 bits in TRCIOR0 and the IOC2 and IOD2 bits in TRCIOR1 of both registers should be the same.

### 13.3.8 Timer RC Output Enable Register (TRCOER)

TRCOER enables or disables the timer outputs. When setting the PTO bit to 1 and driving the $\overline{\text{TRCOI}}$ signal low, the ED, EC, EB and EA bits are set to 1 and timer RC outputs are disabled.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PTO | 0 | R/W | Timer Output Disabled Mode<br>0: The ED, EC, EV, EA bits are not set to 1 by the low level input of the TRCOI signal<br>1: The ED, EC, EV, EA bits are set to 1 by the low level input of the TRCOI signal |
| 6 to 4 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 3 | ED | 1 | R/W | Master Enable D<br>0: The FTIOD output is enabled according to TRCMR, TRCIOR0, and TRCIOR1<br>1: The FTIOD output is disabled regardless of TRCMR, TRCIOR0, and TRCIOR1 (The FTIOD pin functions as an I/O port) |
| 2 | EC | 1 | R/W | Master Enable C<br>0: The FTIOC output is enabled according to TRCMR, TRCIOR0, and TRCIOR1<br>1: The FTIOC output is disabled regardless of TRCMR, TRCIOR0, and TRCIOR1 (The FTIOC pin functions as an I/O port) |
| 1 | EB | 1 | R/W | Master Enable B<br>0: The FTIOB output is enabled according to TRCMR, TRCIOR0, and TRCIOR1<br>1: The FTIOB output is disabled regardless of TRCMR, TRCIOR0, and TRCIOR1 (The FTIOB pin functions as an I/O port) |
| 0 | EA | 1 | R/W | Master Enable A<br>0: The FTIOA output is enabled according to TRCIOR0 and TRCIOR1<br>1: The FTIOA output is disabled regardless of TRCIOR0 and TRCIOR1 (The FTIOA pin functions as an I/O port) |

### 13.3.9 Timer RC Digital Filtering Function Select Register (TRCDF)

TRCDF enables or disables the digital filter for each of the FTIOA to FTIOD and TRGC pin. The setting in this register is valid on the corresponding pin when the FTIOA to FTIOA inputs are enabled by TRCIOR0 and TRCIOR1 and the TRGC input is selected by bits TCEG1 and TCEG0 in TRCCR2.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | DFCK1 | 0 | R/W | These bits select the clock to be used by the digital filter.<br>00: $\phi$/32<br>01: $\phi$/8<br>10: $\phi$<br>11: Clock specified by bits CKS2 to CKS0 in TRCCR1 |
| 6 | DFCK0 | 0 | R/W | |
| 5 | — | 0 | — | Reserved<br>This bit is always read as 0. |
| 4 | DFRG | 0 | R/W | Enables or disables the digital filter for the TRGC pin.<br>0: Disables the digital filter<br>1: Enables the digital filter |
| 3 | DFD | 0 | R/W | Enables or disables the digital filter for the FTIOD pin.<br>0: Disables the digital filter<br>1: Enables the digital filter |
| 2 | DFC | 0 | R/W | Enables or disables the digital filter for the FTIOC pin.<br>0: Disables the digital filter<br>1: Enables the digital filter |
| 1 | DFB | 0 | R/W | Enables or disables the digital filter for the FTIOB pin.<br>0: Disables the digital filter<br>1: Enables the digital filter |
| 0 | DFA | 0 | R/W | Enables or disables the digital filter for the FTIOA pin.<br>0: Disables the digital filter<br>1: Enables the digital filter |

### 13.3.10 Timer RC Counter (TRCCNT)

TRCCNT is a 16-bit readable/writable up-counter. The input clock is selected by bits CKS2 to CKS0 in TRCCR1. TRCCNT can be cleared to H'0000 through a compare match of GRA by setting the CCLR bit in TRCCR1 to 1. When TRCCNT overflows (changes from H'FFFF to H'0000), the OVF flag in TRCSR is set to 1. If the OVIE bit in TRCIER is set to 1 at this time, an interrupt request is generated. TRCCNT must always be read from or written to in units of 16 bits; 8-bit accesses are not allowed. TRCCNT is initialized to H'0000 by a reset.

### 13.3.11 General Registers A, B, C, and D (GRA, GRB, GRC, and GRD)

Each general register is a 16-bit readable/writable register that can function as either an output-compare register or an input-capture register. The function is selected by settings in TRCIOR0 and TRCIOR1.

When a general register is used as an input-compare register, its value is constantly compared with the TRCCNT value. When the two values match (a compare match), the corresponding flag (the IMFA, IMFB, IMFC, or IMFD bit) in TRCSR is set to 1. An interrupt request is generated at this time, when the IMIEA, IMIEB, IMIEC, or IMIED bit in TRCIER is set to 1. A compare match output can be selected in TRCIOR.

When a general register is used as an input-capture register, an external input-capture signal is detected and the current TRCCNT value is stored in the general register. The corresponding flag (the IMFA, IMFB, IMFC, or IMFD bit) in TRCSR is set to 1. If the corresponding interrupt-enable bit (the IMIEA, IMIEB, IMIEC, or IMIED bit) in TRIER is set to 1 at this time, an interrupt request is generated. The edge of the input-capture signal is selected in TRCIOR.

GRC and GRD can be used as buffer registers of GRA and GRB, respectively, by setting BUFEA and BUFEB in TRCMR.

For example, when GRA is set as an output-compare register and GRC is set as the buffer register for GRA, the value in the buffer register GRC is sent to GRA whenever compare match A is generated.

When GRA is set as an input-capture register and GRC is set as the buffer register for GRA, the value in TRCCNT is transferred to GRA and the value in the buffer register GRC is transferred to GRA whenever an input capture is generated.

GRA to GRD must be written or read in 16-bit units; 8-bit access is not allowed. GRA to GRD are initialized to H'FFFF by a reset.

## 13.4 Operation

Timer RC has the following operating modes.

- Timer mode operation
  - Enables output compare and input capture functions by setting the IOA2 to IOA0 and IOB2 to IOB0 bits in TRCIOR0 and the IOC2 to IOC0 and IOD2 to IOD0 bits in TRCIOR1
- PWM mode operation
  - Enables PWM mode operation by setting the PWMD, PWMC, and PWMB bits in TRCMR
- PWM2 mode operation
  - Enables PWM2 mode operation by setting the PWM2 bit in TRMR

The FTIOA to FTIOD pins indicate the timer output mode by each register setting.

- FTIOA pin

| **Register Name** | **TRCOER** | **TRCIOR0** | | |
|---|---|---|---|---|
| **Bit Name** | **EA** | **PWM2** | **IOA2 to IOA0** | **Function** |
| Setting values | 0 | 1 | 001, 01X | Timer mode waveform output (output compare function) |
| | 0 | 1 | 1XX | Timer mode (input capture function) |
| | 1 | | | |
| | Other than above | | | General I/O port |

[Legend]

X: Don't care.

- FTIOB pin

| Register Name | TRCOER | TRCMR | | TRCIOR0 | |
|---|---|---|---|---|---|
| **Bit Name** | **EB** | **PWM2** | **PWMB** | **IOB2 to IOB0** | **Function** |
| Setting values | 0 | 0 | X | XXX | PWM2 mode waveform output |
| | 0 | 1 | 1 | XXX | PWM mode waveform output |
| | 0 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | 0 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | 1 | | | | |
| | Other than above | | | | General I/O port |

[Legend]

X: Don't care.

- FTIOC pin

| Register Name | TRCOER | TRCMR | | TRCIOR1 | |
|---|---|---|---|---|---|
| **Bit Name** | **EC** | **PWM2** | **PWMC** | **IOC2 to IOC0** | **Function** |
| Setting values | 0 | 1 | 1 | XXX | PWM mode waveform output |
| | 0 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | 0 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | 1 | | | | |
| | Other than above | | | | General I/O port |

[Legend]

X: Don't care.

- FTIOD pin

| Register Name | TRCOER | TRCMR | | TRCIOR1 | |
|---|---|---|---|---|---|
| Bit Name | ED | PWM2 | PWMD | IOD2 to IOD0 | Function |
| Setting values | 0 | 1 | 1 | XXX | PWM mode waveform output |
| | 0 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | 0 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | 1 | | | | |
| | Other than above | | | | General I/O port |

[Legend]

X: Don't care.

### 13.4.1 Timer Mode Operation

TRCCNT performs free-running or periodic counting operations. After a reset, TRCCNT is set as a free-running counter. When the CTS bit in TRCMR is set to 1, TRCCNT starts counting. When the TRCCNT value overflows from H'FFFF to H'0000, the OVF flag in TRCSR is set to 1. If the OVIE in TRCIER is set to 1, an interrupt request is generated. Figure 13.2 shows an example of free-running counting.

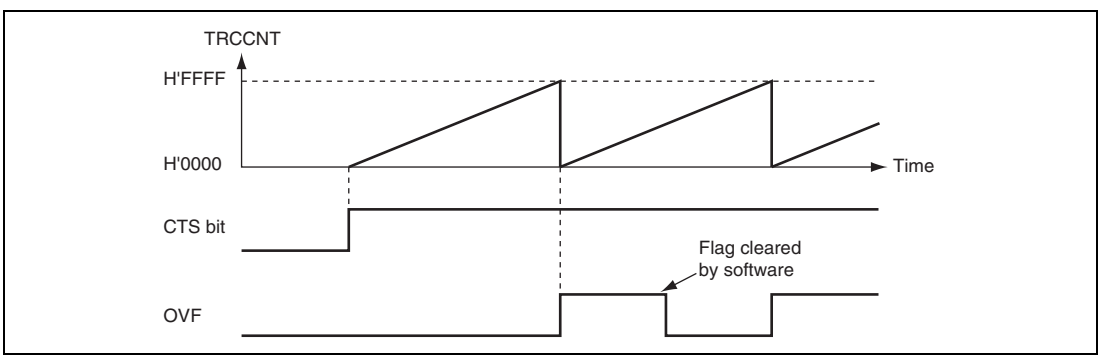



**Figure 13.2 Free-Running Counter Operation**

Periodic counting operation can be performed when GRA is set as an output compare register and the CCLR bit in TRCCR1 is set to 1. When the counter value matches GRA, TRCCNT is cleared to H'0000, the IMFA flag in TRCSR is set to 1. If the corresponding IMIEA bit in TRCIER is set to 1, an interrupt request is generated. TRCCNT continues counting from H'0000. Figure 13.3 shows an example of periodic counting.

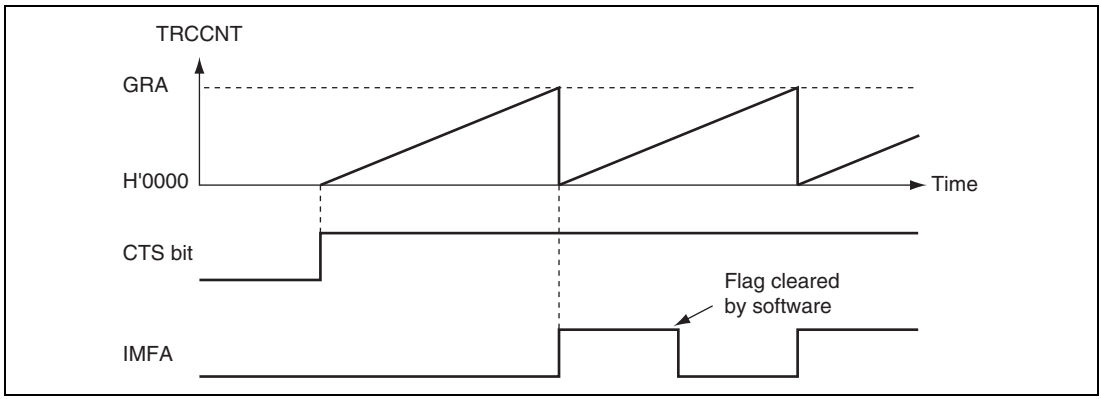



**Figure 13.3 Periodic Counter Operation**

By setting a general register as an output compare register, the specified level of a signal can be output on the FTIOA, FTIOB, FTIOC, or FTIOD pin on compare match A, B, C, or D. The output level can be selected from 0, 1, or toggle. Figure 13.4 shows an example of TRCCNT functioning as a free-running counter. In this example, 1 is output on compare match A and 0 is output on compare match B. When the signal level is already at the selected output level, it is not changed on a compare match.

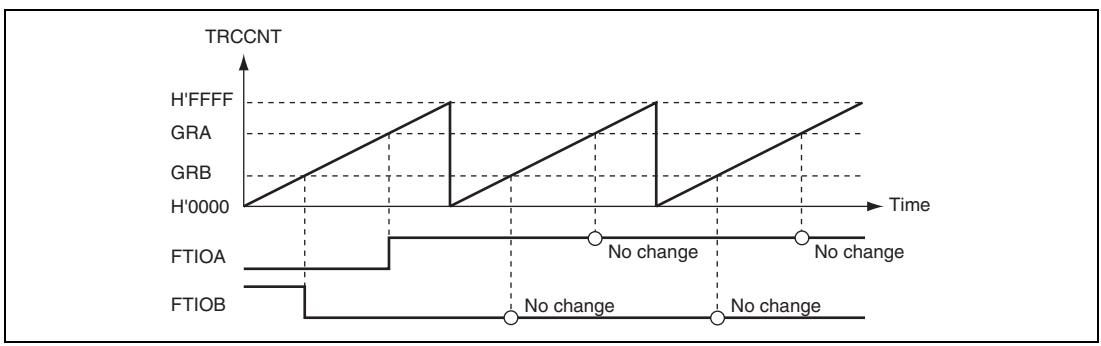



**Figure 13.4 0 and 1 Output Example (TOA = 0, TOB = 1)**

Figure 13.5 shows an example of toggled output when TRCCNT functions as a free-running counter, and the toggled output is selected for both compare matches A and B.

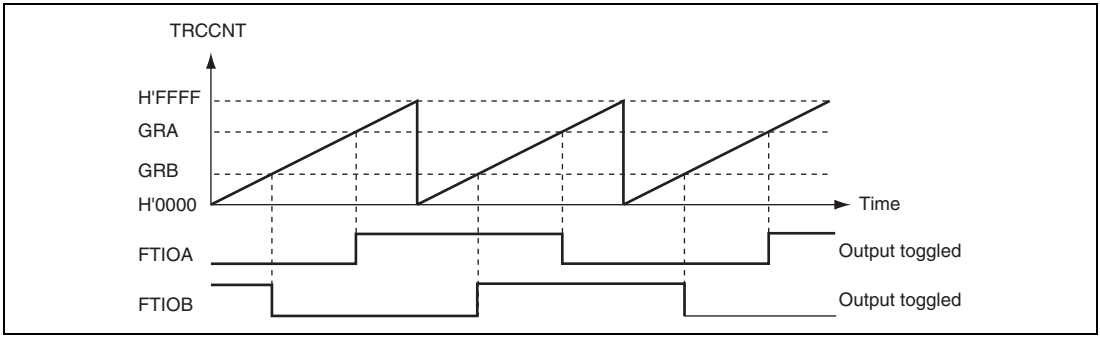



**Figure 13.5 Toggle Output Example (TOA = 0, TOB = 1)**

Figure 13.6 shows another example of toggled output when TRCCNT functions as a periodic counter on both compare matches A and B.

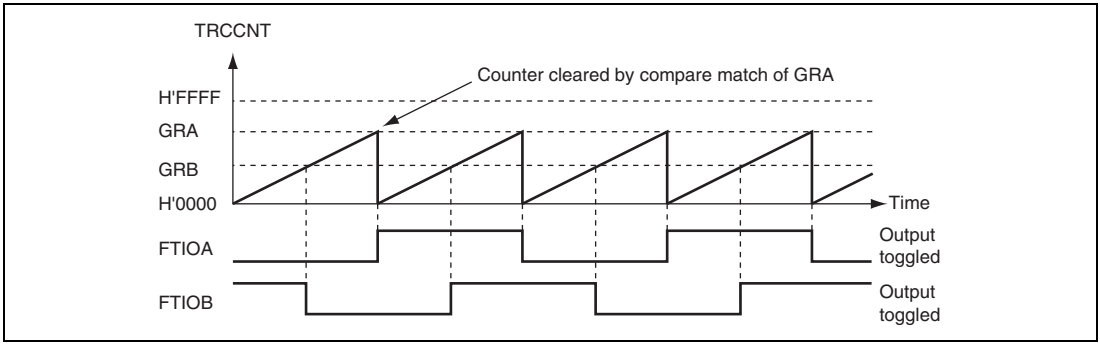



**Figure 13.6 Toggle Output Example (TOA = 0, TOB = 1)**

The TRCCNT value can be captured into a general register (GRA, GRB, GRC, or GRD) when signal levels are changed on an input-capture pin (FTIOA, FTIOB, FTIOC, or FTIOD) by specifying the general register as an input capture register. The capture timing can be selected from the rising, falling, or both edges. By using the input-capture function, the width or cycle of a pulse can be measured. Figure 13.7 shows an example of an input capture when both edges of the FTIOA signal and the falling edge of the FTIOB signal are selected as capture timings. TRCCNT functions as a free-running counter.

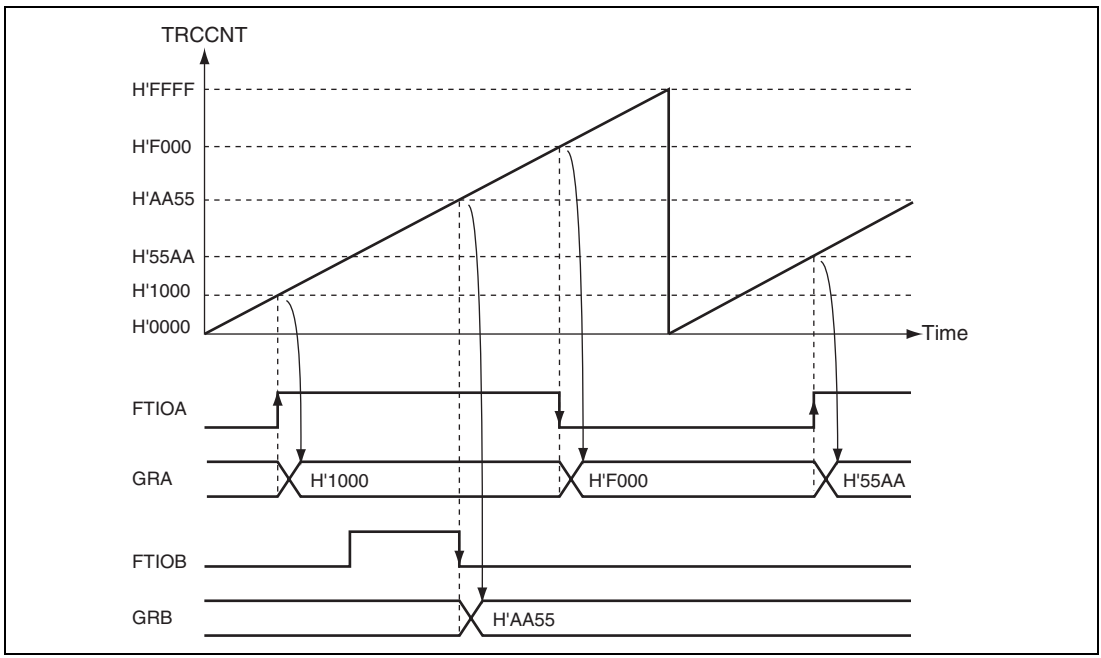



**Figure 13.7 Input Capture Operating Example**

Figure 13.8 shows an example of buffer operation when GRA is set as an input-capture register and GRC is set as the buffer register for GRA. TRCCNT functions as a free-running counter and is captured at both rising and falling edges of the FTIOA signal. Due to the buffer operation, the GRA value is transferred to GRC on an input-capture A and the TRCCNT value is stored in GRA.

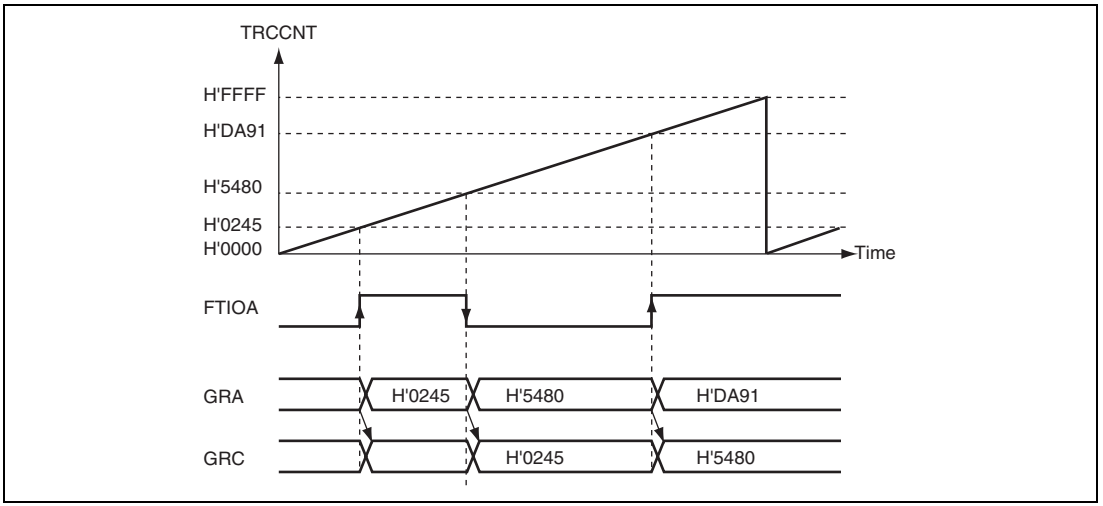



**Figure 13.8 Buffer Operation Example (Input Capture)**

### 13.4.2 PWM Mode Operation

In PWM mode, PWM waveforms are generated by using GRA as the cycle register and GRB, GRC, and GRD as duty cycle registers. PWM waveforms are output from the FTIOB, FTIOC, and FTIOD pins. Up to three-phase PWM waveforms can be output. In PWM mode, a general register functions as an output compare register automatically. The output level of each pin depends on the corresponding timer output level set bit (TOB, TOC, or TOD) in TRCCR1. When the TOB bit is set to 1, the FTIOB output goes 1 on compare match A and 0 on compare match B. When the TOB bit is cleared to 0, the FTIOB output goes 0 on compare match A and 1 on compare match B. When an output pin is set to PWM mode, the settings in TRCIOR0 and TRCIOR1 are ignored. If the same value is set in the cycle register and duty cycle register, output levels are not changed when a compare match occurs.

Figure 13.9 shows an example of operation in PWM mode. The output signals go 1 (TOB = TOC = TOD = 1) and TRCCNT is cleared on compare match A, and the output signals go 0 on compare match B, C, and D .

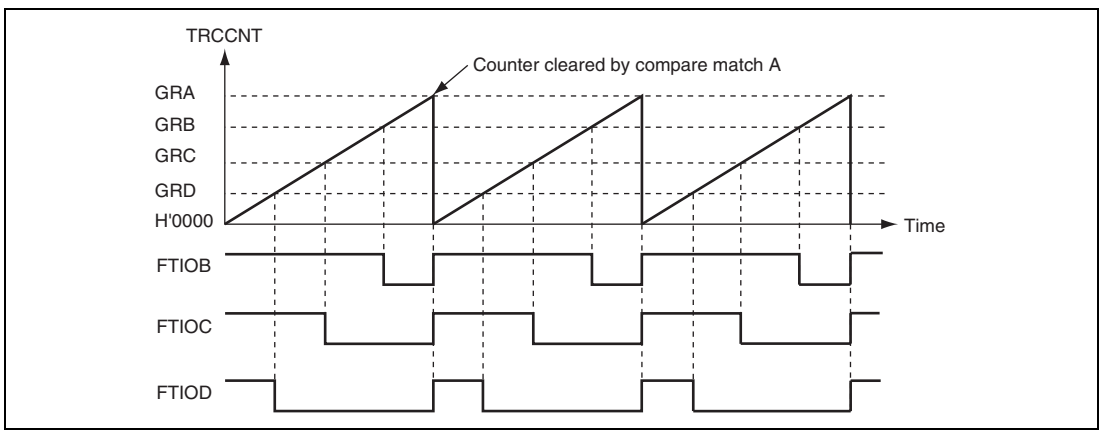



**Figure 13.9 PWM Mode Example (1)**

Figure 13.10 shows another example of operation in PWM mode. The output signals go 0 (TOB = TOC = TOD = 0) and TRCCNT is cleared on compare match A, and the output signals go 1 on compare match B, C, and D .

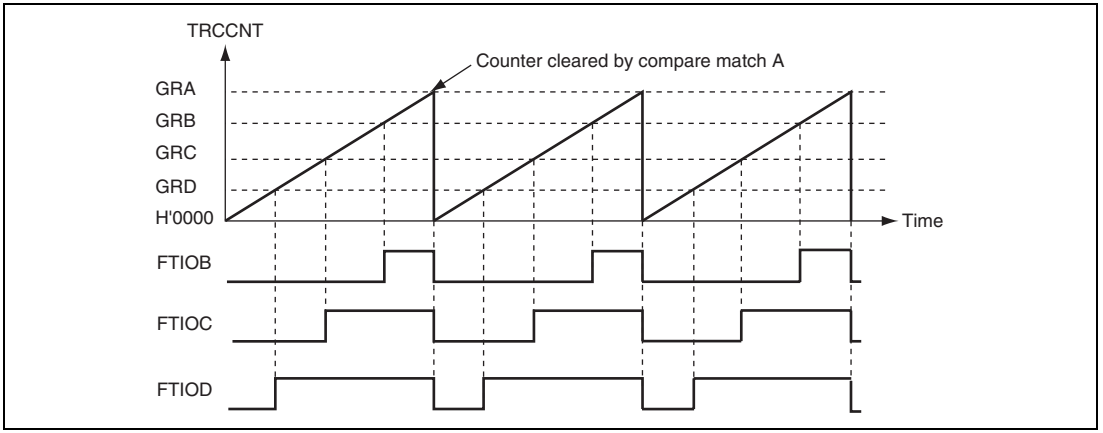



**Figure 13.10 PWM Mode Example (2)**

Figure 13.11 shows an example of buffer operation when the FTIOB pin is set to PWM mode and GRD is set as the buffer register for GRB. TRCCNT is cleared on compare match A, and the FTIOB pin outputs 1 on compare match B and 0 on compare match A.

Due to the buffer operation, the FTIOB output levels are changed and the value of buffer register GRD is transferred to GRB whenever compare match B occurs. This procedure is repeated every time compare match B occurs.

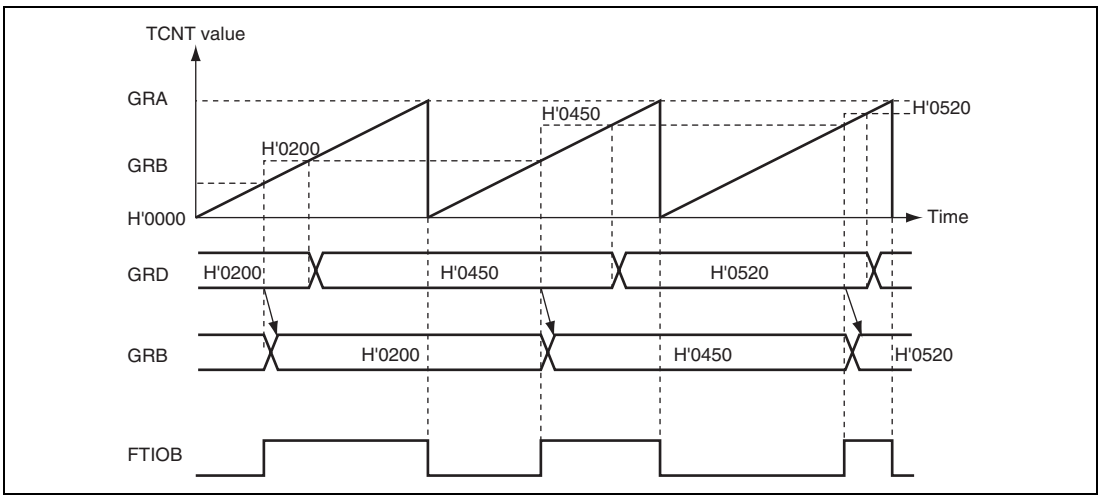



**Figure 13.11 Buffer Operation Example (Output Compare)**

Figures 13.12 and 13.13 show examples of the output of PWM waveforms with duty cycles of 0% and 100%.

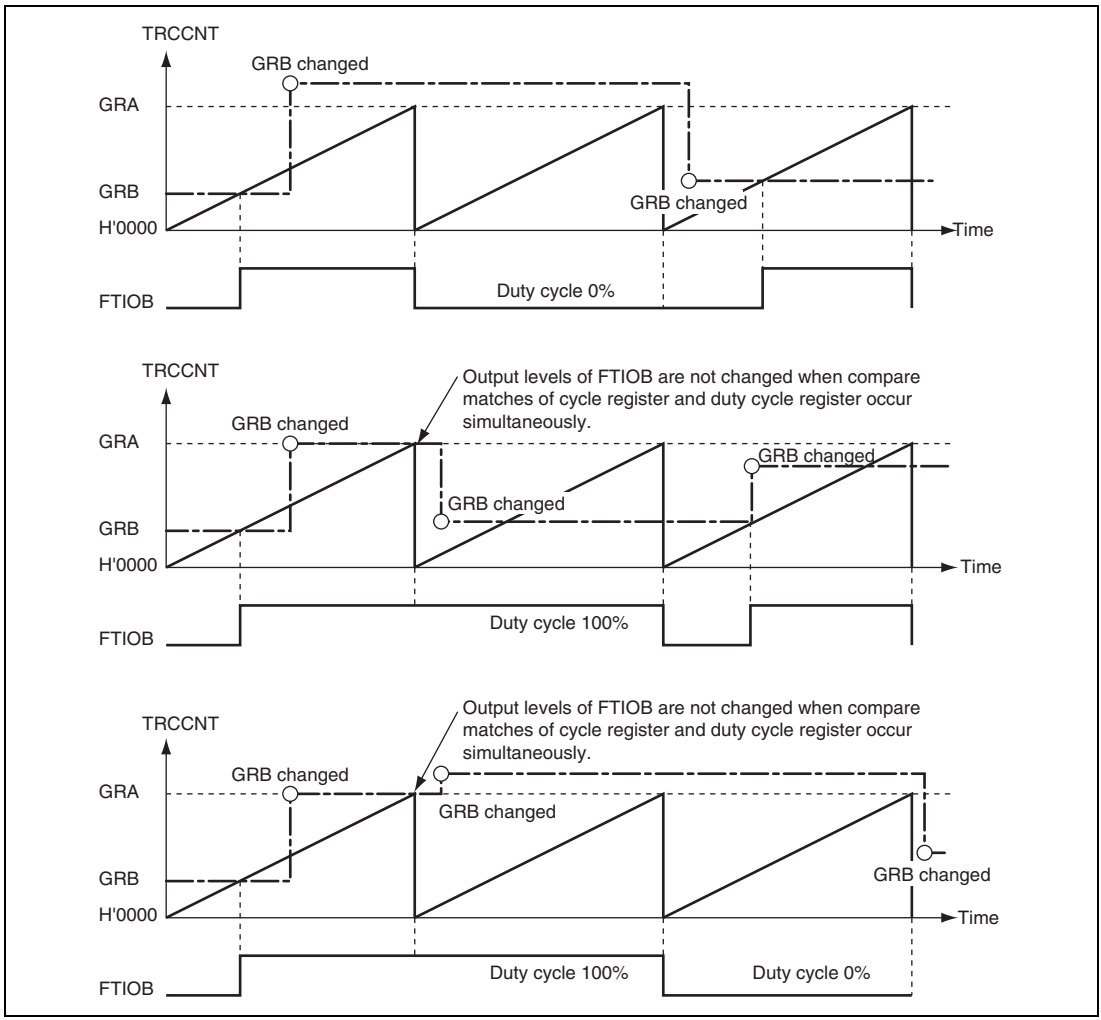



**Figure 13.12 PWM Mode Example**
**(TOB, TOC, and TOD = 0: Initial Output Set to 0)**

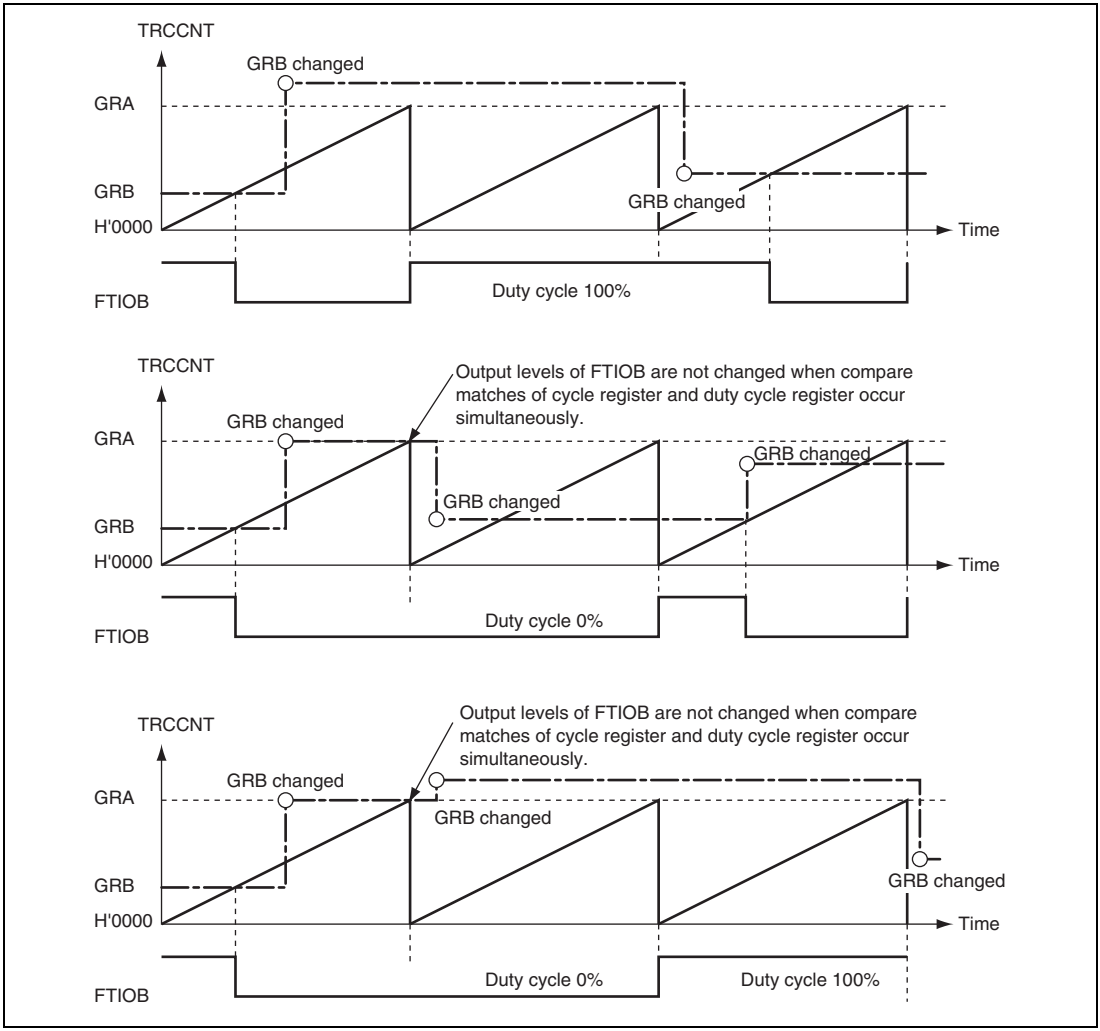



**Figure 13.13 PWM Mode Example**
**(TOB, TOC, and TOD = 1: Initial Output Set to 1)**

### 13.4.3 PWM2 Mode Operation

In PWM2 mode, waveforms are output on the FTIOB pin when a compare match occurs on GRB or GRC. GRD functions as a buffer register for GRB by setting the BUFEB bit in TRCMR to 1. The output level of the FTIOB signal is specified by the TOB bit in TRCCR1. When TOB = 0, 1 is output on a compare match of GRC and 0 is output on a compare match of GRB. When TOB = 1, 0 is output on a compare match of GRC and 1 is output on a compare match of GRB.

Table 13.3 shows the correspondence between the pin configuration and GR registers and figure 13.14 is a block diagram of PWM2 mode.

Figures 13.15 and 13.16 show the GRD and GRB buffer operating timing in PWM2 mode.

In PWM2 mode, the value of GRD is transferred to GRB on a compare match of GRA and the counter is cleared. Note, however, that the counter is only cleared when the CCLR bit in TRCCR1 is set to 1. Moreover, when the trigger input is enabled by the TCEG1 and TCEG0 bits in TRCCR2, the value of GRD is transferred to GRB by the trigger signal and the counter is cleared. The input/output pins of timers which do not operate in PWM2 mode are only used as general I/O ports.

**Table 13.3 Pin Configuration in PWM2 Mode and GR Registers**

| Pin Name | Input/Output | Compare Match Register | Buffer Register |
|---|---|---|---|
| FTIOA | I/O | Port/TRGC | Port/TRGC |
| FTIOB | Output | GRB | GRD |
| | | GRC | — |
| FTIOC | I/O | Port | Port |
| FTIOD | I/O | Port | Port |

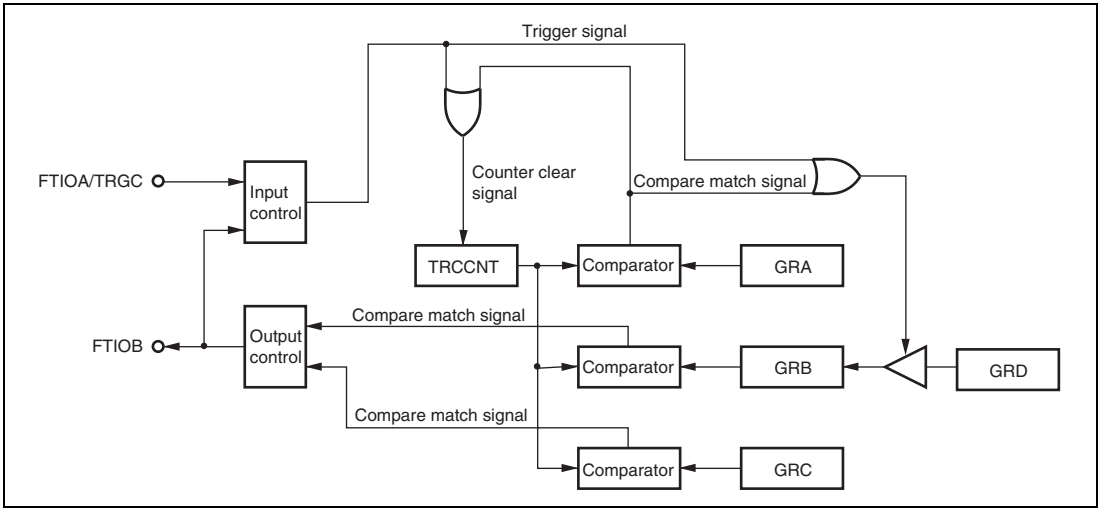



**Figure 13.14 Block Diagram in PWM2 Mode**

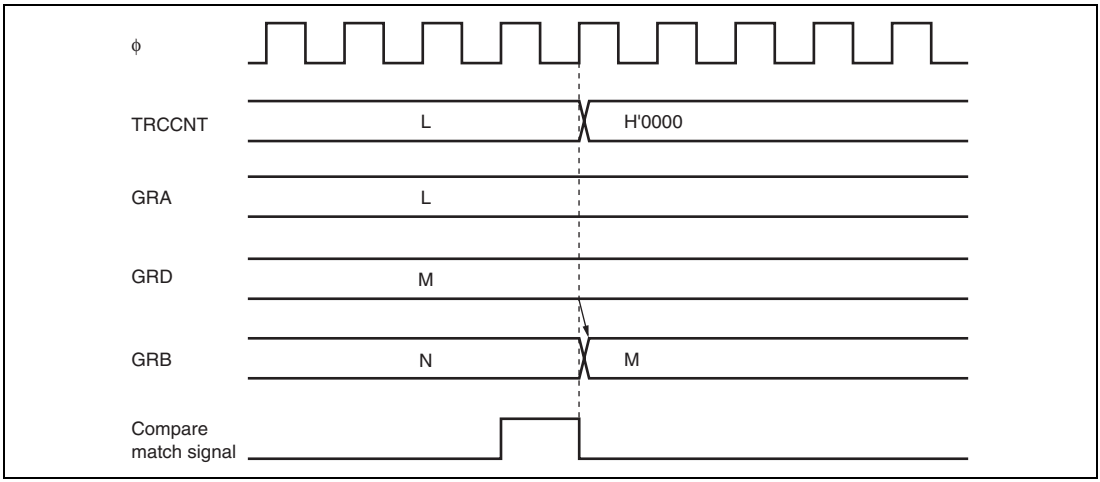



**Figure 13.15 GRD and GRB Buffer Operating Timing in PWM2 Mode (1)**

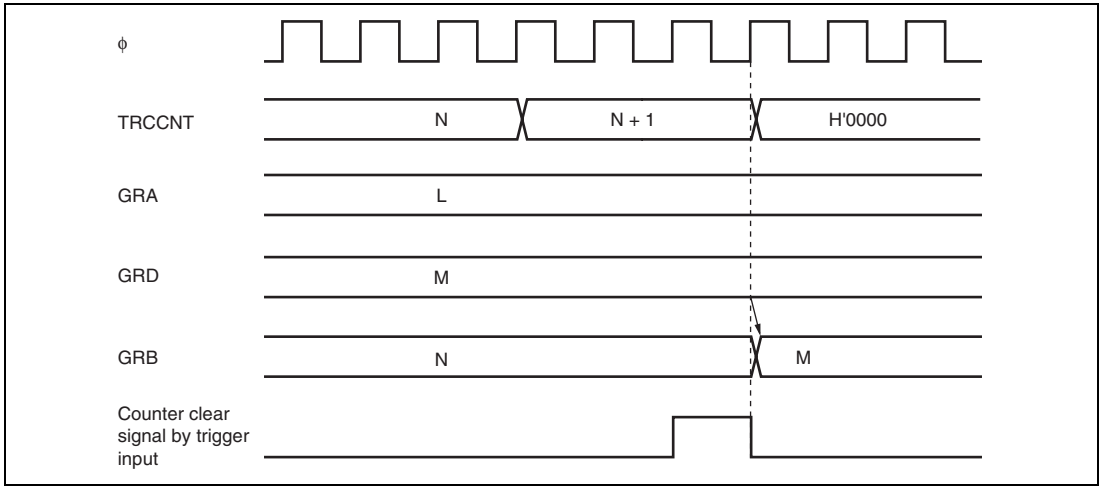



**Figure 13.16 GRD and GRB Buffer Operating Timing in PWM2 Mode (2)**

In PWM2 mode, a pulse with a specified pulse width can be output on the FTIOB pin when a specified delay time has elapsed since the TRGC signal was asserted. An assertion of the TRGC signal starts counting up. Arbitrary values can be specified for the pulse width and delay time.

Figures 13.17 and 13.18 show these examples in PWM2 mode. In these examples, the falling edge of the TRGC input is selected by TRCCR2 (setting the TCEG1 bit to 1 and clearing the TCEG0 bit to 0), TRCCNT continues counting-up on compare match A of GRA (clearing the CSTP bit in TRCCR2 to 0), and GRD is set as the buffer register (setting the BUFEB bit in TRCMR to 1). The initial value of the output signal is set to either 0 or 1 by TRCCR1 (clearing the TOB bit to 0 or setting the TOB bit to 1), TRCCNT is cleared on compare match A (setting the CCLR bit in TRCCR1 to 1), and the waveform is output from the FTIOB pin (clearing the PWM2 bit in TRCMR to 0).

When the TOB bit in TRCCR1 is cleared to 0 with the PWM2 mode function, the input edge is ignored while the FTIOB pin is driven high. Whereas, when the TOB bit is set to 1, the input edge is ignored while the FTIOB pin is driven low. The transfer from GRD to GRB is carried out on a compare match of GRA and the TRGC input. However, if the TRGC input is canceled due to the change of the FTIOB level, the transfer from GRD to GRB is not carried out.

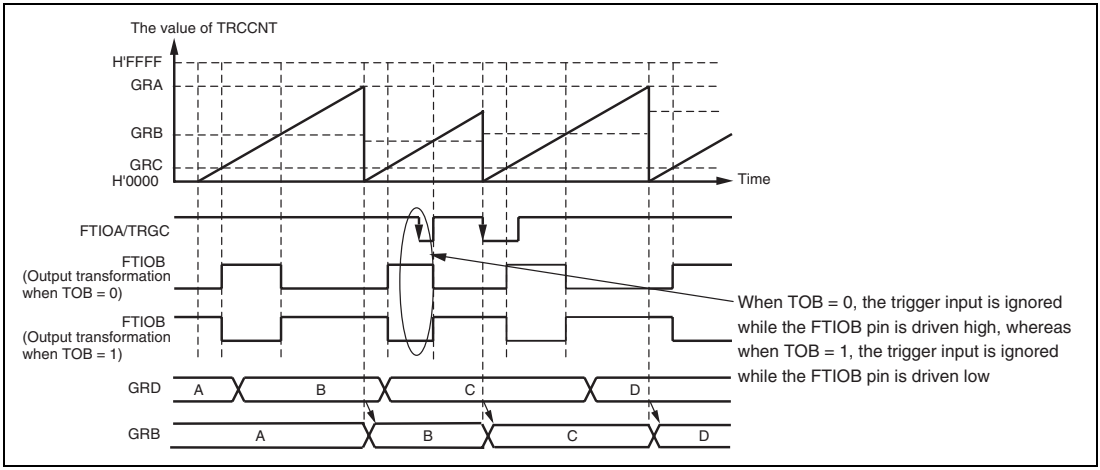



**Figure 13.17 Example (1) of TRGC Synchronous Operation in PWM2 Mode**

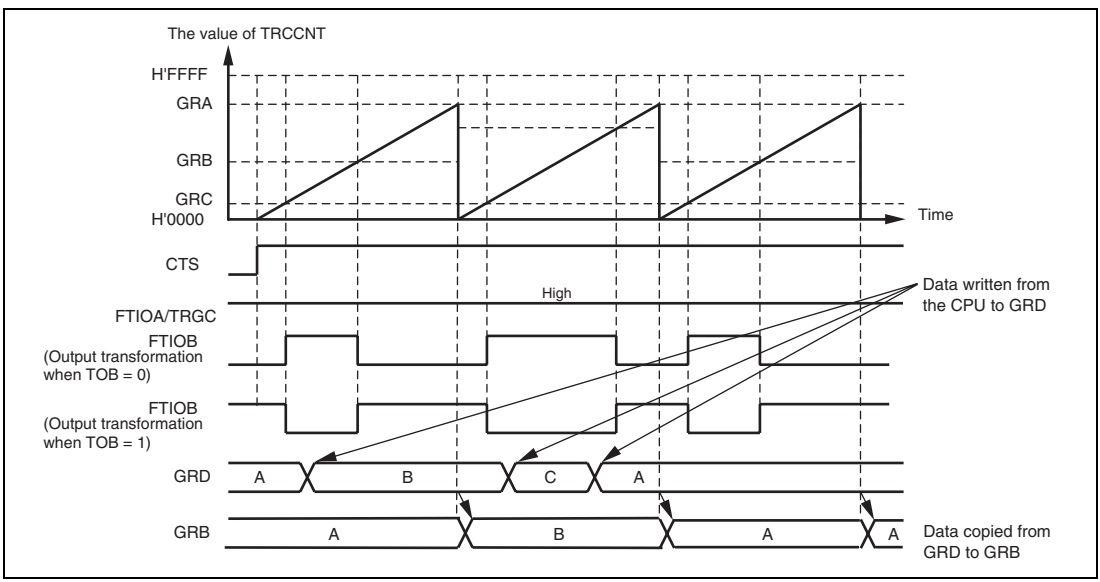



**Figure 13.18 Example (2) of TRGC Synchronous Operation in PWM2 Mode**

The following is an example of stopping operation of the counter in PWM2 mode. When the CSTP bit in TRCCR2 is set to 1 and the CCLR bit in TRCCR1 is set to 1, TRCCNT is cleared to H'0000 on a compare match of GRA and stops counting. Moreover, TRCCNT is forcibly stopped counting and cleared to the initial value when the CTS bit in TRCMR is cleared to 0. Figure 13.19 shows such an example when the TOB bit in TRCCR1 is cleared to 0 and set to 1.

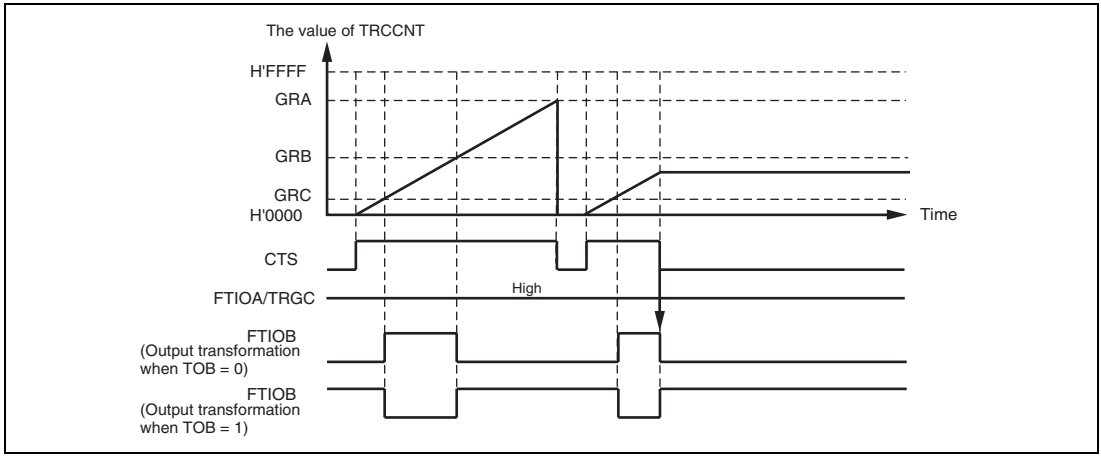



**Figure 13.19 Example of Stopping Operation of the Counter in PWM2 Mode**

The following is an example of output operation of the one-shot pulse waveform in PWM2 mode. When the TRGC input is disabled by TRCCR2 (clearing the TCEG1 and TCEG0 bits to 0), TRCCNT is set to counting-up on compare match A of GRA (setting the CSTP bit in TRCCR2 to 1), TRCCNT is cleared on compare match A (setting the CCLR bit in TRCCR1 to 1), and the initial value of the output signal is set to 0 by TRCCR1 (clearing the TOB bit to 0), TRCCNT starts counting when the CTS bit in TRCMR is set to 1. Then, TRCCNT is cleared to H'0000 on a compare match of GRA and stops counting, and the one-shot pulse waveform is output. Figure 13.20 shows such an example.

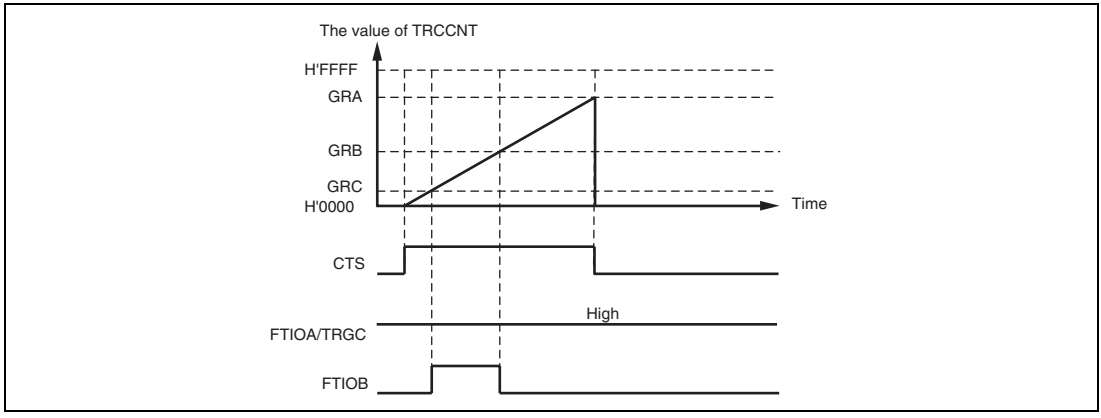



**Figure 13.20 Example (1) of Output Operation of One-Shot Pulse Waveform in PWM2 Mode**

The following is an example of operation when TRCCNT starts counting by the TRGC input and the one-shot pulse waveform is output in PWM2 mode. When the falling edge of the TRGC input is selected by TRCCR2 (setting the TCEG1 bit to 1 and clearing the TCEG0 bit to 0), TRCCNT is set to counting-up on compare match A of GRA (setting the CSTP bit in TRCCR2 to 1), TRCCNT is cleared on compare match A (setting the CCLR bit in TRCCR1 to 1), and the initial value of the output signal is set to 0 by TRCCR1 (clearing the TOB bit to 0), TRCCNT starts counting at the falling edge of FTIOA/TRGC after the CTS bit in TRCMR has been set to 1. Then, TRCCNT is cleared to H'0000 on a compare match of GRA and stops counting, and the one-shot pulse waveform is output. Figure 13.21 shows such an example.

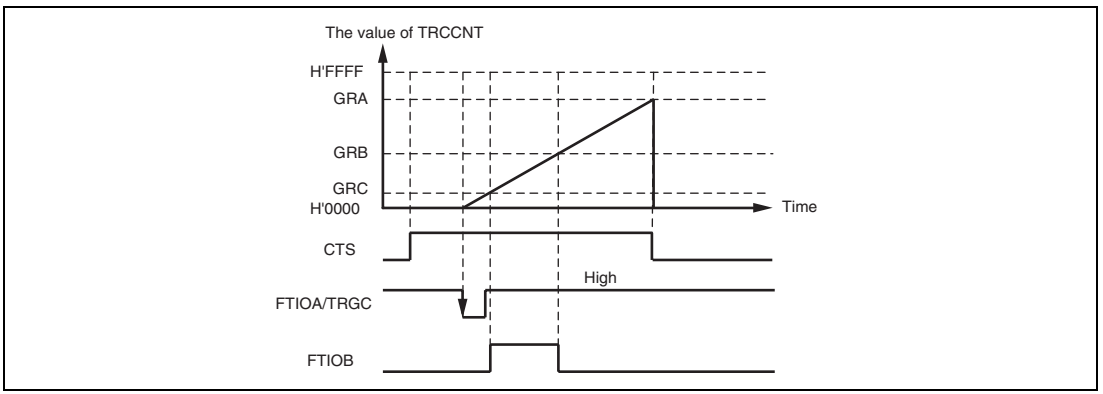



**Figure 13.21 Example (2) of Output Operation of One-Shot Pulse Waveform in PWM2 Mode**

### 13.4.4 Digital Filtering Function for Input Capture Inputs

Input signals on the FTIOA to FIOD and TRGC pin can be input via the digital filters. The digital filter includes three latches connected in series and a matching detecting circuit. The latches operate on the sampling clock specified by bits DFCK1 and DFCK0 in TRCDF and stores an input signal on the FTIOA to FTIOD pins or TRGC pin. When outputs of the three latches match, the matching detecting circuit outputs the signal level of the input. Otherwise, the output remains unchanged. That is, when a pulse width is equal to or greater than three sampling clock cycles, the pulse is input as a signal. When a pulse width is less than three sampling clock cycles, the pulse is considered as a noise to be removed.

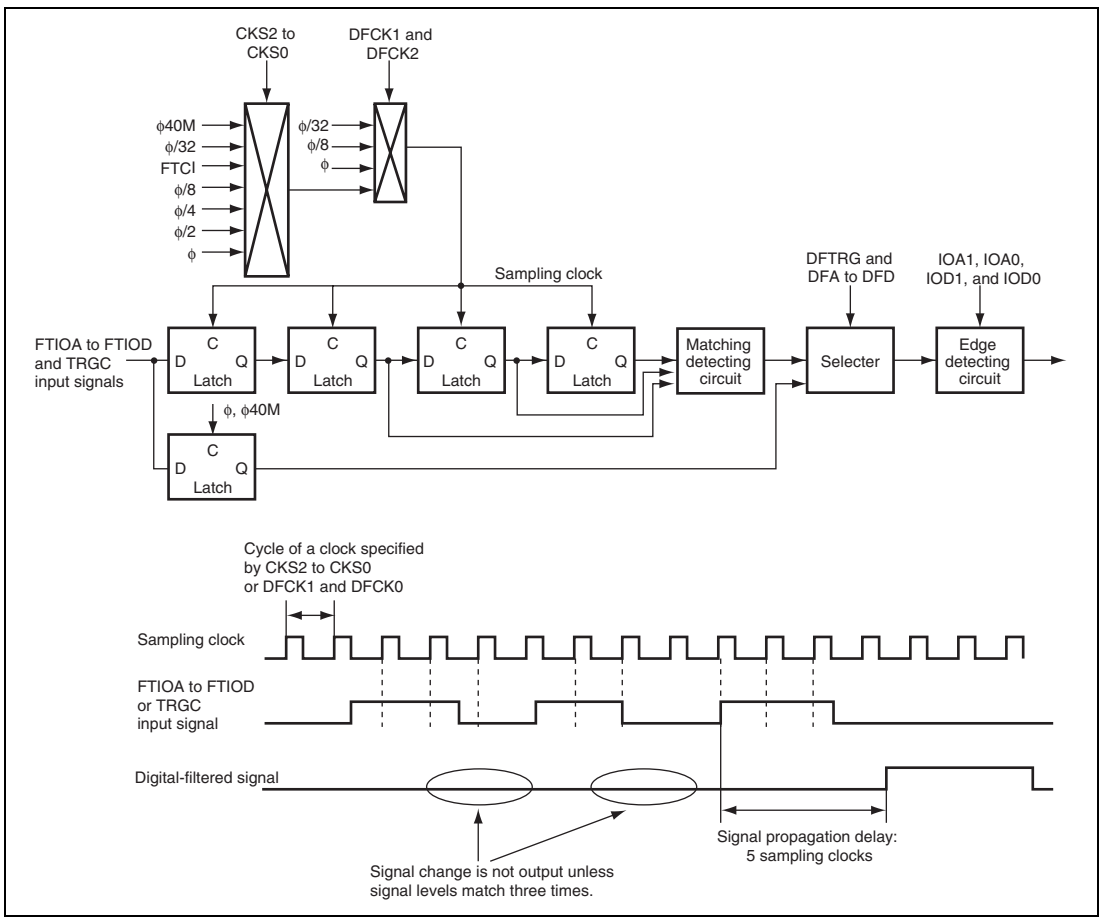



**Figure 13.22 Block Diagram of Digital Filter**

## 13.5 Operation Timing

### 13.5.1 TRCCNT Counting Timing

Figure 13.23 shows the TRCCNT count timing when the internal clock source is selected. Figure 13.24 shows the timing when the external clock source is selected.

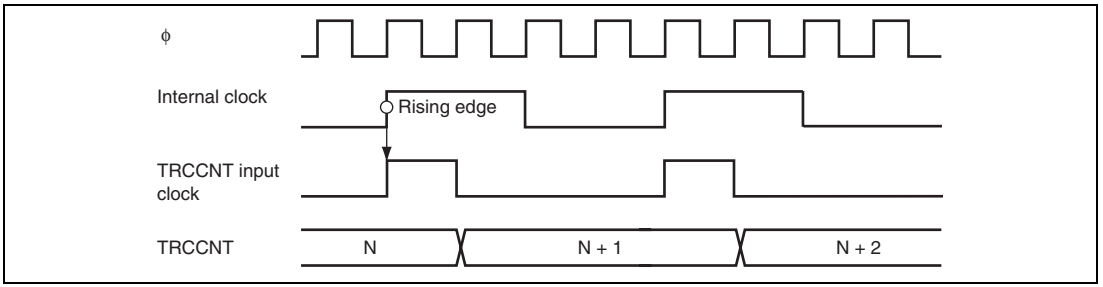



**Figure 13.23 Count Timing for Internal Clock Source**

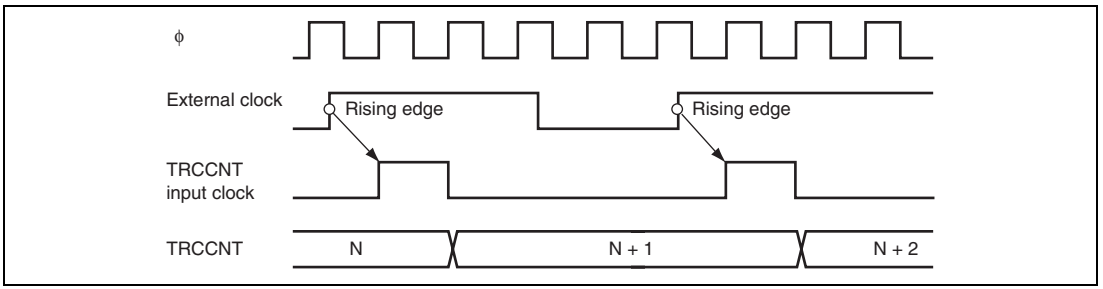



**Figure 13.24 Count Timing for External Clock Source**

### 13.5.2 Output Compare Output Timing

The compare match signal is generated in the last state in which TRCCNT and GR match (when TRCCNT changes from the matching value to the next value). When the compare match signal is generated, the output value selected in TRCIOR is output on the compare match output pin (FTIOA, FTIOB, FTIOC, or FTIOD).

When TRCCNT matches GR, the compare match signal is generated only after the next counter clock pulse is input.

Figure 13.25 shows the output compare timing.

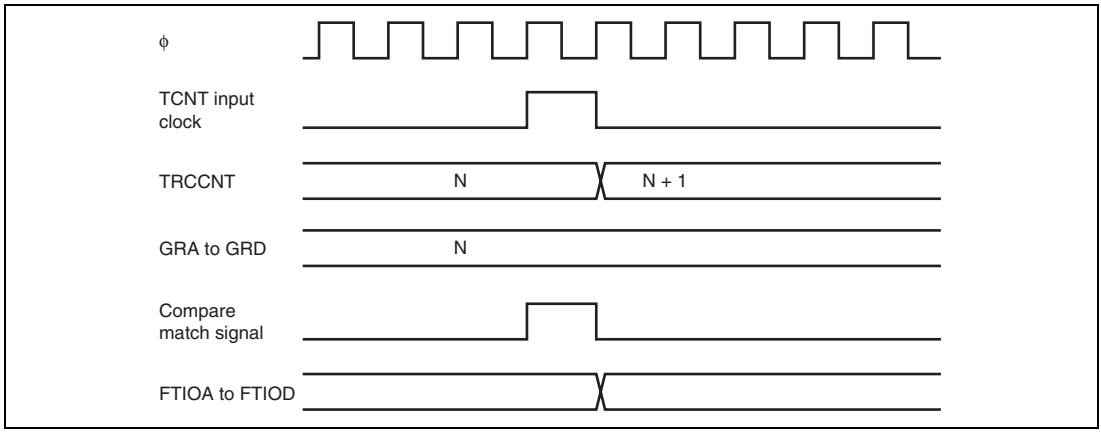



**Figure 13.25 Output Compare Output Timing**

### 13.5.3 Input Capture Timing

Input capture on the rising edge, falling edge, or both edges can be selected through settings in TRCIOR0 and TRCIOR1. Figure 13.26 shows the timing when the falling edge is selected.

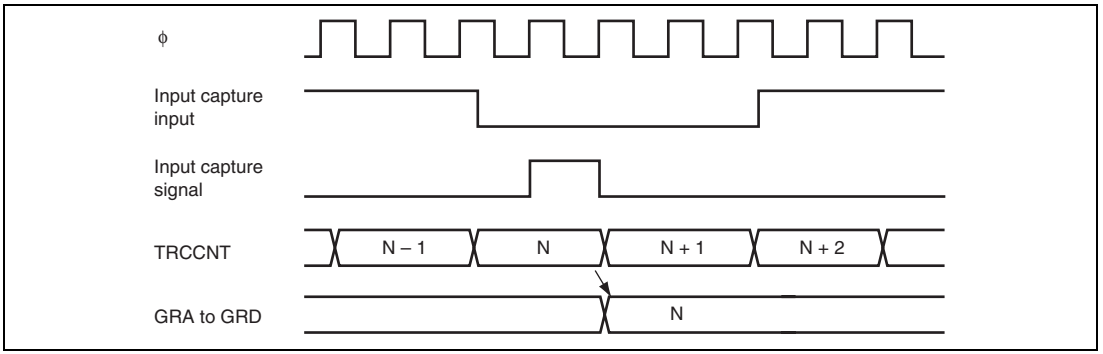



**Figure 13.26 Input Capture Input Signal Timing**

### 13.5.4 Timing of Counter Clearing by Compare Match

Figure 13.27 shows the timing when the counter is cleared by compare match A. When the GRA value is N, the counter counts from 0 to N, and its cycle is N + 1.

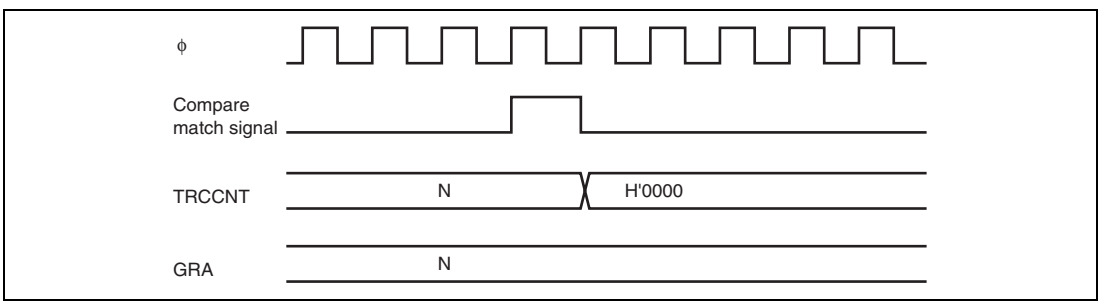



**Figure 13.27 Timing of Counter Clearing by Compare Match**

### 13.5.5 Buffer Operation Timing

Figures 13.28 and 13.29 show the buffer operation timing.

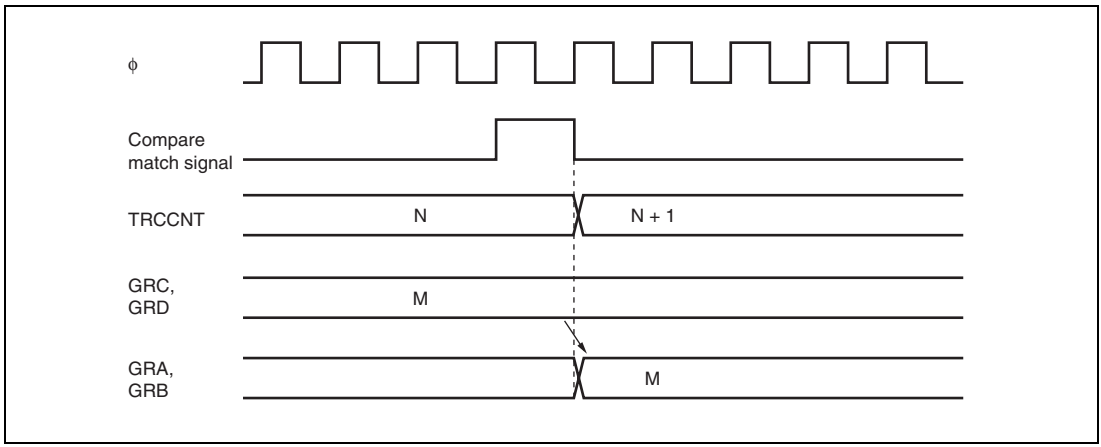



**Figure 13.28 Buffer Operation Timing (Compare Match)**

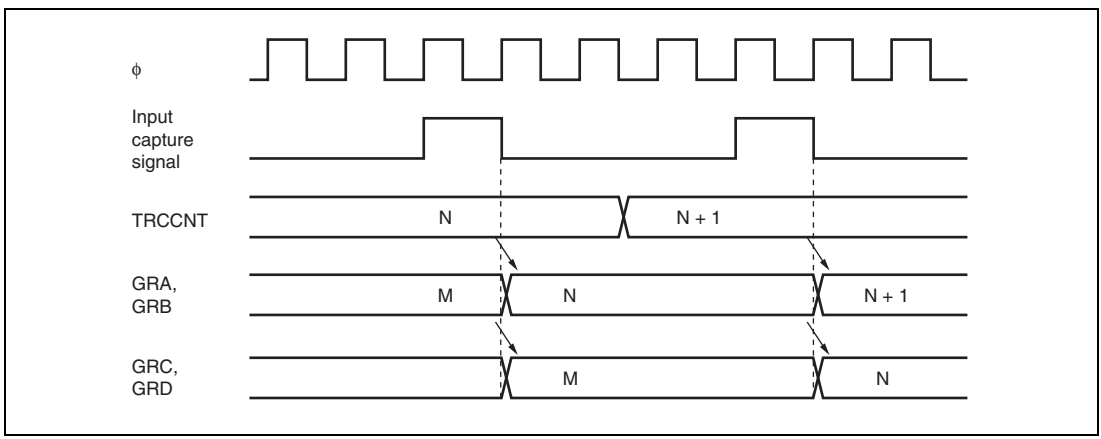



**Figure 13.29 Buffer Operation Timing (Input Capture)**

### 13.5.6 Timing of IMFA to IMFD Flag Setting at Compare Match

If a general register (GRA, GRB, GRC, or GRD) is used as an output compare register, the corresponding IMFA, IMFB, IMFC, or IMFD flag is set to 1 when TRCCNT matches the general register.

The compare match signal is generated in the last state in which the values match (when TRCCNT is updated from the matching count to the next count). Therefore, when TRCCNT matches a general register, the compare match signal is generated only after the next TRCCNT clock pulse is input.

Figure 13.30 shows the timing of the IMFA to IMFD flag setting at compare match.

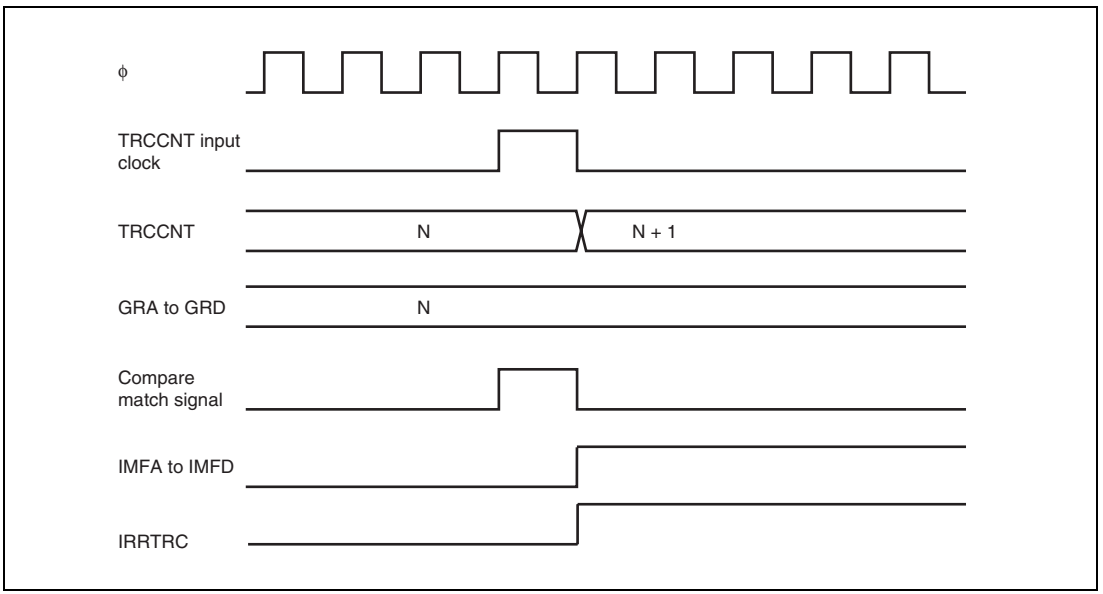



**Figure 13.30 Timing of IMFA to IMFD Flag Setting at Compare Match**

### 13.5.7 Timing of IMFA to IMFD Setting at Input Capture

If a general register (GRA, GRB, GRC, or GRD) is used as an input capture register, the corresponding IMFA, IMFB, IMFC, or IMFD flag is set to 1 when an input capture occurs. Figure 13.31 shows the timing of the IMFA to IMFD flag setting at input capture.

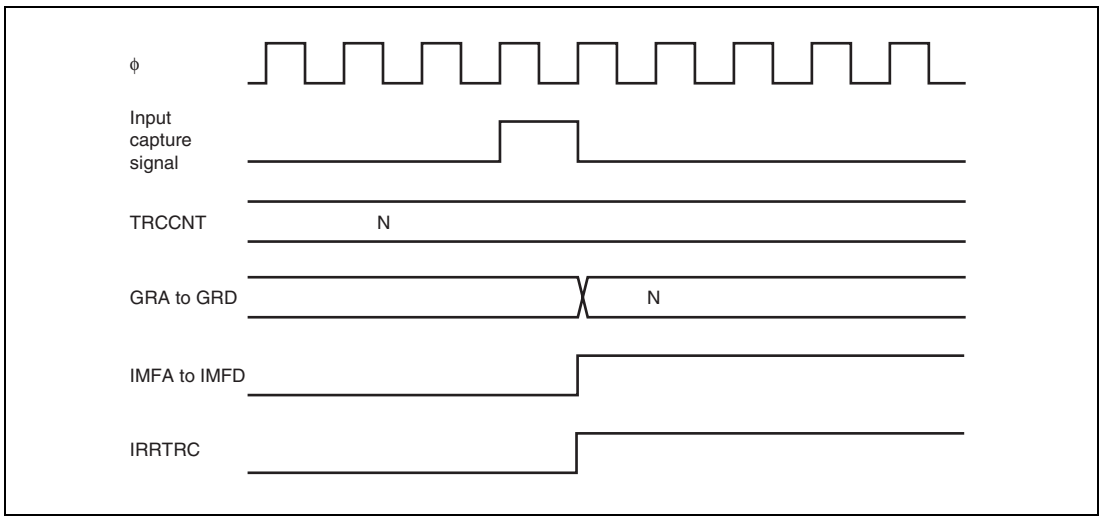



**Figure 13.31 Timing of IMFA to IMFD Flag Setting at Input Capture**

### 13.5.8 Timing of Status Flag Clearing

When the CPU reads a status flag while it is set to 1, then writes 0 in the status flag, the status flag is cleared. Figure 13.32 shows the status flag clearing timing.

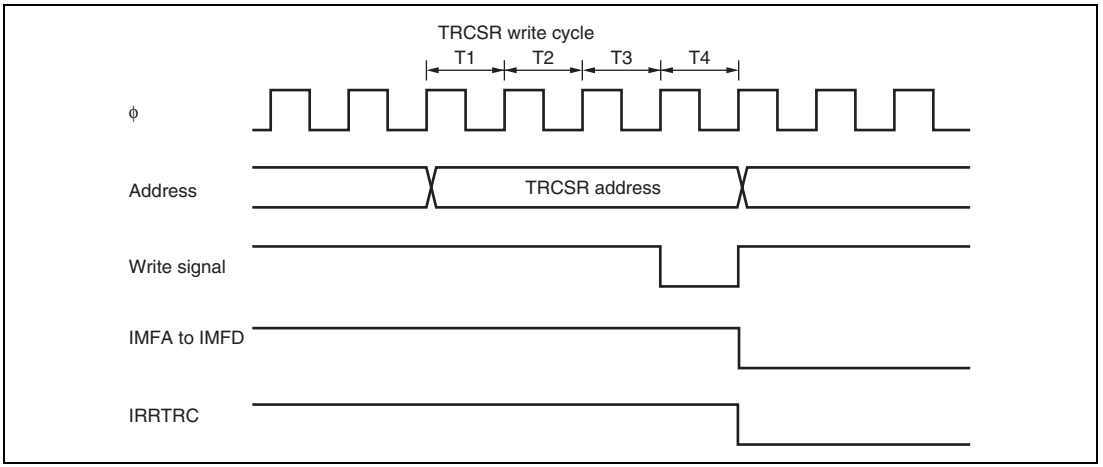



**Figure 13.32 Timing of Status Flag Clearing by CPU**

## 13.6 Usage Notes

The following types of contention or operation can occur in timer RC operation.

1. The pulse width of the input clock signal and the input capture signal must be at least three system clock (ϕ) cycles when the CKS2 to CKS0 bits in TRCCR1 = B'0XX or B'10X, and at least three on-chip oscillator clock (ϕ40M) cycles for B'110; shorter pulses will not be detected correctly.
2. Writing to registers is performed in the T4 state of a TRCCNT write cycle.

   If counter clear signal occurs in the T4 state of a TRCCNT write cycle, clearing of the counter takes priority and the write is not performed, as shown in figure 13.33. If counting-up is generated in the TRCCNT write cycle to contend with the TRCCNT counting-up, writing takes precedence.
3. TRCCNT may erroneously count up when switching internal clocks. TRCCNT counts the rising edge of the divided system clock (ϕ) when the internal clock is selected. If clocks are switched as shown in figure 13.34, the change from the low level of the previous clock to the high level of the new clock is considered as the rising edge. In this case, TRCCNT counts up erroneously.
4. If timer RC enters the module standby mode while an interrupt is being requested, the interrupt request cannot be cleared. Before entering the module standby mode, disable interrupt requests.

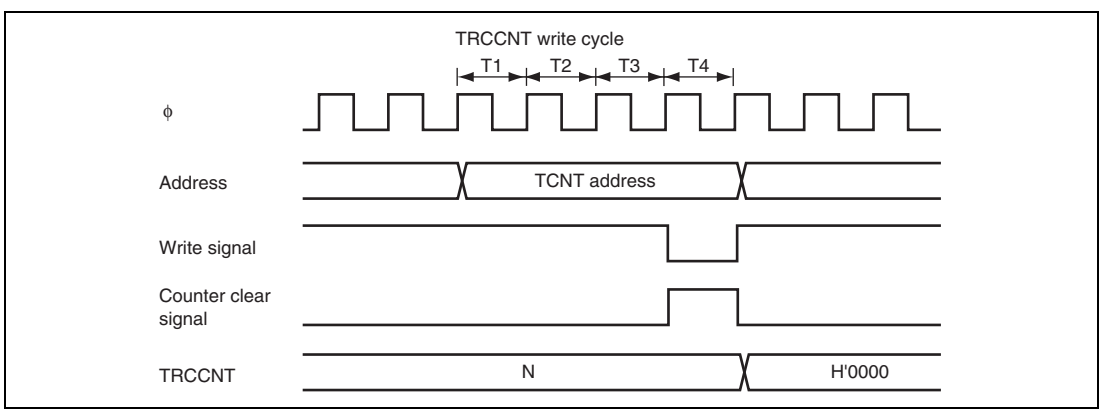



**Figure 13.33 Contention between TRCCNT Write and Clear**

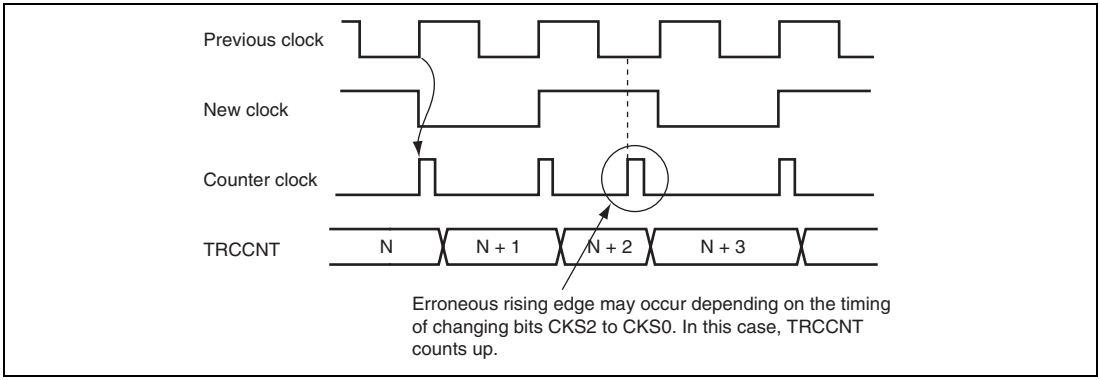



**Figure 13.34 Internal Clock Switching and TRCCNT Operation**

5. The TOA to TOD bits in TRCCR1 decide the value of the FTIO pin, which is output until the first compare match occurs. Once a compare match occurs and this compare match changes the values of FTIOA to FTIOD output, the values of the FTIOA to FTIOD pin output and the values read from the TOA to TOD bits may differ. Moreover, when the writing to TRCCR1 and the generation of the compare match A to D occur at the same timing, the writing to TRCCR1 has the priority. Thus, output change due to the compare match is not reflected to the FTIOA to FTIOD pins. Therefore, when bit manipulation instruction is used to write to TRCCR1, the values of the FTIOA to FTIOD pin output may result in an unexpected result. When TRCCR1 is to be written to while compare match is operating, stop the counter once before accessing to TRCCR1, read the port 8 state to reflect the values of FTIOA to FTIOD output, to TOA to TOD, and then restart the counter. Figure 13.35 shows an example when the compare match and the bit manipulation instruction to TRCCR1 occur at the same timing.

TRCCR1 has been set to H'06. Compare match B and compare match C are used. The FTIOB pin output 1, and is set to the toggle output or the 0 output on compare match B.
When the TOC bit is cleared (the FTIOC signal is low) by execution of BCLR #2, @TRCCR1 and compare match B occurs at the same timing as shown below, writing H'02 to TRCCR has priority and the FTIOB signal is not driven low on compare match B; the FTIOB signal remains high.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRCCR1 | CCLR | CKS2 | CKS1 | CKS0 | TOD | TOC | TOB | TOA |
| Setting | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

BCLR #2, @TRCCR1
(1) TRCCR1 is read as H'06.
(2) TRCCR1 is modified from H'06 to H'02.
(3) H'02 is written to TRCCR1.

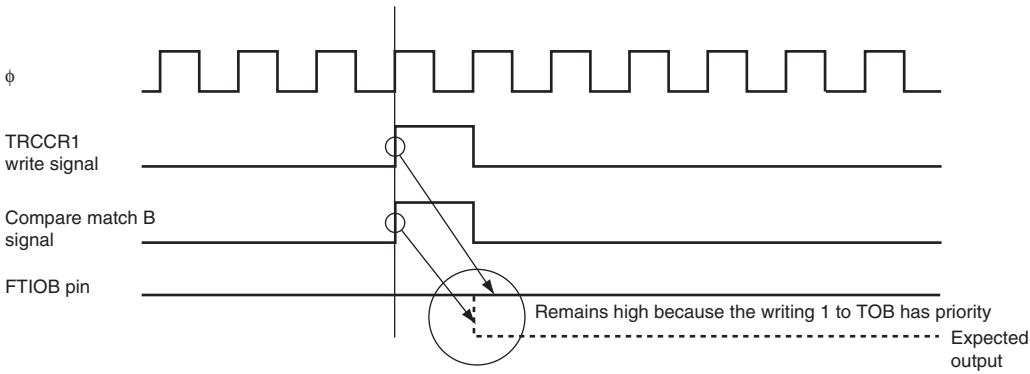



**Figure 13.35 When Compare Match and Bit Manipulation Instruction to TRCCR1 Occur at the Same Timing**

# Section 14 Timer RD

This LSI has two units of 16-bit timers (timer RD_0 and timer RD_1), each of which has two channels. Table 14.1 lists the timer RD functions, table 14.2 lists the channel configuration of timer RD, and figure 14.1 is a block diagram of the entire timer RD. Block diagrams of channels 1 and 2 are shown in figures 14.2 and 14.3.

Timer RD_1 has the same functions as timer RD_0. Therefore, the unit number (_0 or _1) is not explicitly mentioned in this section unless otherwise noted.

## 14.1 Features

- Capability to process up to eight inputs/outputs 0
- Eight general registers (GR): four registers for each channel
  - Independently assignable output compare or input capture functions
- Selection of seven counter clock sources: six internal clocks (ϕ, ϕ/2, ϕ/4, ϕ/8, ϕ/32, and ϕ40M which is a 40-MHz/32-MHz clock derived from the on-chip oscillator) and an external clock
- Seven selectable operating modes
  - Timer mode
    Output compare function (Selection of 0 output, 1 output, or toggle output)
    Input capture function (Rising edge, falling edge, or both edges)
  - Synchronous operation
    Timer counters_0 and _1 (TRDCNT_0 and TRDCNT_1) can be written simultaneously.
    Simultaneous clearing by compare match or input capture is possible.
  - PWM mode
    Up to six-phase PWM output can be provided with desired duty ratio.
  - PWM3 mode
    One-phase PWM output for non-overlapped normal and counter phases
  - Reset synchronous PWM mode
    Three-phase PWM output for normal and counter phases
  - Complementary PWM mode
    Three-phase PWM output for non-overlapped normal and counter phases
    The A/D conversion start trigger can be set for PWM cycles.
  - Buffer operation
    The input capture register can be consisted of double buffers.
    The output compare register can automatically be modified.

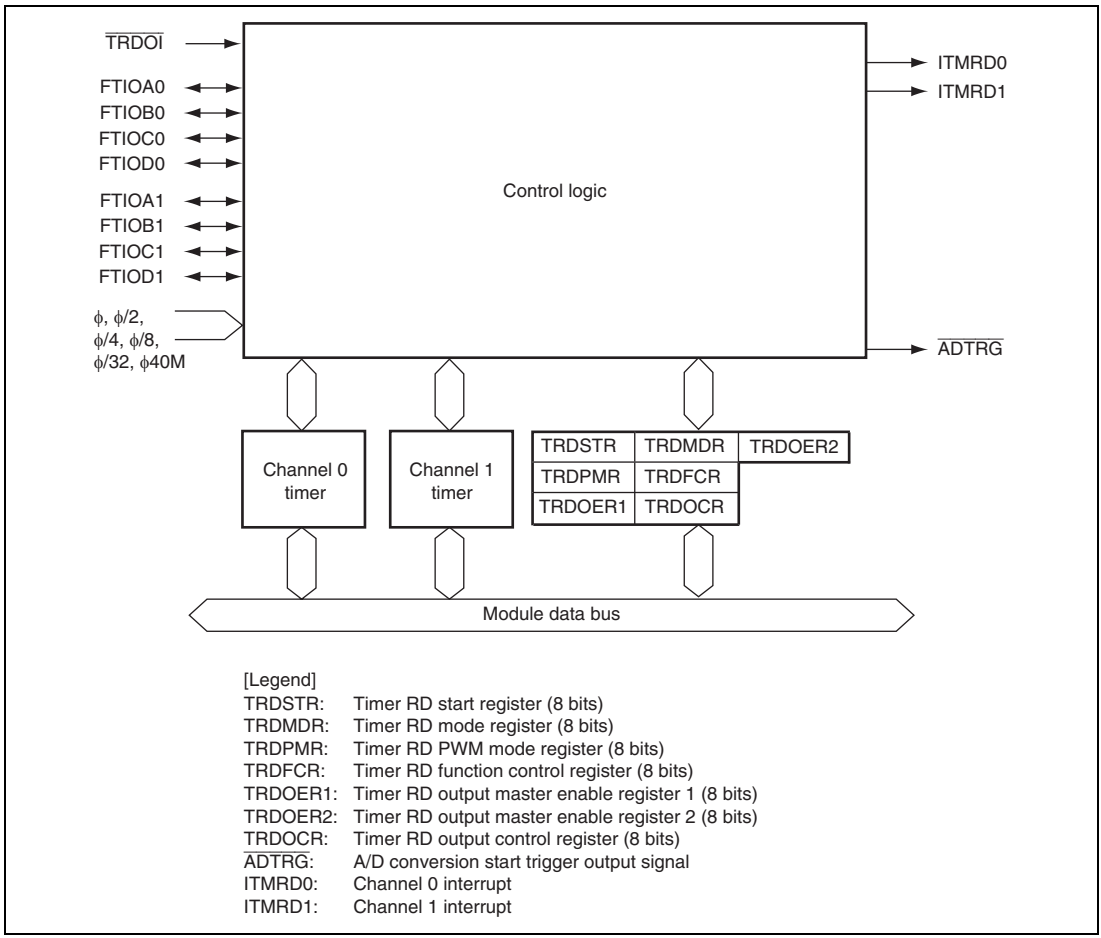
- High-speed access by the internal 16-bit bus
  - 16-bit TRDCNT and GR registers can be accessed in high speed by a 16-bit bus interface
- Any initial timer output value can be set
- Output of the timer is disabled by external trigger
- Eleven interrupt sources
  - Four compare match/input capture interrupts and an overflow interrupt are available for each channel. An underflow interrupt can be set for channel 1.

**Table 14.1 Timer RD Functions**

| Item | | Channel 0 | Channel 1 |
|---|---|---|---|
| Count clock | | Internal clocks: ϕ, ϕ/2, ϕ/4, ϕ/8, ϕ/32, ϕ40M<br>External clock: FTIOA0 (TCLK) | |
| General registers (output compare/input capture registers) | | GRA_0, GRB_0, GRC_0, GRD_0 | GRA_1, GRB_1, GRC_1, GRD_1 |
| Buffer register | | GRC_0, GRD_0 | GRC_1, GRD_1 |
| I/O pins | | FTIOA0, FTIOB0, FTIOC0, FTIOD0 | FTIOA1, FTIOB1, FTIOC1, FTIOD1 |
| Counter clearing function | | Compare match/input capture of GRA_0, GRB_0, GRC_0, or GRD_0 | Compare match/input capture of GRA_1, GRB_1, GRC_1, or GRD_1 |
| Compare match output | 0 output | Yes | Yes |
| | 1 output | Yes | Yes |
| | output | Yes | Yes |
| Input capture function | | Yes | Yes |
| Synchronous operation | | Yes | Yes |
| PWM mode | | Yes | Yes |
| PWM3 mode | | Yes | Yes |
| Reset synchronous PWM mode | | Yes | Yes |
| Complementary PWM mode | | Yes | Yes |
| Buffer function | | Yes | Yes |
| Interrupt sources | | Compare match/ input capture A0 to D0<br>Overflow | Compare match/ input capture A1 to D1<br>Overflow<br>Underflow |

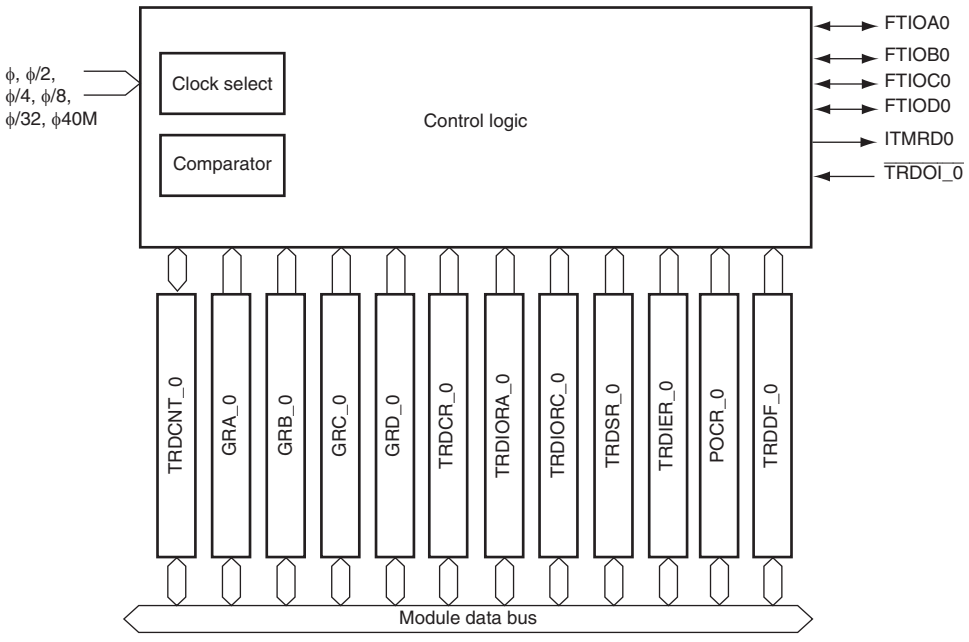
**Table 14.2 Channel Configuration of Timer RD**

| Unit | Channel | Pin |
|---|---|---|
| Timer RD_0 | 0 | FTIOA0 |
| | | FTIOB0 |
| | | FTIOC0 |
| | | FTIOD0 |
| | 1 | FTIOA1 |
| | | FTIOB1 |
| | | FTIOC1 |
| | | FTIOD1 |
| | Shared by channels 0 and 1 | $\overline{\text{TRDOI_0}}$ |
| Timer RD_1 | 2 | FTIOA2 |
| | | FTIOB2 |
| | | FTIOC2 |
| | | FTIOD2 |
| | 3 | FTIOA3 |
| | | FTIOB3 |
| | | FTIOC3 |
| | | FTIOD3 |
| | Shared by channels 2 and 3 | $\overline{\text{TRDOI_1}}$ |




[Legend]
TRDSTR: Timer RD start register (8 bits)
TRDMDR: Timer RD mode register (8 bits)
TRDPMR: Timer RD PWM mode register (8 bits)
TRDFCR: Timer RD function control register (8 bits)
TRDOER1: Timer RD output master enable register 1 (8 bits)
TRDOER2: Timer RD output master enable register 2 (8 bits)
TRDOCR: Timer RD output control register (8 bits)
$\overline{\text{ADTRG}}$: A/D conversion start trigger output signal
ITMRD0: Channel 0 interrupt
ITMRD1: Channel 1 interrupt

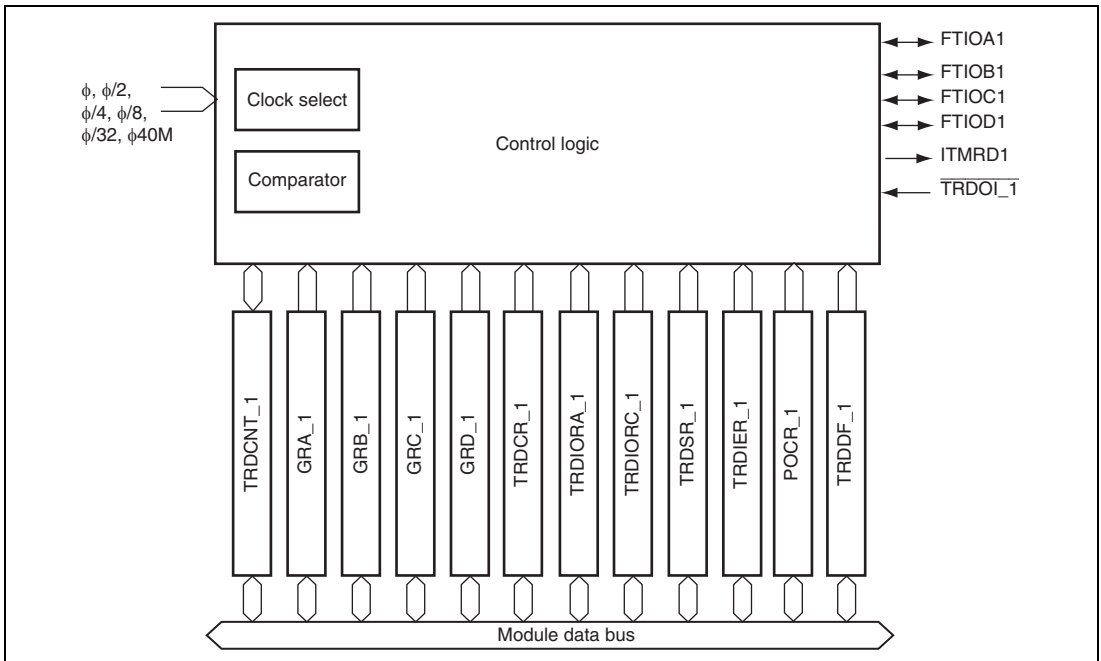
**Figure 14.1 Timer RD Block Diagram**




[Legend]

| | |
|---|---|
| TRDCNT_0: | Timer RD counter_0 (16 bits) |
| GRA_0, GRB_0, | General registers A_0, B_0, C_0, and D_0 |
| GRC_0, GRD_0: | (input capture/output compare registers: 16 bits × 4) |
| TRDCR_0: | Timer RD control register_0 (8 bits) |
| TRDIORA_0: | Timer RD I/O control register A_0 (8 bits) |
| TRDIORC_0: | Timer RD I/O control register C_0 (8 bits) |
| TRDSR_0: | Timer RD status register_0 (8 bits) |
| TRDIER_0: | Timer RD interrupt enable register_0 (8 bits) |
| POCR_0: | PWM mode output level control register_0 (8 bits) |
| TRDDF_0: | Timer RD digital filtering function select register_0 (8 bits) |
| ITMRD0: | Channel 0 interrupt |

**Figure 14.2 Timer RD (Channel 0) Block Diagram**




[Legend]

| | |
|---|---|
| TRDCNT_1 | Timer RD counter_1 (16 bits) |
| GRA_1, GRB_1, | General registers A_1, B_1, C_1, and D_1 |
| GRC_1, GRD_1: | (input capture/output compare registers: 16 bits × 4) |
| TRDCR_1: | Timer RD control register_1 (8 bits) |
| TRDIORA_1: | Timer RD I/O control register A_1 (8 bits) |
| TRDIORC_1: | Timer RD I/O control register C_1 (8 bits) |
| TRDSR_1: | Timer RD status register_1 (8 bits) |
| TRDIER_1: | Timer RD interrupt enable register_1 (8 bits) |
| POCR_1: | PWM mode output level control register_1 (8 bits) |
| TRDDF_1: | Timer RD digital filtering function select register_1 (8 bits) |
| ITMRD1: | Channel 1 interrupt |

**Figure 14.3 Timer RD (Channel 1) Block Diagram**

## 14.2 Input/Output Pins

Table 14.3 summarizes the timer RD pins.

**Table 14.3 Pin Configuration**

| Name | Abbreviation | Input/Output | Function |
|---|---|---|---|
| Input capture/ output compare A0 | FTIOA0 | Input/output | GRA_0 output compare output, GRA_0 input capture input, or external clock input (TCLK) |
| Input capture/ output compare B0 | FTIOB0 | Input/output | GRB_0 output compare output, GRB_0 input capture input, or PWM output |
| Input capture/ output compare C0 | FTIOC0 | Input/output | GRC_0 output compare output, GRC_0 input capture input, or PWM synchronous output (in reset synchronous PWM and complementary PWM modes) |
| Input capture/ output compare D0 | FTIOD0 | Input/output | GRD_0 output compare output, GRD_0 input capture input, or PWM output |
| Input capture/ output compare A1 | FTIOA1 | Input/output | GRA_1 output compare output, GRA_1 input capture input, or PWM output (in reset synchronous PWM and complementary PWM modes) |
| Input capture/ output compare B1 | FTIOB1 | Input/output | GRB_1 output compare output, GRB_1 input capture input, or PWM output |
| Input capture/ output compare C1 | FTIOC1 | Input/output | GRC_1 output compare output, GRC_1 input capture input, or PWM output |
| Input capture/ output compare D1 | FTIOD1 | Input/output | GRD_1 output compare output, GRD_1 input capture input, or PWM output |
| Timer output control | $\overline{\text{TRDOI}}$ | Input | Input pin for timer output disabling signal |

## 14.3 Register Descriptions

Timer RD has the following registers.

Common

- Timer RD start register (TRDSTR)
- Timer RD mode register (TRDMDR)
- Timer RD PWM mode register (TRDPMR)
- Timer RD function control register (TRDFCR)
- Timer RD output master enable register 1 (TRDOER1)
- Timer RD output master enable register 2 (TRDOER2)
- Timer RD output control register (TRDOCR)

Channel 0

- Timer RD control register_0 (TRDCR_0)
- Timer RD I/O control register A_0 (TRDIORA_0)
- Timer RD I/O control register C_0 (TRDIORC_0)
- Timer RD status register_0 (TRDSR_0)
- Timer RD interrupt enable register_0 (TRDIER_0)
- PWM mode output level control register_0 (POCR_0)
- Timer RD digital filtering function select register_0 (TRDDF_0)
- Timer RD counter_0 (TRDCNT_0)
- General register A_0 (GRA_0)
- General register B_0 (GRB_0)
- General register C_0 (GRC_0)
- General register D_0 (GRD_0)

Channel 1

- Timer RD control register_1 (TRDCR_1)
- Timer RD I/O control register A_1 (TRDIORA_1)
- Timer RD I/O control register C_1 (TRDIORC_1)
- Timer RD status register_1 (TRDSR_1)
- Timer RD interrupt enable register_1 (TRDIER_1)
- PWM mode output level control register_1 (POCR_1)
- Timer RD digital filtering function select register_1 (TRDDF_1)

- Timer RD counter_1 (TRDCNT_1)
- General register A_1 (GRA_1)
- General register B_1 (GRB_1)
- General register C_1 (GRC_1)
- General register D_1 (GRD_1)

### 14.3.1 Timer RD Start Register (TRDSTR)

TRDSTR selects the operation/stop for the TRDCNT counter. Use a MOV instruction to modify this register.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 4 | — | All 1 | — | Reserved<br>These bits are always read as 1, and cannot be modified. |
| 3 | CSTPN1 | 1 | R/W | Channel 1 Counter Stop<br>0: Counting is stopped on a compare match of TRDCNT_1 and GRA_1<br>1: Counting is continued on a compare match of TRDCNT_1 and GRA_1<br>Set this bit to 1 to restart counting after the counting has been stopped on a compare match. |
| 2 | CSTPN0 | 1 | R/W | Channel 0 Counter Stop<br>0: Counting is stopped on a compare match of TRDCNT_0 and GRA_0<br>1: Counting is continued on a compare match of TRDCNT_0 and GRA_0<br>Set this bit to 1 to restart counting after the counting has been stopped on a compare match. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | STR1 | 0 | R/W | Channel 1 Counter Start<br>TRDCNT_1 stops counting when this bit is 0, while it performs counting when this bit is 1.<br>[Setting condition]<br>• When 1 is written in STR1<br>[Clearing conditions]<br>• When 0 is written in STR1 while CSTPN1 = 1<br>• When the compare match A1 signal is generated while CSTPN1 = 0 |
| 0 | STR0 | 0 | R/W | Channel 0 Counter Start<br>TRDCNT_0 stops counting when this bit is 0, while it performs counting when this bit is 1.<br>[Setting condition]<br>• When 1 is written in STR0<br>[Clearing conditions]<br>• When 0 is written in STR0 while CSTPN0 = 1<br>• When the compare match A0 signal is generated while CSTPN0 = 0 |

Figures 14.4 and 14.5 show examples of stopping operation of the counter in PWM3 mode, when the CCLR2 to CCLR0 bits in TRDCR are set to clear TRDCNT_0 on GRA_0 compare match. For details on PWM3 mode, refer to section 14.4.8, PWM3 Mode Operation.

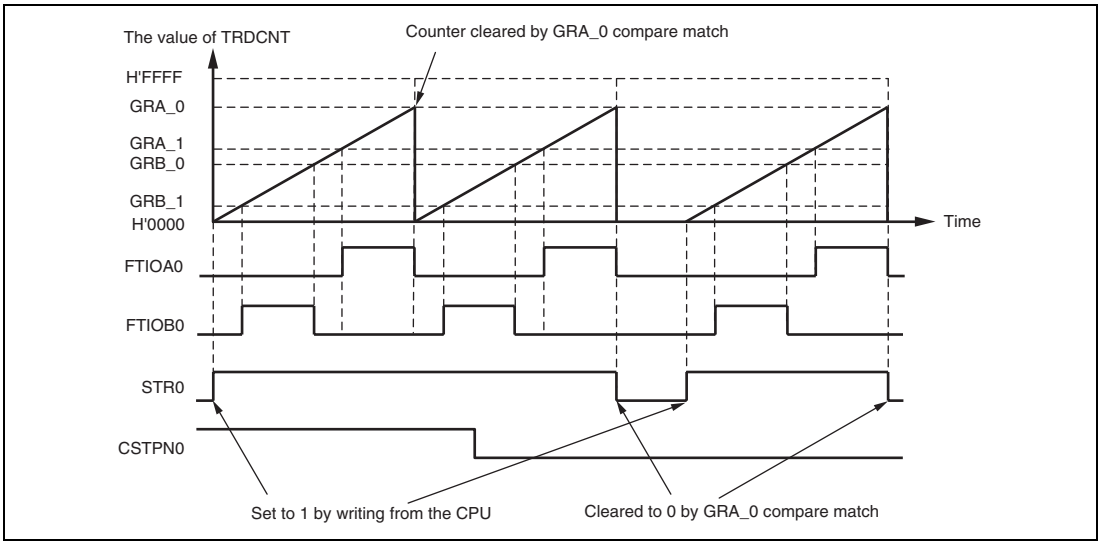



**Figure 14.4 Example (1) of Stopping Operation of the Counter (in PWM3 Mode)**

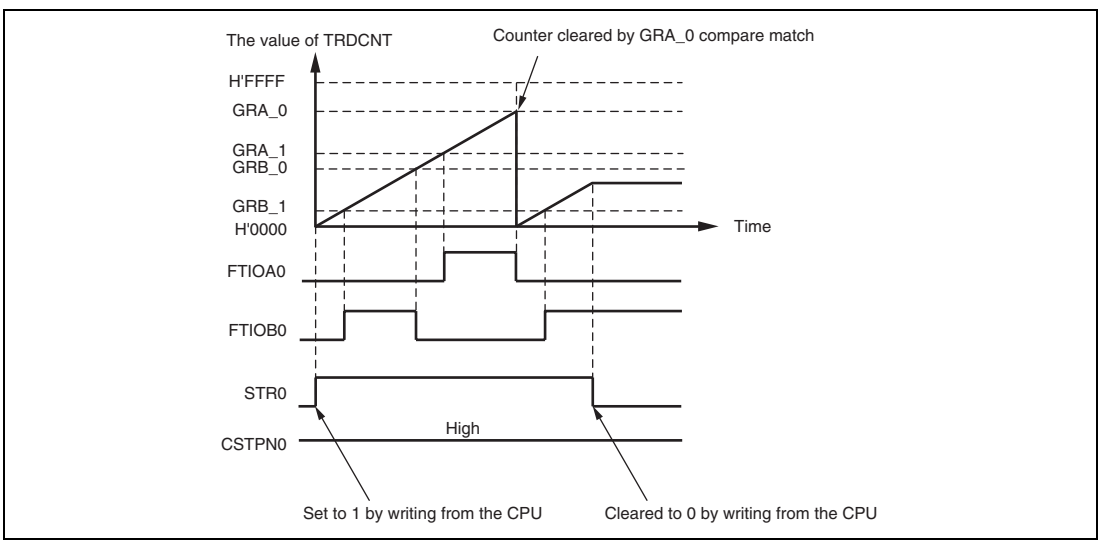



**Figure 14.5 Example (2) of Stopping Operation of the Counter (in PWM3 Mode)**

Figure 14.6 shows an example of starting and stopping operations of counters in PWM3 mode, when TRDCNT_0 is set to be cleared and stopped on GRA_0 compare match (CCLR2 to CCLR0 = 001, CSTPNT0 = 0) and TRDCNT_1 is used as a free-running counter. When TRDCNT_1 starts counting by setting the STR1 bit to 1 after TRDCNT_0 has started counting by setting the STR0 bit to 1, set 0 in the STR0 bit and 1 in the STR1 bit by using a MOV instruction. If the bit manipulation instruction is used to set 1 in the STR1 bit, there is a possibility that the STR0 bit is set to 1 after the counting has stopped on GRA_0 compare match, and that TRDCNT_0 starts counting again.

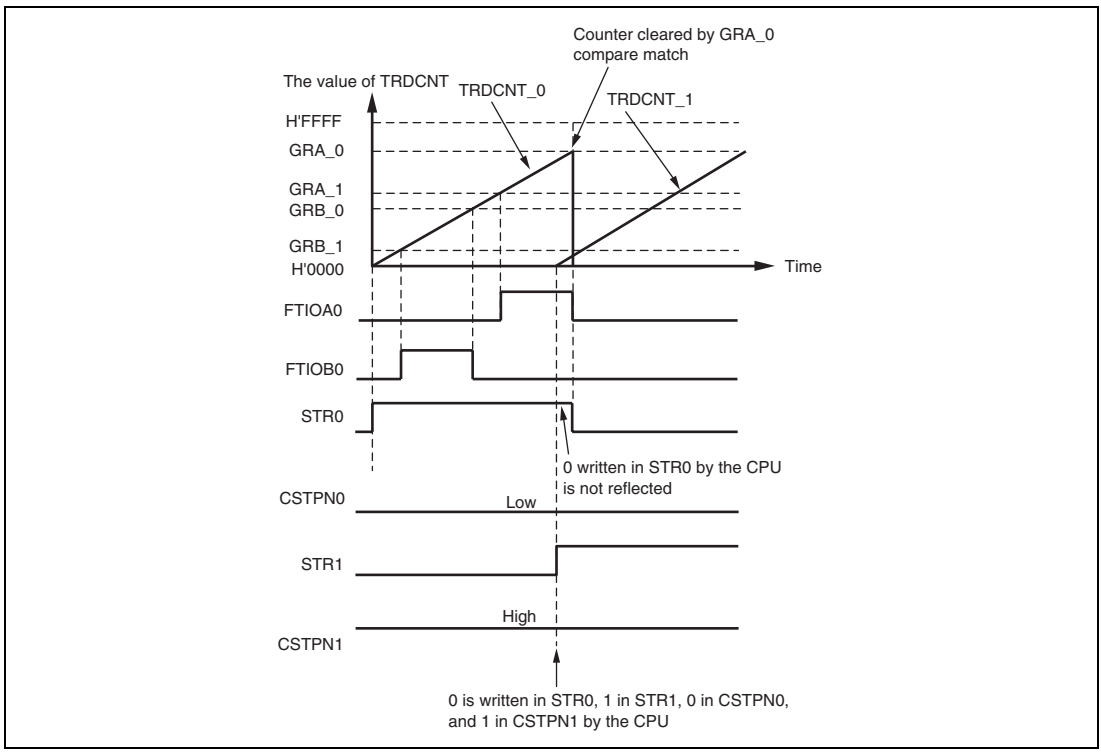



**Figure 14.6 Example of Starting and Stopping Operations of Counters (in PWM3 Mode)**

### 14.3.2 Timer RD Mode Register (TRDMDR)

TRDMDR selects buffer operation settings and synchronized operation.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | BFD1 | 0 | R/W | Buffer Operation D1<br>0: GRD_1 operates normally<br>1: GRB_1 and GRD_1 are used together for buffer operation |
| 6 | BFC1 | 0 | R/W | Buffer Operation C1<br>0: GRC_1 operates normally<br>1: GRA_1 and GRD_1 are used together for buffer operation |
| 5 | BFD0 | 0 | R/W | Buffer Operation D0<br>0: GRD_0 operates normally<br>1: GRB_0 and GRD_0 are used together for buffer operation |
| 4 | BFC0 | 0 | R/W | Buffer Operation C0<br>0: GRC_0 operates normally<br>1: GRA_0 and GRC_0 are used together for buffer operation |
| 3 to 1 | — | All 1 | — | Reserved<br>These bits are always read as 1, and cannot be modified. |
| 0 | SYNC | 0 | R/W | Timer Synchronization<br>0: TRDCNT_1 and TRDCNT_0 operate as independent timer counters<br>1: TRDCNT_1 and TRDCNT_0 operate synchronously<br>TRDCNT_1 and TRDCNT_0 can be pre-set or cleared synchronously |

### 14.3.3 Timer RD PWM Mode Register (TRDPMR)

TRDPMR sets the pin to enter PWM mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 1 | — | Reserved<br>This bit is always read as 1, and cannot be modified. |
| 6 | PWMD1 | 0 | R/W | PWM Mode D1<br>0: FTIOD1 operates normally<br>1: FTIOD1 operates in PWM mode |
| 5 | PWMC1 | 0 | R/W | PWM Mode C1<br>0: FTIOC1 operates normally<br>1: FTIOC1 operates in PWM mode |
| 4 | PWMB1 | 0 | R/W | PWM Mode B1<br>0: FTIOB1 operates normally<br>1: FTIOB1 operates in PWM mode |
| 3 | — | 1 | — | Reserved<br>This bit is always read as 1, and cannot be modified. |
| 2 | PWMD0 | 0 | R/W | PWM Mode D0<br>0: FTIOD0 operates normally<br>1: FTIOD0 operates in PWM mode |
| 1 | PWMC0 | 0 | R/W | PWM Mode C0<br>0: FTIOC0 operates normally<br>1: FTIOC0 operates in PWM mode |
| 0 | PWMB0 | 0 | R/W | PWM Mode B0<br>0: FTIOB0 operates normally<br>1: FTIOB0 operates in PWM mode |

### 14.3.4 Timer RD Function Control Register (TRDFCR)

TFCR selects the settings and output levels for each operating mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PWM3 | 1 | R/W | PWM3 Mode Select<br>Selects the PWM3 mode.<br>0: PWM3 mode is selected<br>1: PWM3 mode is not selected<br>This bit is valid when both bits CMD1 and CMD0 are cleared to 0. When PWM3 mode is selected, TRDPMR, TRDIORA, and TRDIORC are invalid. |
| 6 | STCLK | 0 | R/W | External Clock Input Select<br>0: External clock input is disabled<br>1: External clock input is enabled |
| 5 | ADEG | 0 | R/W | A/D Trigger Edge Select<br>The A/D converter registers should be set so that A/D conversion is started by an external trigger.<br>0: The A/D trigger signal is asserted when TRDCNT_0 matches GRA_0 in complementary PWM mode<br>1: The A/D trigger signal is asserted when TRDCNT_1 underflows in complementary PWM mode |
| 4 | ADTRG | 0 | R/W | External Trigger Disable<br>0: A/D trigger for PWM cycles is disabled in complementary PWM mode<br>1: A/D trigger for PWM cycles is enabled in complementary PWM mode |
| 3 | OLS1 | 0 | R/W | Output Level Select 1<br>Selects the counter-phase output levels in reset synchronous PWM mode or complementary PWM mode.<br>0: Initial output is high and the active level is low.<br>1: Initial output is low and the active level is high. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | OLS0 | 0 | R/W | Output Level Select 0<br>Selects the normal-phase output levels in reset synchronous PWM mode or complementary PWM mode.<br>0: Initial output is high and the active level is low.<br>1: Initial output is low and the active level is high.<br>Figure 14.7 shows an example of outputs in reset synchronous PWM mode and complementary PWM mode when OLS1 = 0 and OLS0 = 0. |
| 1 | CMD1 | 0 | R/W | Combination Mode 1 and 0 |
| 0 | CMD0 | 0 | R/W | 00: Channel 0 and channel 1 operate normally<br>01: Channel 0 and channel 1 are used together to operate in reset synchronous PWM mode<br>10: Channel 0 and channel 1 are used together to operate in complementary PWM mode (transferred when TRDCNT_0 matches GRA_0)<br>11: Channel 0 and channel 1 are used together to operate in complementary PWM mode (transferred when TRDCNT_1 underflows)<br>Note: When the reset synchronous PWM mode or complementary PWM mode is selected by these bits, this setting has the priority to the settings for PWM mode by each bit in TRDPMR. Stop TRDCNT_0 and TRDCNT_1 before making settings for reset synchronous PWM mode or complementary PWM mode. |

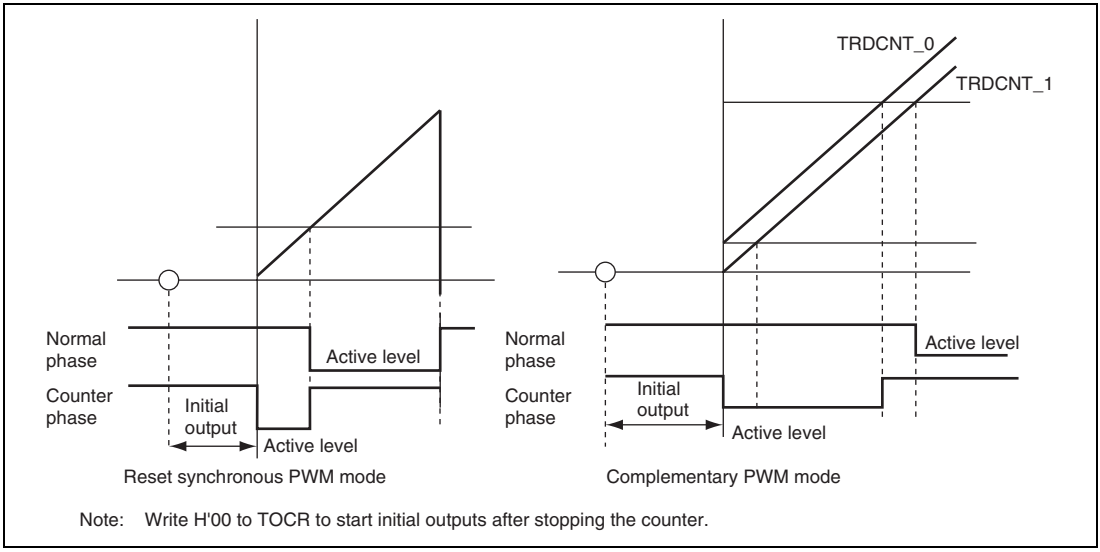



**Figure 14.7 Example of Outputs in Reset Synchronous PWM Mode and Complementary PWM Mode**

### 14.3.5 Timer RD Output Master Enable Register 1 (TRDOER1)

TRDOER1 enables/disables the outputs for channel 0 and channel 1. When $\overline{\text{TRDOI}}$ is selected for inputs, if a low level signal is input to $\overline{\text{TRDOI}}$, the bits in TRDOER1 are set to 1 to disable the output for timer RD.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | ED1 | 1 | R/W | Master Enable D1<br>0: FTIOD1 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORC_1 settings<br>1: FTIOD1 pin output is disabled regardless of the TRDMR, TRDFCR, and TRDIORC_1 settings (FTIOD1 pin is operated as an I/O port). |
| 6 | EC1 | 1 | R/W | Master Enable C1<br>0: FTIOC1 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORC_1 settings<br>1: FTIOC1 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORC_1 settings (FTIOC1 pin is operated as an I/O port). |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | EB1 | 1 | R/W | Master Enable B1<br>0: FTIOB1 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORA_1 settings<br>1: FTIOB1 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORA_1 settings (FTIOB1 pin is operated as an I/O port). |
| 4 | EA1 | 1 | R/W | Master Enable A1<br>0: FTIOA1 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORA_1 settings<br>1: FTIOA1 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORA_1 settings (FTIOA1 pin is operated as an I/O port). |
| 3 | ED0 | 1 | R/W | Master Enable D0<br>0: FTIOD0 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORC_0 settings<br>1: FTIOD0 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORC_0 settings (FTIOD0 pin is operated as an I/O port). |
| 2 | EC0 | 1 | R/W | Master Enable C0<br>0: FTIOC0 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORC_0 settings<br>1: FTIOC0 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORC_0 settings (FTIOC0 pin is operated as an I/O port). |
| 1 | EB0 | 1 | R/W | Master Enable B0<br>0: FTIOB0 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORA_0 settings<br>1: FTIOB0 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORA_0 settings (FTIOB0 pin is operated as an I/O port). |
| 0 | EA0 | 1 | R/W | Master Enable A0<br>0: FTIOA0 pin output is enabled according to the TRDPMR, TRDFCR, and TRDIORA_0 settings<br>1: FTIOA0 pin output is disabled regardless of the TRDPMR, TRDFCR, and TRDIORA_0 settings (FTIOA0 pin is operated as an I/O port). |

### 14.3.6 Timer RD Output Master Enable Register 2 (TRDOER2)

TRDOER2 selects the output disabled mode for channels 0 and 1.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PTO | 0 | R/W | Timer Output Disabled Mode<br>0: The corresponding bit in TRDOER1 is not set to 1 when the low level is input to the $\overline{\text{TRDOI}}$ pin<br>1: The corresponding bit in TRDOER1 is set to 1 when the low level is input to the $\overline{\text{TRDOI}}$ pin |
| 6 to 0 | — | All 1 | — | Reserved<br>These bits are always read as 1. |

### 14.3.7 Timer RD Output Control Register (TRDOCR)

TRDOCR selects the initial outputs before the first occurrence of a compare match. Note that bits OLS1 and OLS0 in TRDFCR set these initial outputs in reset synchronous PWM mode and complementary PWM mode.

In PWM3 mode, TRDOCR selects the output level on the FTIOB0 pin.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TOD1 | 0 | R/W | Output Level Select D1<br>0: 0 output at the FTIOD1 pin*<br>1: 1 output at the FTIOD1 pin* |
| 6 | TOC1 | 0 | R/W | Output Level Select C1<br>0: 0 output at the FTIOC1 pin*<br>1: 1 output at the FTIOC1 pin* |
| 5 | TOB1 | 0 | R/W | Output Level Select B1<br>0: 0 output at the FTIOB1 pin*<br>1: 1 output at the FTIOB1 pin* |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | TOA1 | 0 | R/W | Output Level Select A1<br>0: 0 output at the FTIOA1 pin*<br>1: 1 output at the FTIOA1 pin* |
| 3 | TOD0 | 0 | R/W | Output Level Select D0<br>0: 0 output at the FTIOD0 pin*<br>1: 1 output at the FTIOD0 pin* |
| 2 | TOC0 | 0 | R/W | Output Level Select C0<br>0: 0 output at the FTIOC0 pin*<br>1: 1 output at the FTIOC0 pin* |
| 1 | TOB0 | 0 | R/W | Output Level Select B0<br>• In modes other than PWM3 mode<br>0: 0 output at the FTIOB0 pin*<br>1: 1 output at the FTIOB0 pin*<br>• In PWM3 mode<br>0: 1 output at the FTIOB0 pin on GRB_1 compare match and 0 output at the FTIOB0 pin on GRB_0 compare match<br>1: 0 output at the FTIOB0 pin on GRB_1 compare match and 1 output at the FTIOB0 pin on GRB_0 compare match |
| 0 | TOA0 | 0 | R/W | Output Level Select A0<br>• In modes other than PWM3 mode<br>0: 0 output at the FTIOA0 pin*<br>1: 1 output at the FTIOA0 pin*<br>• In PWM3 mode<br>0: 1 output at the FTIOB0 pin on GRA_1 compare match and 0 output at the FTIOB0 pin on GRA_0 compare match<br>1: 0 output at the FTIOB0 pin on GRA_1 compare match and 1 output at the FTIOB0 pin on GRA_0 compare match |

Note: * The change of the setting is immediately reflected in the output value.

### 14.3.8 Timer RD Counter (TRDCNT)

Timer RD has two TRDCNT counters (TRDCNT_0 and TRDCNT_1), one for each channel. The TRDCNT counters are 16-bit readable/writable registers that increment/decrement according to input clocks. Input clocks can be selected by bits TPSC2 to TPSC0 in TRDCR. TRDCNT_0 and TRDCNT_1 increment/decrement in complementary PWM mode, while they only increment in other modes.

The TRDCNT counters are initialized to H'0000 by compare matches with corresponding GRA, GRB, GRC, or GRD, or input captures to GRA, GRB, GRC, or GRD (counter clearing function). When the TRDCNT counters overflow, an OVF flag in TRDSR for the corresponding channel is set to 1. When TRDCNT_1 underflows, an UDF flag in TRDSR is set to 1. The TRDCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit. TRDCNT is initialized to H'0000 by a reset.

### 14.3.9 General Registers A, B, C, and D (GRA, GRB, GRC, and GRD)

GR are 16-bit registers. Timer RD has eight general registers (GR), four for each channel. The GR registers are dual function 16-bit readable/writable registers, functioning as either output compare or input capture registers. Functions can be switched by TRDIORA and TRDIORC.

The values in GR and TRDCNT are constantly compared with each other when the GR registers are used as output compare registers. When the both values match, the IMFA to IMFD flags in TSR are set to 1. Compare match outputs can be selected by TRDIORA and TRDIORC.

When the GR registers are used as input capture registers, the TRDCNT value is stored after detecting external signals. At this point, IMFA to IMFD flags in the corresponding TRDSR are set to 1. Detection edges for input capture signals can be selected by TRDIORA and TRDIORC.

When PWM mode, complementary PWM mode, or reset synchronous PWM mode is selected, the values in TRDIORA and TRDIORC are ignored. Upon reset, the GR registers are set as output compare registers (no output) and initialized to H'FFFF. The GR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

### 14.3.10 Timer RD Control Register (TRDCR)

TRDCR selects a TRDCNT counter clock, an edge when an external clock is selected, and counter clearing sources. Timer RD has a total of two TRDCR registers, one for each channel.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | CCLR2 | 0 | R/W | Counter Clear 2 to 0 |
| 6 | CCLR1 | 0 | R/W | 000: Disables TRDCNT clearing |
| 5 | CCLR0 | 0 | R/W | 001: Clears TRDCNT by GRA compare match/input capture*[1] |
| | | | | 010: Clears TRDCNT by GRB compare match/input capture*[1] |
| | | | | 011: Synchronization clear; Clears TRDCNT in synchronous with counter clearing of the other channel's timer*[2] |
| | | | | 100: Disables TRDCNT clearing |
| | | | | 101: Clears TRDCNT by GRC compare match/input capture*[1] |
| | | | | 110: Clears TRDCNT by GRD compare match/input capture*[1] |
| | | | | 111: Synchronization clear; Clears TRDCNT in synchronous with counter clearing of the other channel's timer*[2] |
| 4 | CKEG1 | 0 | R/W | Clock Edge 1 and 0 |
| 3 | CKEG0 | 0 | R/W | 00: Count at rising edge |
| | | | | 01: Count at falling edge |
| | | | | 1X: Count at both edges |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | TPSC2 | 0 | R/W | Time Prescaler 2 to 0 |
| 1 | TPSC1 | 0 | R/W | 000: Internal clock: count by $\phi$*[3] |
| 0 | TPSC0 | 0 | R/W | 001: Internal clock: count by $\phi$/2 |
| | | | | 010: Internal clock: count by $\phi$/4 |
| | | | | 011: Internal clock: count by $\phi$/8 |
| | | | | 100: Internal clock: count by $\phi$/32 |
| | | | | 101: External clock: count by FTIOA0 (TCLK) pin input |
| | | | | 110: Internal clock: count by $\phi$40M*[4] |
| | | | | 111: Reserved (setting prohibited) |
| | | | | Notes: 1. When selecting the internal clock $\phi$, the subclock is counted in subactive and subsleep modes. |
| | | | | 2. When selecting the internal clock $\phi$40M, on-chip oscillator should be in operation. When switching the clock, the counter should be halted. |

[Legend] X: Don't care

Notes: 1. When GR functions as an output compare register, TRDCNT is cleared by compare match. When GR functions as input capture, TRDCNT is cleared by input capture.

2. Synchronous operation is set by TRDMDR.

### 14.3.11 Timer RD I/O Control Registers (TRDIORA and TRDIORC)

TRDIOR control the general registers (GR). Timer RD has four TRDIOR registers (TRDIORA_0, TRDIORA_1, TRDIORC_0, and TRDIORC_1), two for each channel. In PWM mode, PWM3 mode, complementary PWM mode, and reset synchronous PWM mode, the settings of TRDIOR are invalid.

- TRDIORA

  TRDIORA selects whether GRA or GRB is used as an output compare register or an input capture register. When an output compare register is selected, the output setting is selected. When an input capture register is selected, an input edge of an input capture signal is selected. TRDIORA also selects the function of FTIOA or FTIOB pin.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 6 | IOB2 | 0 | R/W | I/O Control B2<br>Selects the GRB function.<br>0: GRB functions as an output compare register<br>1: GRB functions as an input capture register |
| 5 | IOB1 | 0 | R/W | I/O Control B1 and B0 |
| 4 | IOB0 | 0 | R/W | When IOB2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOB pin at GRB compare match<br>10: 1 output to the FTIOB pin at GRB compare match<br>11: Output toggles to the FTIOB pin at GRB compare match<br>When IOB2 = 1,<br>00: Input capture to GRB at rising edge at the FTIOB pin<br>01: Input capture to GRB at falling edge at the FTIOB pin<br>1X: Input capture to GRB at rising and falling edges at the FTIOB pin |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | — | 1 | — | Reserved<br>0 should not be written to this bit. |
| 2 | IOA2 | 0 | R/W | I/O Control A2<br>Selects the GRA function.<br>0: GRA functions as an output compare register<br>1: GRA functions as an input capture register |
| 1 | IOA1 | 0 | R/W | I/O Control A1 and A0 |
| 0 | IOA0 | 0 | R/W | When IOA2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOA pin at GRA compare match<br>10: 1 output to the FTIOA pin at GRA compare match<br>11: Output toggles to the FTIOA pin at GRA compare match<br>When IOA2 = 1,<br>00: Input capture to GRA at rising edge at the FTIOA pin<br>01: Input capture to GRA at falling edge at the FTIOA pin<br>1X: Input capture to GRA at rising and falling edges at the FTIOA pin |

[Legend]

X: Don't care.

Note: When a GR register functions as a buffer register for a paired GR register, the settings in the IOA2 and IOB2 bits in TRDIORA and the IOC2 and IOD2 bits in TRDIORC of both registers should be the same.

- TRDIORC

  TRDIORC selects whether GRC or GRD is used as an output compare register or an input capture register. When an output compare register is selected, the output setting is selected. When an input capture register is selected, an input edge of an input capture signal is selected. TRDIORC also selects the function of the FTIOA to FTIOD pins.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | IOD3 | 1 | R/W | I/O Control D3<br>Specifies GRD to be used as GR for the FTIOB or FTIOD pin.<br>0: GRD is used as GR for the FTIOB pin<br>1: GRD is used as GR for the FTIOD pin |
| 6 | IOD2 | 0 | R/W | I/O Control D2<br>Selects the GRD function.<br>0: GRD functions as an output compare register<br>1: GRD functions as an input capture register |
| 5 | IOD1 | 0 | R/W | I/O Control D1 and D0 |
| 4 | IOD0 | 0 | R/W | When IOD3 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOB pin at GRD compare match<br>10: 1 output to the FTIOB pin at GRD compare match<br>11: Output toggles to the FTIOB pin at GRD compare match<br>When IOD3 = 1 and IOD2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOD pin at GRD compare match<br>10: 1 output to the FTIOD pin at GRD compare match<br>11: Output toggles to the FTIOD pin at GRD compare match<br>When IOD3 = 1 and IOD2 = 1,<br>00: Input capture to GRD at rising edge at the FTIOD pin<br>01: Input capture to GRD at falling edge at the FTIOD pin<br>1X: Input capture to GRD at rising and falling edges at the FTIOD pin |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | IOC3 | 1 | R/W | I/O Control C3<br>Specifies GRC to be used as GR for the FTIOA or FTIOC pin.<br>0: GRC is used as GR for the FTIOA pin<br>1: GRC is used as GR for the FTIOC pin |
| 2 | IOC2 | 0 | R/W | I/O Control C2<br>Selects the GRC function.<br>0: GRC functions as an output compare register<br>1: GRC functions as an input capture register |
| 1 | IOC1 | 0 | R/W | I/O Control C1 and C0 |
| 0 | IOC0 | 0 | R/W | When IOC3 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOA pin at GRC compare match<br>10: 1 output to the FTIOA pin at GRC compare match<br>11: Output toggles to the FTIOA pin at GRC compare match<br>When IOC3 = 1 and IOC2 = 0,<br>00: No output at compare match<br>01: 0 output to the FTIOC pin at GRC compare match<br>10: 1 output to the FTIOC pin at GRC compare match<br>11: Output toggles to the FTIOC pin at GRC compare match<br>When IOC3 = 1 and IOC2 = 1,<br>00: Input capture to GRC at rising edge at the FTIOC pin<br>01: Input capture to GRC at falling edge at the FTIOC pin<br>1X: Input capture to GRC at rising and falling edges at the FTIOC pin |

[Legend]

X: Don't care.

Note: When a GR register functions as a buffer register for a paired GR register, the settings in the IOA2 and IOB2 bits in TRDIORA and the IOC2 and IOD2 bits in TRDIORC of both registers should be the same.

### 14.3.12 Timer RD Status Register (TRDSR)

TRDSR indicates generation of an overflow/underflow of TRDCNT and a compare match/input capture of GRA, GRB, GRC, and GRD. These flags are interrupt sources. If an interrupt is enabled by a corresponding bit in TRDIER, TRDSR requests an interrupt for the CPU. Timer RD has two TRDSR registers, one for each channel.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 5 | UDF* | 0 | R/W | Underflow Flag<br>[Setting condition]<br>• When TRDCNT_1 underflows<br>[Clearing condition]<br>• When 0 is written to UDF after reading UDF = 1 |
| 4 | OVF | 0 | R/W | Overflow Flag<br>[Setting condition]<br>• When the TRDCNT value underflows<br>[Clearing condition]<br>• When 0 is written to OVF after reading OVF = 1 |
| 3 | IMFD | 0 | R/W | Input Capture/Compare Match Flag D<br>[Setting conditions]<br>• When TRDCNT = GRD and GRD is functioning as output compare register<br>• When TRDCNT = GRD while the FTIOD pin operates in PWM mode<br>• When TRDCNT = GRD in PWM3 mode, reset synchronous PWM mode, or complementary PWM mode<br>• When TRDCNT value is transferred to GRD by input capture signal and GRD is functioning as input capture register<br>[Clearing condition]<br>• When 0 is written to IMFD after reading IMFD = 1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | IMFC | 0 | R/W | Input Capture/Compare Match Flag C<br>[Setting conditions]<br>• When TRDCNT = GRC and GRC is functioning as output compare register<br>• When TRDCNT = GRC while the FTIOC pin operates in PWM mode<br>• When TRDCNT = GRC in PWM3 mode, reset synchronous PWM mode, or complementary PWM mode<br>• When TRDCNT value is transferred to GRC by input capture signal and GRC is functioning as input capture register<br>[Clearing condition]<br>• When 0 is written to IMFC after reading IMFC = 1 |
| 1 | IMFB | 0 | R/W | Input Capture/Compare Match Flag B<br>[Setting conditions]<br>• When TRDCNT = GRB and GRB is functioning as output compare register<br>• When TRDCNT = GRB while the FTIOB pin operates in PWM mode<br>• When TRDCNT = GRB in PWM mode, PWM3 mode, reset synchronous PWM mode, or complementary PWM mode (in reset synchronous PWM mode, however, while TRDCNT_0 = GRB_1 and TRDCNT_0 = GRB_0)<br>• When TRDCNT value is transferred to GRB by input capture signal and GRB is functioning as input capture register<br>[Clearing condition]<br>• When 0 is written to IMFB after reading IMFB = 1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | IMFA | 0 | R/W | Input Capture/Compare Match Flag A<br>[Setting conditions]<br>• When TRDCNT = GRA and GRA is functioning as output compare register<br>• When TRDCNT = GRA in PWM mode, PWM3 mode, reset synchronous PWM mode, or complementary PWM mode (in reset synchronous PWM mode, however, while TRDCNT_0 = GRA_1 and TRDCNT_0 = GRA_0)<br>• When TRDCNT value is transferred to GRA by input capture signal and GRA is functioning as input capture register<br>[Clearing condition]<br>• When 0 is written to IMFA after reading IMFA = 1 |

Note: Bit 5 is not the UDF flag in TRDSR_0. It is a reserved bit. It is always read as 1.

### 14.3.13 Timer RD Interrupt Enable Register (TRDIER)

TRDIER enables or disables interrupt requests for overflow or GR compare match/input capture. Timer RD has two TRDIER registers, one for each channel.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 5 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 4 | OVIE | 0 | R/W | Overflow Interrupt Enable<br>0: Interrupt requests (OVI) by OVF or UDF flag are disabled<br>1: Interrupt requests (OVI) by OVF or UDF flag are enabled |
| 3 | IMIED | 0 | R/W | Input Capture/Compare Match Interrupt Enable D<br>0: Interrupt requests (IMID) by IMFD flag are disabled<br>1: Interrupt requests (IMID) by IMFD flag are enabled |
| 2 | IMIEC | 0 | R/W | Input Capture/Compare Match Interrupt Enable C<br>0: Interrupt requests (IMIC) by IMFC flag are disabled<br>1: Interrupt requests (IMIC) by IMFC flag are enabled |
| 1 | IMIEB | 0 | R/W | Input Capture/Compare Match Interrupt Enable B<br>0: Interrupt requests (IMIB) by IMFB flag are disabled<br>1: Interrupt requests (IMIB) by IMFB flag are enabled |
| 0 | IMIEA | 0 | R/W | Input Capture/Compare Match Interrupt Enable A<br>0: Interrupt requests (IMIA) by IMFA flag are disabled<br>1: Interrupt requests (IMIA) by IMFA flag are enabled |

### 14.3.14 PWM Mode Output Level Control Register (POCR)

POCR control the active level in PWM mode. Timer RD has two POCR registers, one for each channel.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 3 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 2 | POLD | 0 | R/W | PWM Mode Output Level Control D<br>0: The output level of FTIOD is low-active<br>1: The output level of FTIOD is high-active |
| 1 | POLC | 0 | R/W | PWM Mode Output Level Control C<br>0: The output level of FTIOC is low-active<br>1: The output level of FTIOC is high-active |
| 0 | POLB | 0 | R/W | PWM Mode Output Level Control B<br>0: The output level of FTIOB is low-active<br>1: The output level of FTIOB is high-active |

### 14.3.15 Timer RD Digital Filtering Function Select Register (TRDDF)

TRDDF enables or disables the digital filter for each of the FTIOA to FTIOD pins. The setting in this register is valid on the corresponding pin when the FTIOA to FTIOD inputs are enabled by TRDIORA and TRDIORC.

Timer RD has two TRDDF registers, one for each channel.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | DFCK1 | 0 | R/W | These bits select the clock to be used by the digital filter. |
| 6 | DFCK0 | 0 | R/W | 00: $\phi$/32<br>01: $\phi$/8<br>10: $\phi$<br>11: Clock specified by bits TPSC2 to TPSC0 in TRDCR |
| 5, 4 | — | All 0 | — | Reserved<br>These bits are always read as 0. |
| 3 | DFD | 0 | R/W | Enables or disables the digital filter for the FTIOD pin.<br>0: Disables the digital filter<br>1: Enables the digital filter |
| 2 | DFC | 0 | R/W | Enables or disables the digital filter for the FTIOC pin.<br>0: Disables the digital filter<br>1: Enables the digital filter |
| 1 | DFB | 0 | R/W | Enables or disables the digital filter for the FTIOB pin.<br>0: Disables the digital filter<br>1: Enables the digital filter |
| 0 | DFA | 0 | R/W | Enables or disables the digital filter for the FTIOA pin.<br>0: Disables the digital filter<br>1: Enables the digital filter |

### 14.3.16 Interface with CPU

#### (1) 16-Bit Register

TRDCNT and GR are 16-bit registers. Reading/writing in a 16-bit unit is enabled but disabled in an 8-bit unit since the data bus with the CPU is 16-bit width. These registers must always be accessed in a 16-bit unit. Figure 14.8 shows an example of accessing the 16-bit registers.

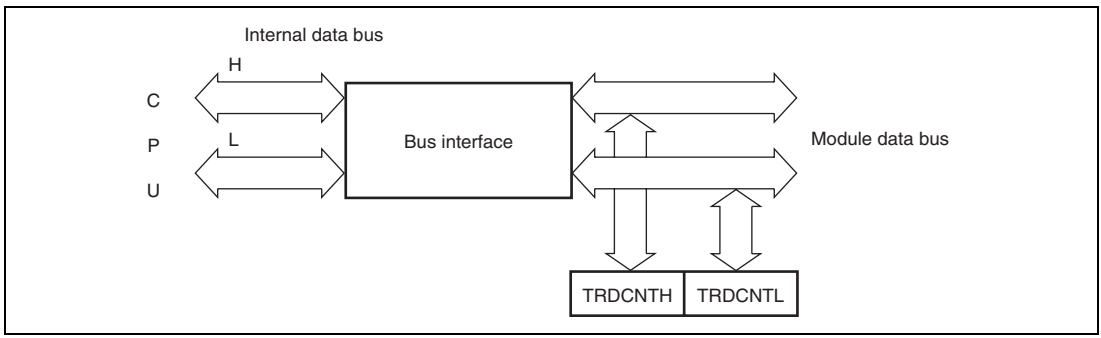



**Figure 14.8 Accessing Operation of 16-Bit Register (between CPU and TRDCNT (16 bits))**

#### (2) 8-Bit Register

Registers other than TRDCNT and GR are 8-bit registers that are connected internally with the CPU in an 8-bit width. Figure 14.9 shows an example of accessing the 8-bit registers.

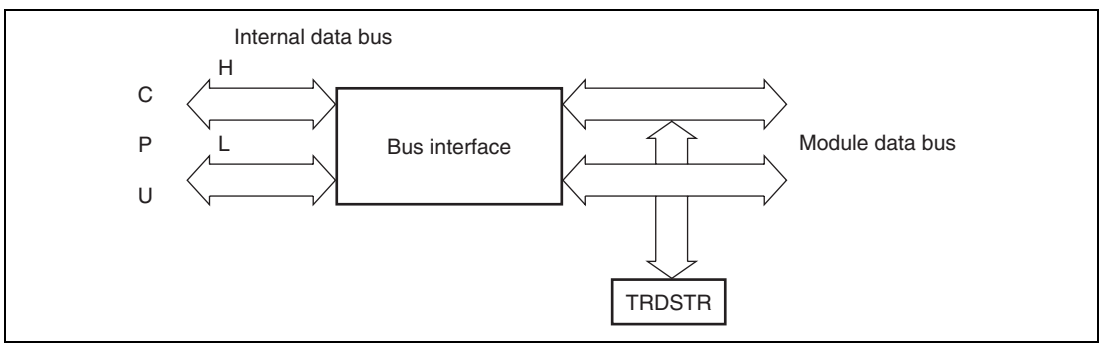



**Figure 14.9 Accessing Operation of 8-Bit Register (between CPU and TRDSTR (8 bits))**

## 14.4 Operation

Timer RD has the following operating modes.

- Timer mode operation
  - Enables output compare and input capture functions by setting the IOA2 to IOA0 and IOB2 to IOB0 bits in TRDIORA and the IOC3 to IOC0 and IOD3 to IOD0 bits in TRDIORC
- PWM mode operation
  - Enables PWM mode operation by setting TRDPMR
- PWM3 mode operation
  - Enables PWM3 mode operation by setting the PWM3 bit in TRDFCR
- Reset synchronous PWM mode operation
  - Enables reset synchronous PWM mode operation by setting the CMD1 and CMD0 bits in TRDFCR
- Complementary PWM mode operation
  - Enables complementary PWM mode operation by setting the CMD1 and CMD0 bits in TRDFCR

The FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pins indicate the timer operation mode by each register setting.

- FTIOA0 pin

| **Register Name** | **TRDOER1** | | **TRCMR** | | **TRDIORA** | |
|---|---|---|---|---|---|---|
| **Bit Name** | **EA0** | **STCLK** | **CMD1, CMD0** | **PWM3** | **IOA2 to IOA0** | **Function** |
| Setting values | 0 | 0 | 00 | 0 | XXX | PWM3 mode waveform output |
| | 0 | 0 | 00 | 1 | 001, 01X | Timer mode waveform output (output compare function) |
| | 0 | 1 | XX | 1 | 1XX | Timer mode (input capture function) |
| | 1 | | | | | |
| | Other than above | | | | | General I/O port |

[Legend]

X: Don't care.

- FTIOB0 pin

| Register Name | TRDOER1 | TRDFCR | | TRDPMR | TRDIORA | |
|---|---|---|---|---|---|---|
| Bit Name | EB0 | CMD1, CMD0 | PWM3 | PWMB0 | IOB2 to IOB0 | Function |
| Setting values | 0 | 10, 11 | X | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 0 | X | XXX | PWM3 mode waveform out |
| | 0 | 00 | 1 | 1 | XXX | PWM mode waveform out |
| | 0 | 00 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | 0 | 00 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | 1 | | | | | |
| | Other than above | | | | | General I/O port |

[Legend]

X: Don't care.

- FTIOC0 pin

| Register Name | TRDOER1 | TRDFCR | | TRDPMR | TRDIORC | |
|---|---|---|---|---|---|---|
| **Bit Name** | **EC0** | **CMD1, CMD0** | **PWM3** | **PWMC0** | **IOC2 to IOC0** | **Function** |
| Setting values | 0 | 10, 11 | X | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 1 | 1 | XXX | PWM mode waveform out |
| | 0 | 00 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | 0 | 00 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | 1 | | | | | |
| | Other than above | | | | | General I/O port |

[Legend]

X: Don't care.

- FTIOD0 pin

| Register Name | TRDOER1 | TRDFCR | | TRDPMR | TRDIORC | |
|---|---|---|---|---|---|---|
| **Bit Name** | **ED0** | **CMD1, CMD0** | **PWM3** | **PWMD0** | **IOD2 to IOD0** | **Function** |
| Setting values | 0 | 10, 11 | X | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 1 | 1 | XXX | PWM mode waveform out |
| | 0 | 00 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | 0 | 00 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | 1 | | | | | |
| | | Other than above | | | | General I/O port |

[Legend]
X: Don't care.

- FTIOA1 pin

| Register Name | TRDOER1 | TRDFCR | | TRDIORA | |
|---|---|---|---|---|---|
| **Bit Name** | **EA1** | **CMD1, CMD0** | **PWM3** | **IOA2 to IOA0** | **Function** |
| Setting values | 0 | 10, 11 | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 1 | 001, 01X | Timer mode waveform output (output compare function) |
| | 0 | 00 | 1 | 1XX | Timer mode (input capture function) |
| | 1 | | | | |
| | | Other than above | | | General I/O port |

[Legend]
X: Don't care.

- FTIOB1 pin

| Register Name | TRDOER1 | TRDFCR | | TRDPMR | TRDIORA | |
|---|---|---|---|---|---|---|
| **Bit Name** | **EB1** | **CMD1, CMD0** | **PWM3** | **PWMB1** | **IOB2 to IOB0** | **Function** |
| Setting values | 0 | 10, 11 | X | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 1 | 1 | XXX | PWM mode waveform out |
| | 0 | 00 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | 0 | 00 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | 1 | | | | | |
| | Other than above | | | | | General I/O port |

[Legend]

X: Don't care.

- FTIOC1 pin

| Register Name | TRDOER1 | TRDFCR | | TRDPMR | TRDIORC | |
|---|---|---|---|---|---|---|
| **Bit Name** | **EC1** | **CMD1, CMD0** | **PWM3** | **PWMC1** | **IOC2 to IOC0** | **Function** |
| Setting values | 0 | 10, 11 | X | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 1 | 1 | XXX | PWM mode waveform out |
| | 0 | 00 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | 0 | 00 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | 1 | | | | | |
| | Other than above | | | | | General I/O port |

[Legend]

X: Don't care.

- FTIOD1 pin

| Register Name | TRDOER1 | TRDFCR | | TRDPMR | TRDIORC | |
|---|---|---|---|---|---|---|
| Bit Name | ED1 | CMD1, CMD0 | PWM3 | PWMD1 | IOD2 to IOD0 | Function |
| Setting values | 0 | 10, 11 | X | X | XXX | Complementary PWM mode waveform output |
| | 0 | 01 | X | X | XXX | Reset synchronous PWM mode waveform output |
| | 0 | 00 | 1 | 1 | XXX | PWM mode waveform out |
| | 0 | 00 | 1 | 0 | 001, 01X | Timer mode waveform output (output compare function) |
| | 0 | 00 | 1 | 0 | 1XX | Timer mode (input capture function) |
| | 1 | | | | | |
| | Other than above | | | | | General I/O port |

[Legend]

X: Don't care.

### 14.4.1 Counter Operation

When one of bits STR0 and STR1 in TRDSTR is set to 1, the TRDCNT counter for the corresponding channel begins counting. TRDCNT can operate as a free-running counter, periodic counter, for example. Figure 14.10 shows an example of the counter operation setting procedure.

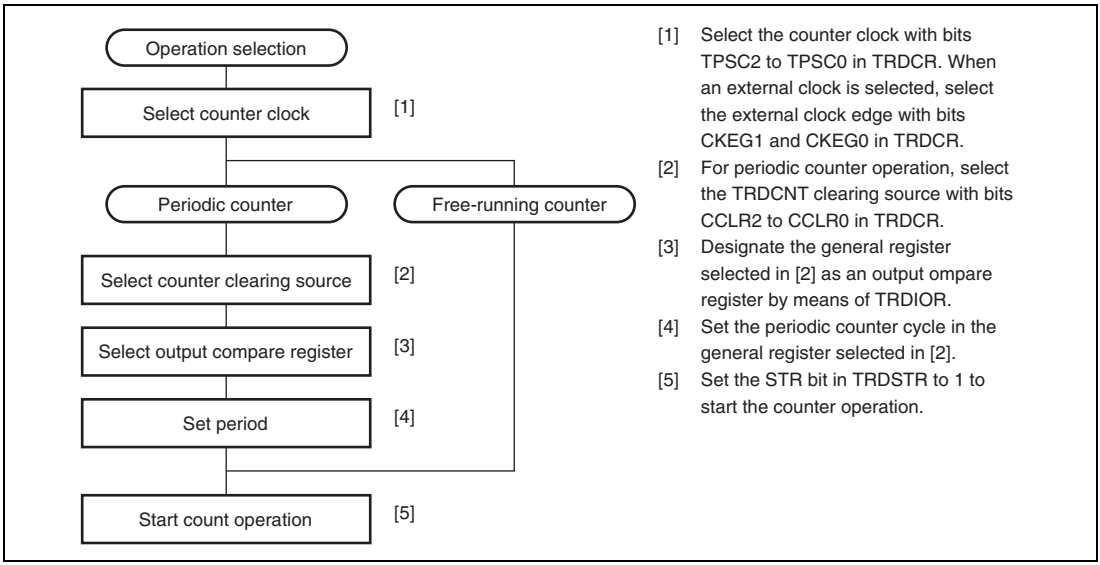



**Figure 14.10 Example of Counter Operation Setting Procedure**

### (1) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the TRDCNT counters for channels 0 and 1 are all designated as free-running counters. When the relevant bit in TRDSTR is set to 1, the corresponding TRDCNT counter starts an increment operation as a free-running counter. When TRDCNT overflows, the OVF flag in TRDSR is set to 1. If the value of the OVIE bit in the corresponding TRDIER is 1 at this point, timer RD requests an interrupt. After overflow, TRDCNT starts an increment operation again from H'0000.

Figure 14.11 illustrates free-running counter operation.

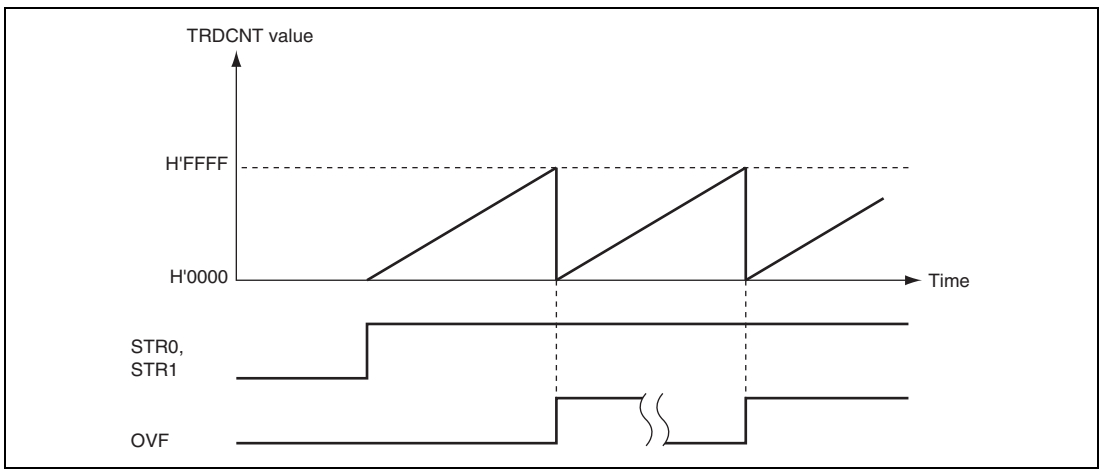



**Figure 14.11 Free-Running Counter Operation**

When compare match is selected as the TRDCNT clearing source, the TRDCNT counter for the relevant channel performs periodic count operation. The GR registers for setting the period are designated as output compare registers, and counter clearing by compare match is selected by means of bits CCLR1 and CCLR0 in TRDCR. After the settings have been made, TRDCNT starts an increment operation as a periodic counter when the corresponding bit in TRDSTR is set to 1. When the count value matches the value in GR, the IMFA, IMFB, IMFC, or IMFD flag in TRDSR is set to 1 and TRDCNT is cleared to H'0000.

If the value of the corresponding IMIEA, IMIEB, IMIEC, or IMIED bit in TRDIER is 1 at this point, timer RD requests an interrupt. After a compare match, TRDCNT starts an increment operation again from H'0000.

Figure 14.12 illustrates periodic count operation.

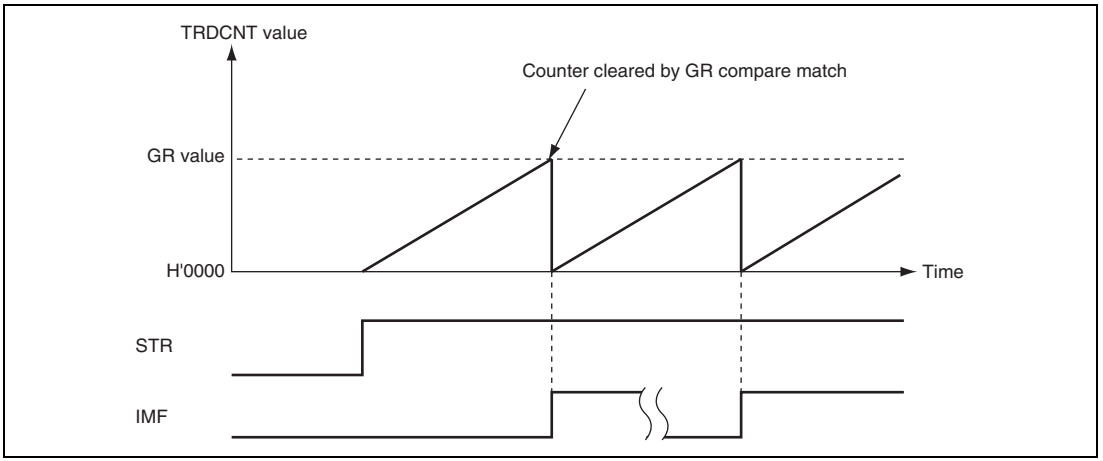



**Figure 14.12 Periodic Counter Operation**

### (2) TRDCNT Count Timing

- Internal clock operation

  A system clock (ϕ), four types of clocks (ϕ/2, ϕ/4, ϕ/8, or ϕ/32) that are generated by dividing the system clock, or on-chip oscillator clock (ϕ40M) can be selected by bits TPSC2 to TPSC0 in TRDCR.

  Figure 14.13 illustrates this timing.

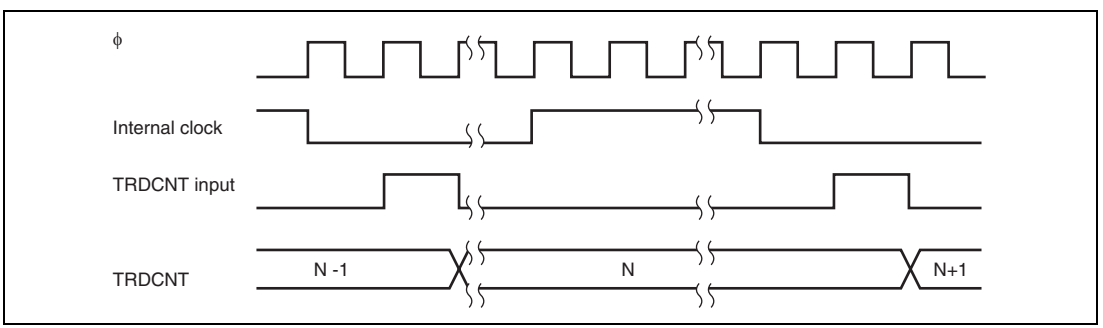



**Figure 14.13 Count Timing at Internal Clock Operation**

- External clock operation

  An external clock input pin (TCLK) can be selected by bits TPSC2 to TPSC0 in TRDCR, and a detection edge can be selected by bits CKEG1 and CKEG0. To detect an external clock, the rising edge, falling edge, or both edges can be selected.

  Figure 14.14 illustrates the detection timing of the rising and falling edges.

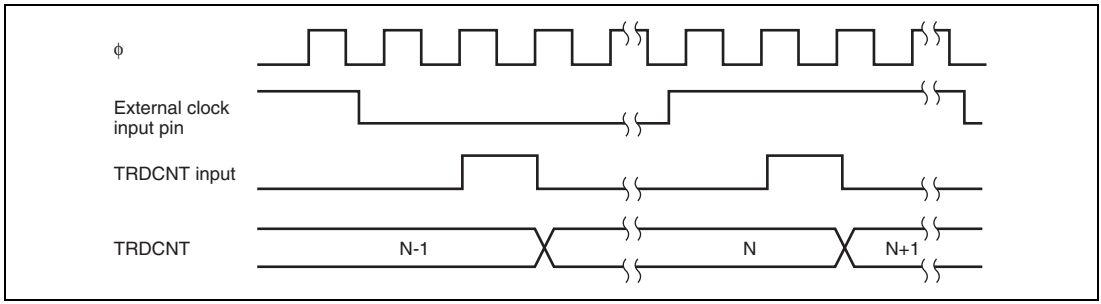



**Figure 14.14 Count Timing at External Clock Operation (Both Edges Detected)**

### 14.4.2 Waveform Output by Compare Match

Timer RD can perform 0, 1, or toggle output from the corresponding FTIOA, FTIOB, FTIOC, or FTIOD output pin using compare match A, B, C, or D.

Figure 14.15 shows an example of the setting procedure for waveform output by compare match.

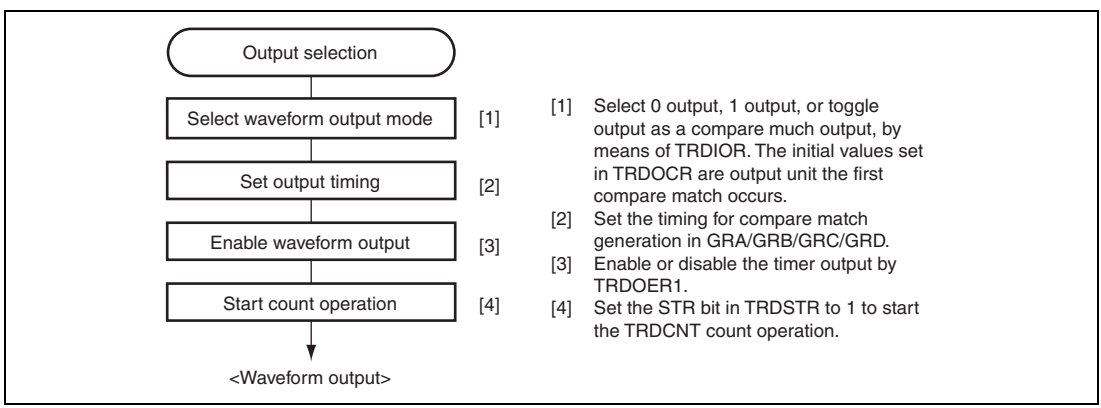



**Figure 14.15 Example of Setting Procedure for Waveform Output by Compare Match**

### (1) Examples of Waveform Output Operation

Figure 14.16 shows an example of 0 output/1 output.

In this example, TRDCNT has been designated as a free-running counter, and settings have been made such that 0 is output by compare match A, and 1 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

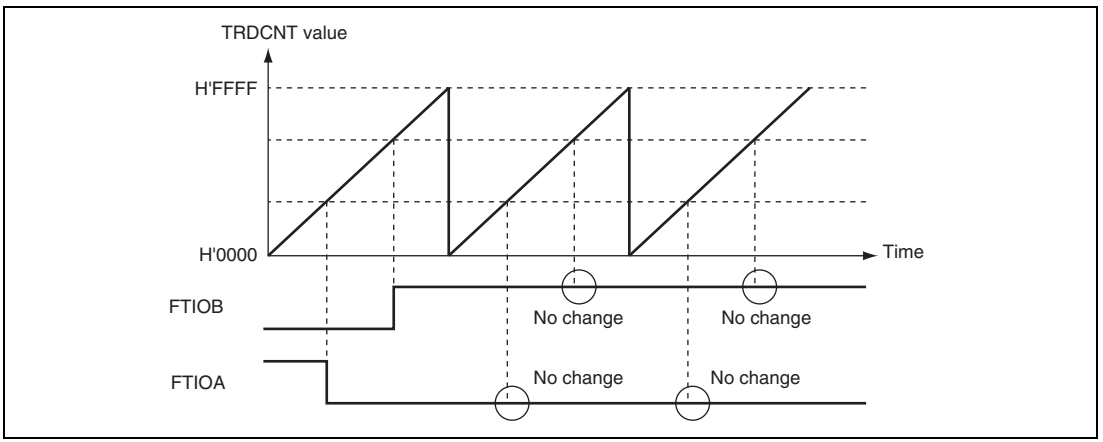



**Figure 14.16 Example of 0 Output/1 Output Operation**

Figure 14.17 shows an example of toggle output.

In this example, TRDCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made such that the output is toggled by both compare match A and compare match B.

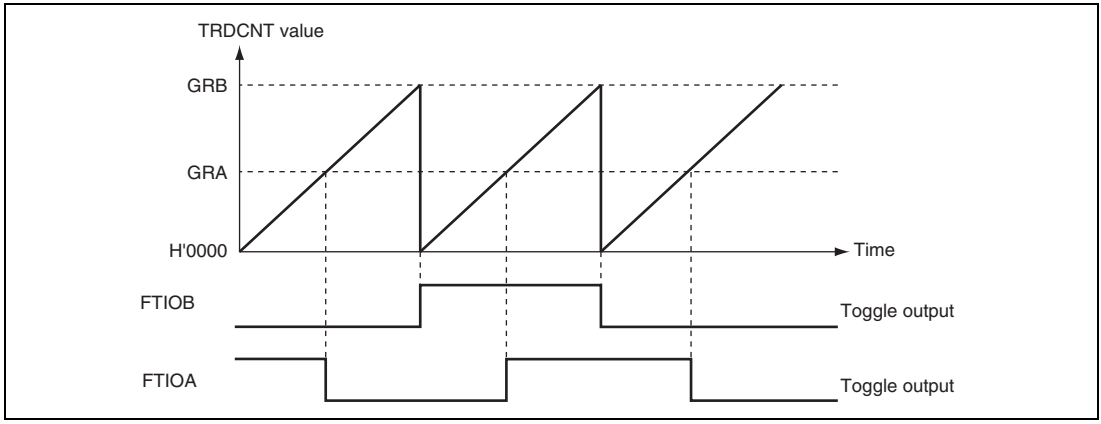



**Figure 14.17 Example of Toggle Output Operation**

### (2) Output Compare Timing

The compare match signal is generated in the last state in which TRDCNT and GR match (when TRDCNT changes from the matching value to the next value). When the compare match signal is generated, the output value selected in TRDIOR is output at the compare match output pin (FTIOA, FTIOB, FTIOC, or FTIOD). When TRDCNT matches GR, the compare match signal is generated only after the next TRDCNT input clock pulse is input.

Figure 14.18 shows an example of the output compare timing.

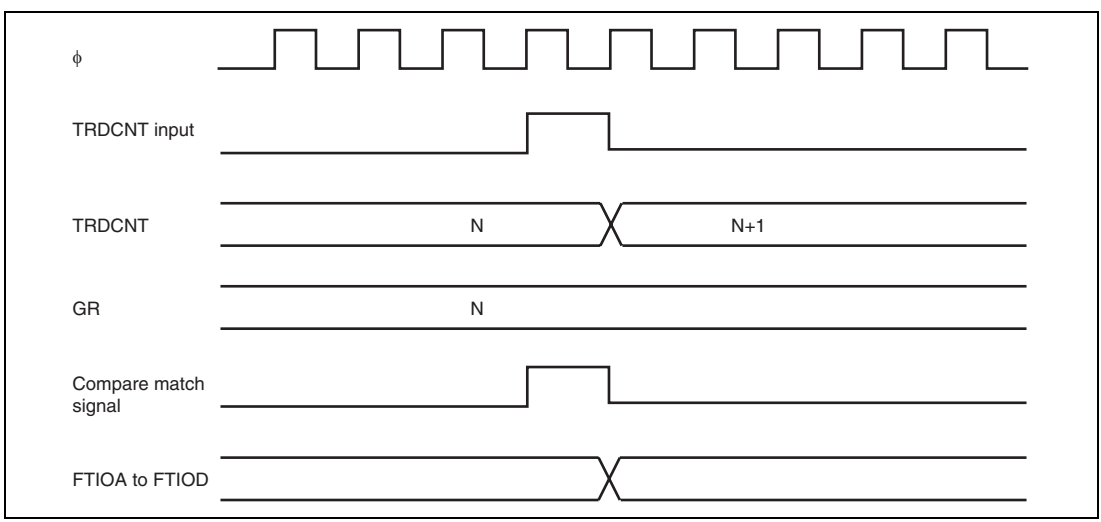



**Figure 14.18 Output Compare Timing**

### 14.4.3 Input Capture Function

The TRDCNT value can be transferred to GR on detection of the input edge of the input capture/output compare pin (FTIOA, FTIOB, FTIOC, or FTIOD). Rising edge, falling edge, or both edges can be selected as the detected edge. When the input capture function is used, the pulse width or period can be measured.

Figure 14.19 shows an example of the input capture operation setting procedure.

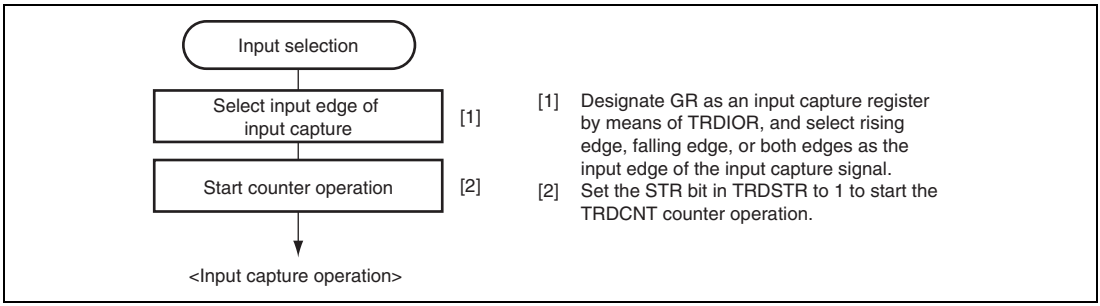



**Figure 14.19 Example of Input Capture Operation Setting Procedure**

### (1) Example of Input Capture Operation

Figure 14.20 shows an example of input capture operation.

In this example, both rising and falling edges have been selected as the FTIOA pin input capture input edge, the falling edge has been selected as the FTIOB pin input capture input edge, and counter clearing by GRB input capture has been designated for TRDCNT.

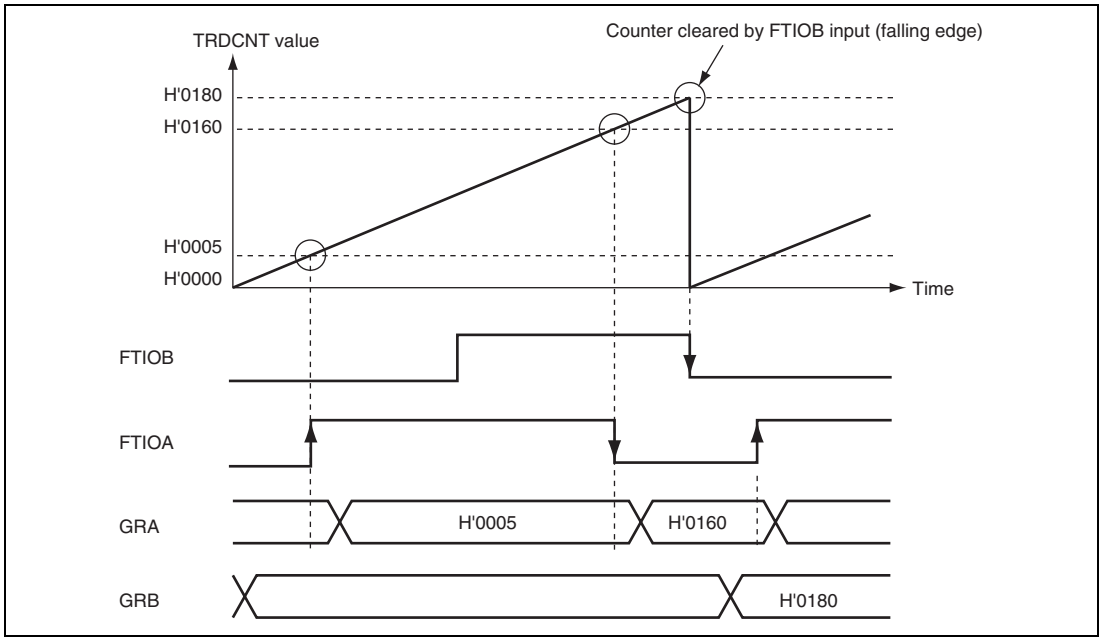



**Figure 14.20 Example of Input Capture Operation**

### (2) Input Capture Signal Timing

Input capture on the rising edge, falling edge, or both edges can be selected through settings in TRDIOR. Figure 14.21 shows the timing when the rising edge is selected.

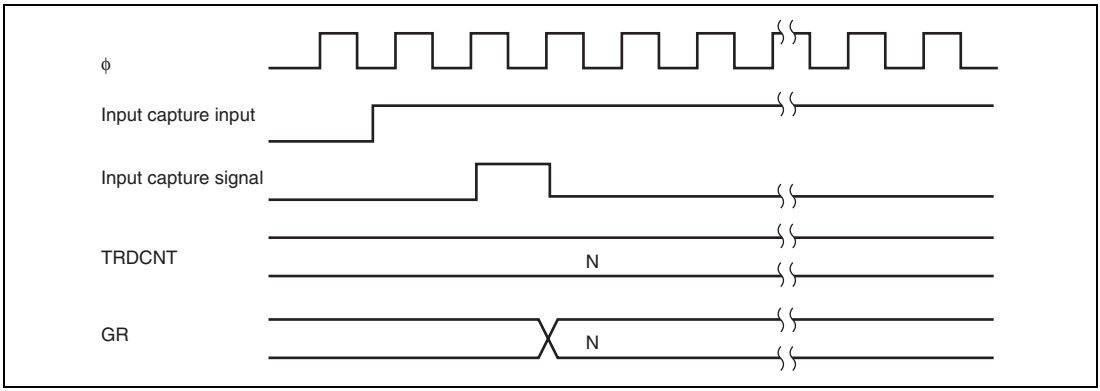
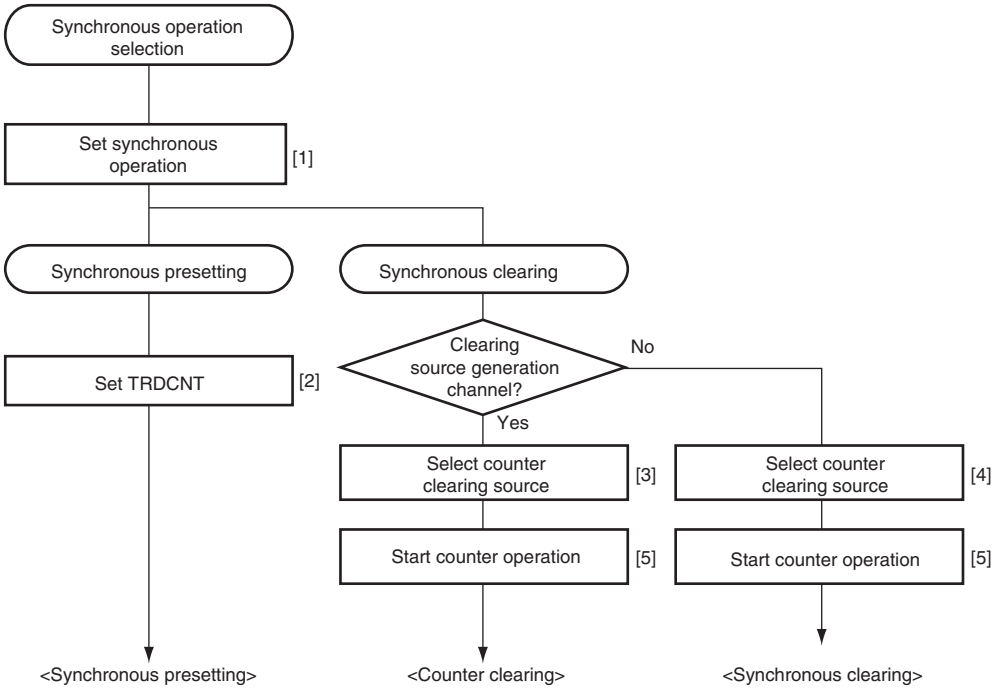



**Figure 14.21 Input Capture Signal Timing**

### 14.4.4 Synchronous Operation

In synchronous operation, the values in a number of TRDCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TRDCNT counters can be cleared simultaneously by making the appropriate setting in TRDCR (synchronous clearing). Synchronous operation enables GR to be increased with respect to a single time base.

Figure 14.22 shows an example of the synchronous operation setting procedure.




[1] Set the SYNC bits in TRDMDR to 1.
[2] When a value is written to either of the TRDCNT counters, the same value is simultaneously written to the other TRDCNT counter.
[3] Set bits CCLR1 and CCLR0 in TRDCR to specify counter clearing by compare match/input capture.
[4] Set bits CCLR1 and CCLR0 in TRDCR to designate synchronous clearing for the counter clearing source.
[5] Set the STR bit in TRDSTR to 1 to start the count operation.

**Figure 14.22 Example of Synchronous Operation Setting Procedure**

Figure 14.23 shows an example of synchronous operation. In this example, synchronous operation has been selected, FTIOB0 and FTIOB1 have been designated for PWM mode, GRA_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 counter clearing source. The same input clock has been set for the channel 0 and channel 1 counter input clocks. Two-phase PWM waveforms are output from pins FTIOB0 and FTIOB1. At this time, synchronous presetting and synchronous operation by GRA_0 compare match are performed by TRDCNT counters.

For details on PWM mode, see section 14.4.5, PWM Mode.

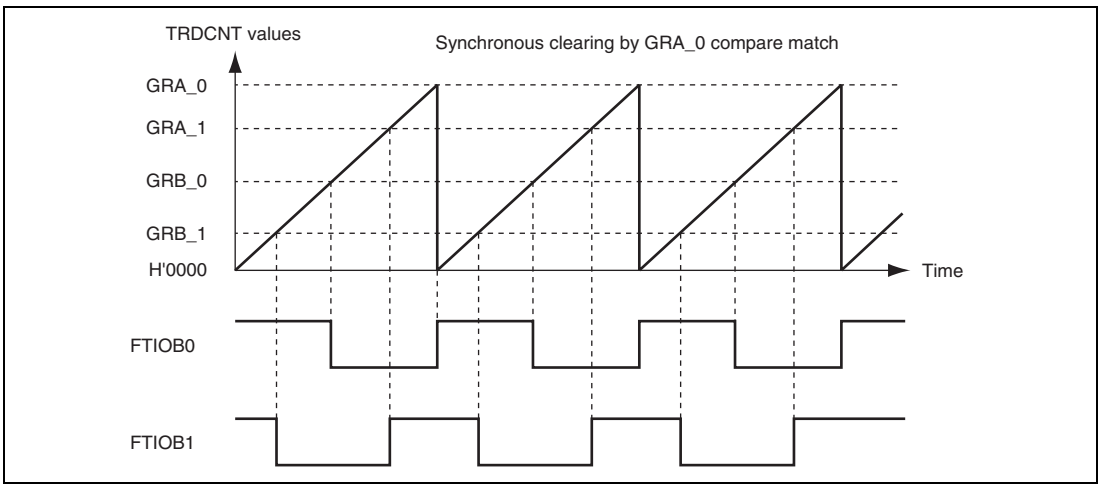



**Figure 14.23 Example of Synchronous Operation**

### 14.4.5 PWM Mode

In PWM mode, PWM waveforms are output from the FTIOB, FTIOC, and FTIOD output pins with GRA as a cycle register and GRB, GRC, and GRD as duty registers. The initial output level of the corresponding pin depends on the setting values of TRDOCR and POCR. Table 14.4 shows an example of the initial output level of the FTIOB0 pin.

The output level is determined by the POLB to POLD bits corresponding to POCR. When POLB is 0, the FTIOB output pin is set to 0 by compare match B and set to 1 by compare match A. When POLB is 1, the FTIOB output pin is set to 1 by compare match B and cleared to 0 by compare match A. In PWM mode, maximum 6-phase PWM outputs are possible.

Figure 14.24 shows an example of the PWM mode setting procedure.

**Table 14.4 Initial Output Level of FTIOB0 Pin**

| TOB0 | POLB | Initial Output Level |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

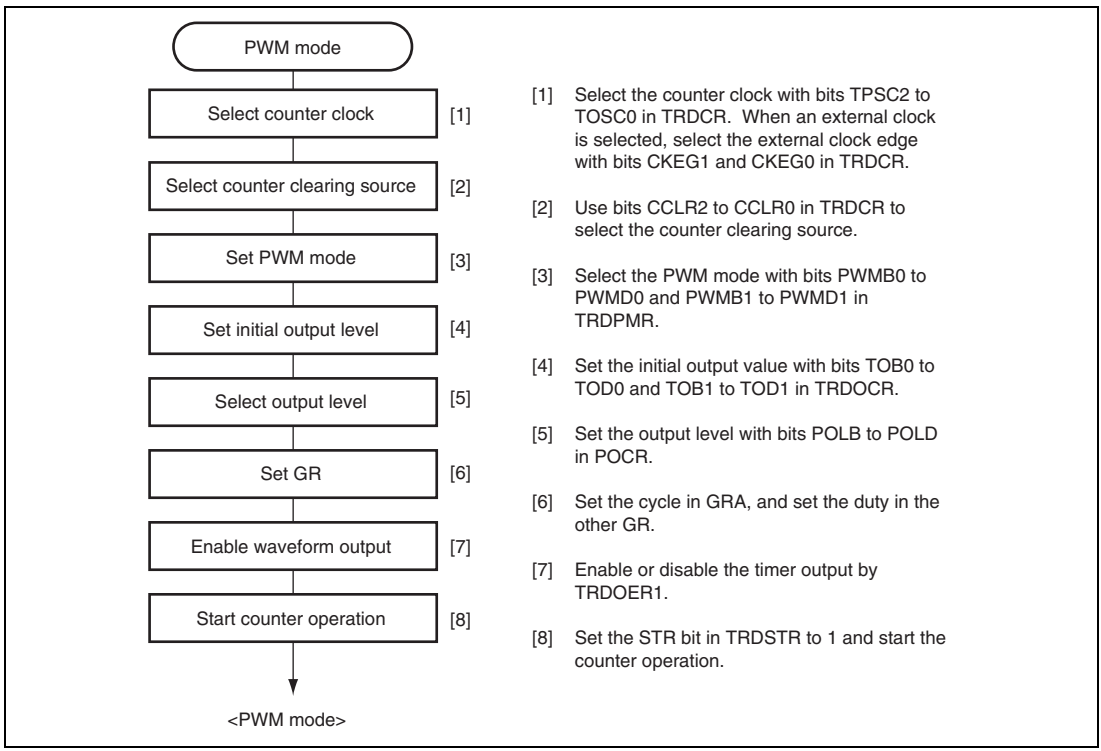



**Figure 14.24 Example of PWM Mode Setting Procedure**

Figure 14.25 shows an example of operation in PWM mode. The output signals go to 1 and TRDCNT is reset at compare match A, and the output signals go to 0 at compare match B, C, and D (TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 0).

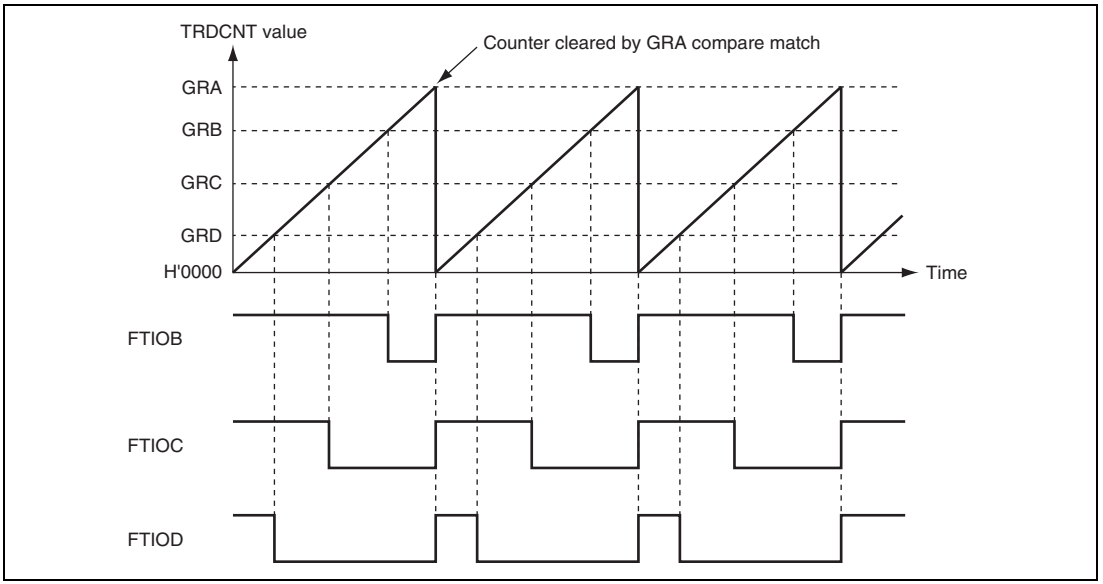



**Figure 14.25 Example of PWM Mode Operation (1)**

Figure 14.26 shows another example of operation in PWM mode. The output signals go to 0 and TRDCNT is reset at compare match A, and the output signals go to 1 at compare match B, C, and D (TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 1).

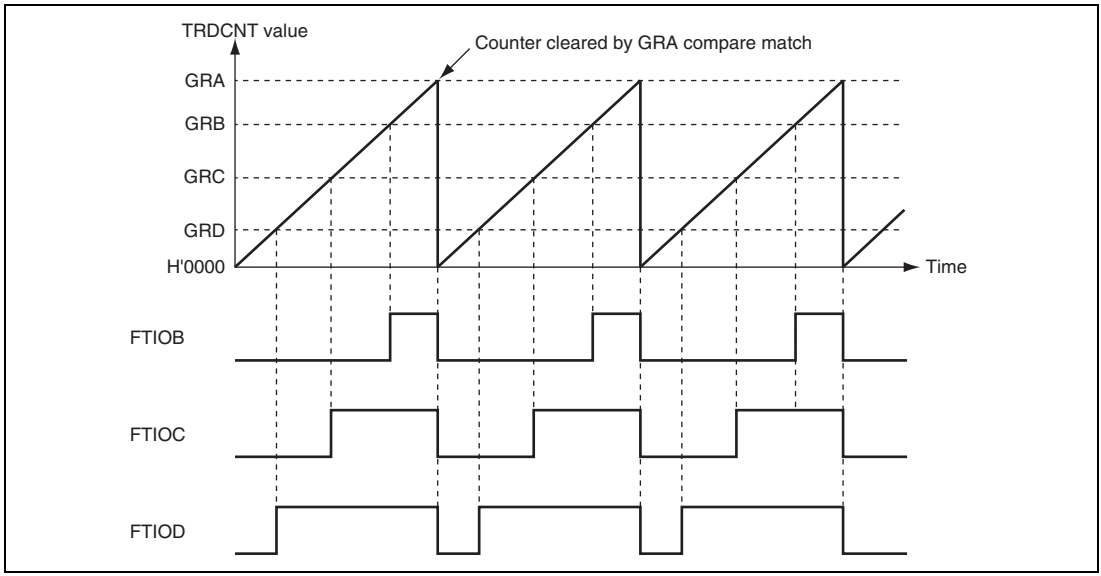



**Figure 14.26 Example of PWM Mode Operation (2)**

Figures 14.27 (when TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 0) and 14.28 (when TOB, TOC, and TOD = 0, POLB, POLC, and POLD = 1) show examples of the output of PWM waveforms with duty cycles of 0% and 100% in PWM mode.

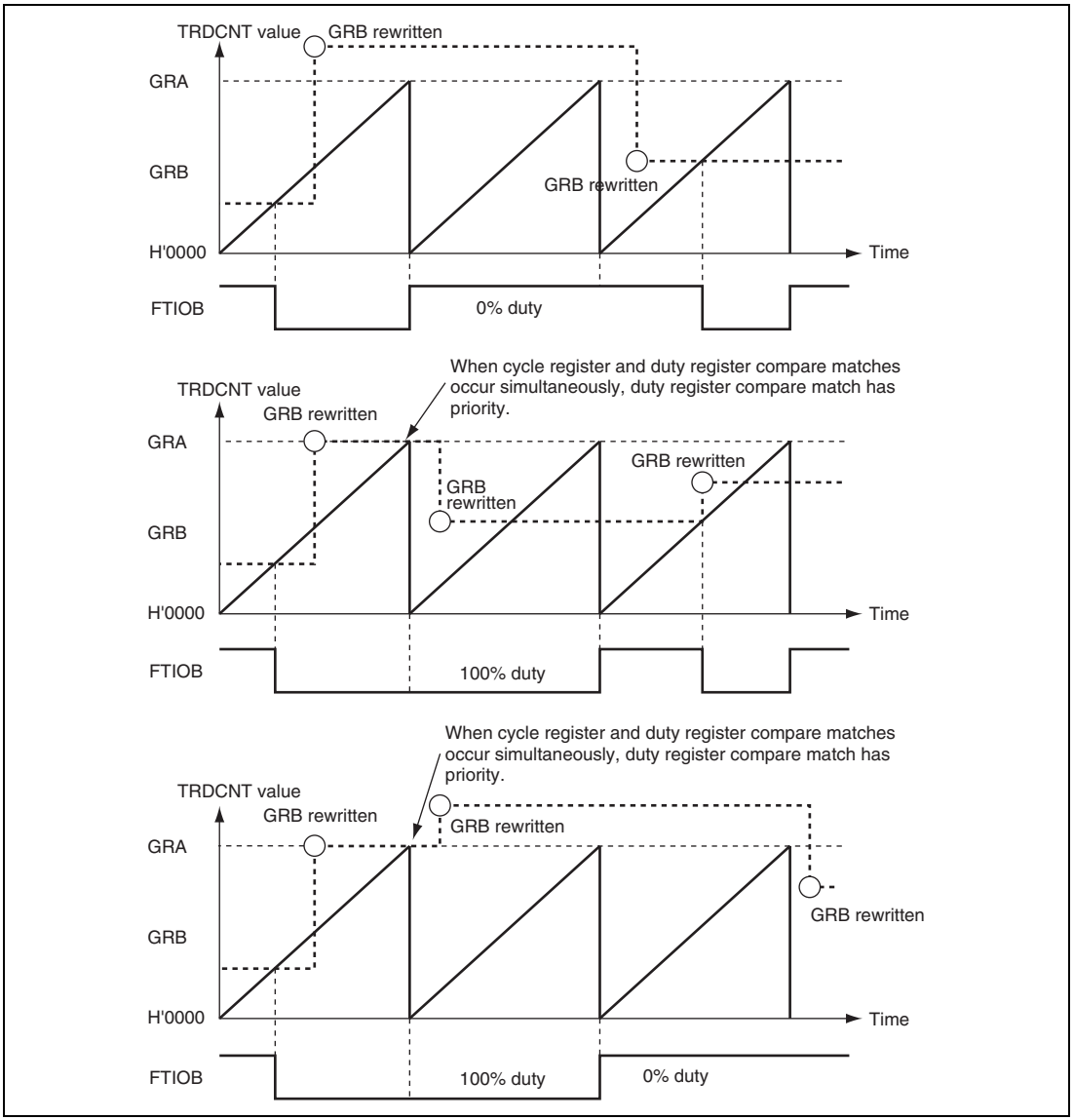



**Figure 14.27 Example of PWM Mode Operation (3)**

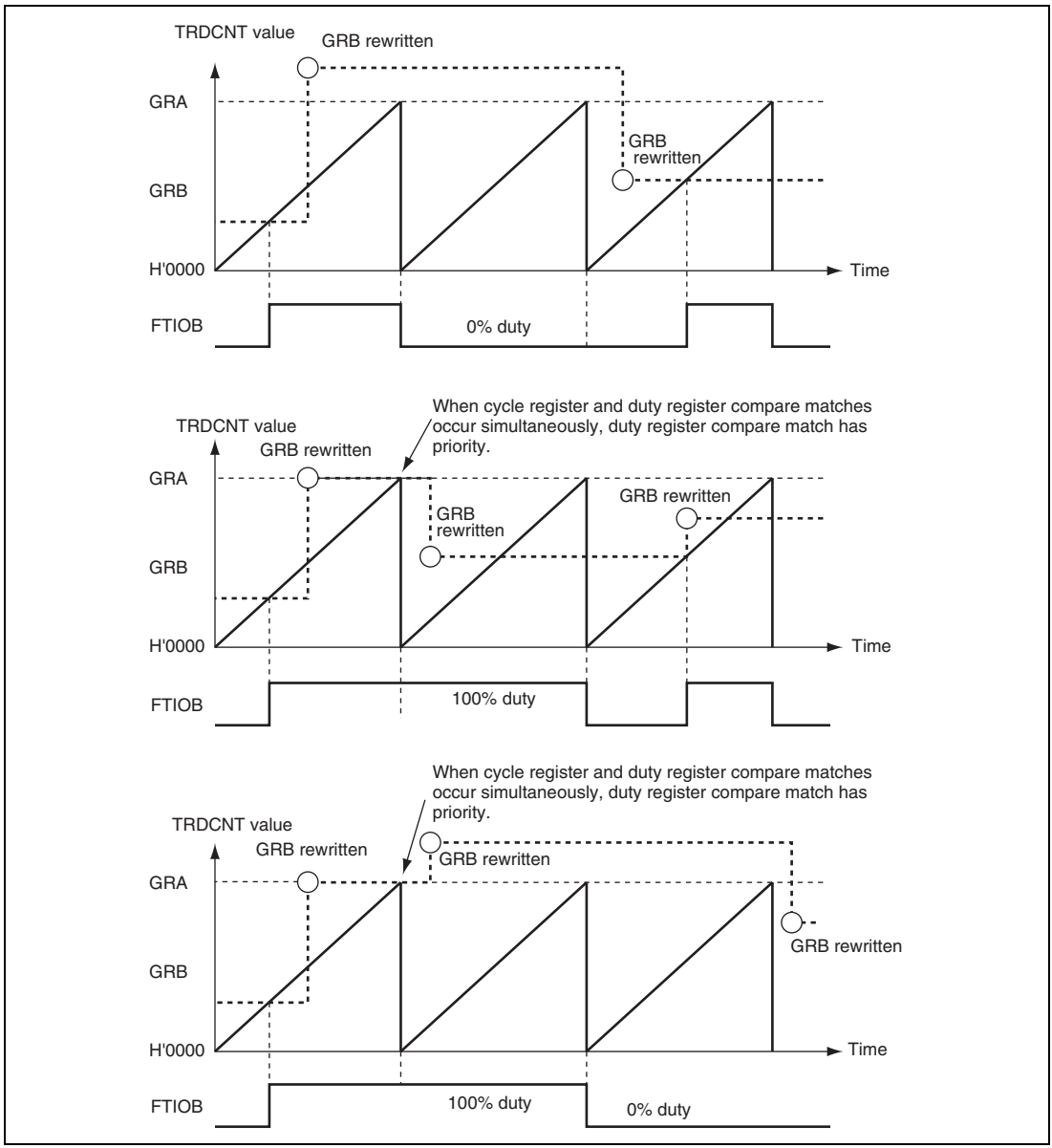



**Figure 14.28 Example of PWM Mode Operation (4)**

### 14.4.6 Reset Synchronous PWM Mode

Three normal- and counter-phase PWM waveforms are output by combining channels 0 and 1 that one of changing points of waveforms will be common.

In reset synchronous PWM mode, the FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 pins become PWM-output pins automatically. TRDCNT_0 performs an increment operation. Tables 14.5 and 14.6 show the PWM-output pins used and the register settings, respectively.

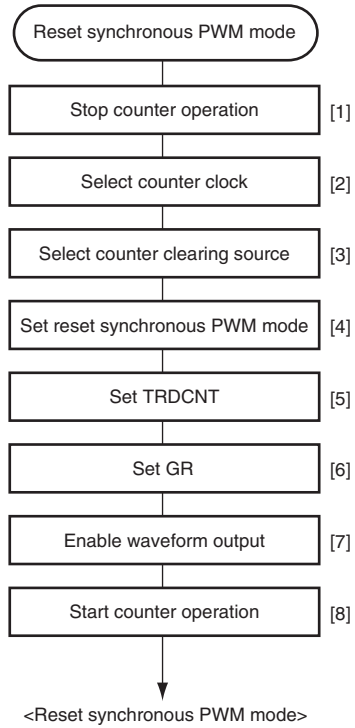
Figure 14.29 shows the example of reset synchronous PWM mode setting procedure.

**Table 14.5 Output Pins in Reset Synchronous PWM Mode**

| Channel | Pin Name | Input/Output | Pin Function |
|---|---|---|---|
| 0 | FTIOC0 | Output | Toggle output in synchronous with PWM cycle |
| 0 | FTIOB0 | Output | PWM output 1 |
| 0 | FTIOD0 | Output | PWM output 1 (counter-phase waveform of PWM output 1) |
| 1 | FTIOA1 | Output | PWM output 2 |
| 1 | FTIOC1 | Output | PWM output 2 (counter-phase waveform of PWM output 2) |
| 1 | FTIOB1 | Output | PWM output 3 |
| 1 | FTIOD1 | Output | PWM output 3 (counter-phase waveform of PWM output 3) |

**Table 14.6 Register Settings in Reset Synchronous PWM Mode**

| Register | Description |
|---|---|
| TRDCNT_0 | Initial setting of H'0000 |
| TRDCNT_1 | Not used (independently operates) |
| GRA_0 | Sets counter cycle of TRDCNT_0 |
| GRB_0 | Set a changing point of the PWM waveform output from pins FTIOB0 and FTIOD0. |
| GRA_1 | Set a changing point of the PWM waveform output from pins FTIOA1 and FTIOC1. |
| GRB_1 | Set a changing point of the PWM waveform output from pins FTIOB1 and FTIOD1. |




[1] Clear bit STR0 in TRDSTR to 0 and stop the counter operation of TRDCNT_0. Set reset synchronous PWM mode after TRDCNT_0 stops.

[2] Select the counter clock with bits TPSC2 to TPSC0 in TRDCR. When an external clock is selected, select the external clock edge with bits CKEG1 and CKEG0 in TRDCR.

[3] Use bits CCLR2 to CCLR0 in TRDCR to select counter clearing source GRA_0.

[4] Select the reset synchronous PWM mode with bits CMD1 and CMD0 in TRDFCR. FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 become PWM output pins automatically.

[5] Set TRDCNT_0 as H'0000. TRDCNT_1 does not need to be set.

[6] GRA_0 is a cycle register. Set a cycle for GRA_0. Set the changing point timing of the PWM output waveform for GRB_0, GRA_1, and GRB_1.

[7] Enable or disable the timer output by TRDOER1.

[8] Set the STR bit in TRDSTR to 1 and start the counter operation.

**Figure 14.29 Example of Reset Synchronous PWM Mode Setting Procedure**

Figures 14.30 and 14.31 show examples of operation in reset synchronous PWM mode.

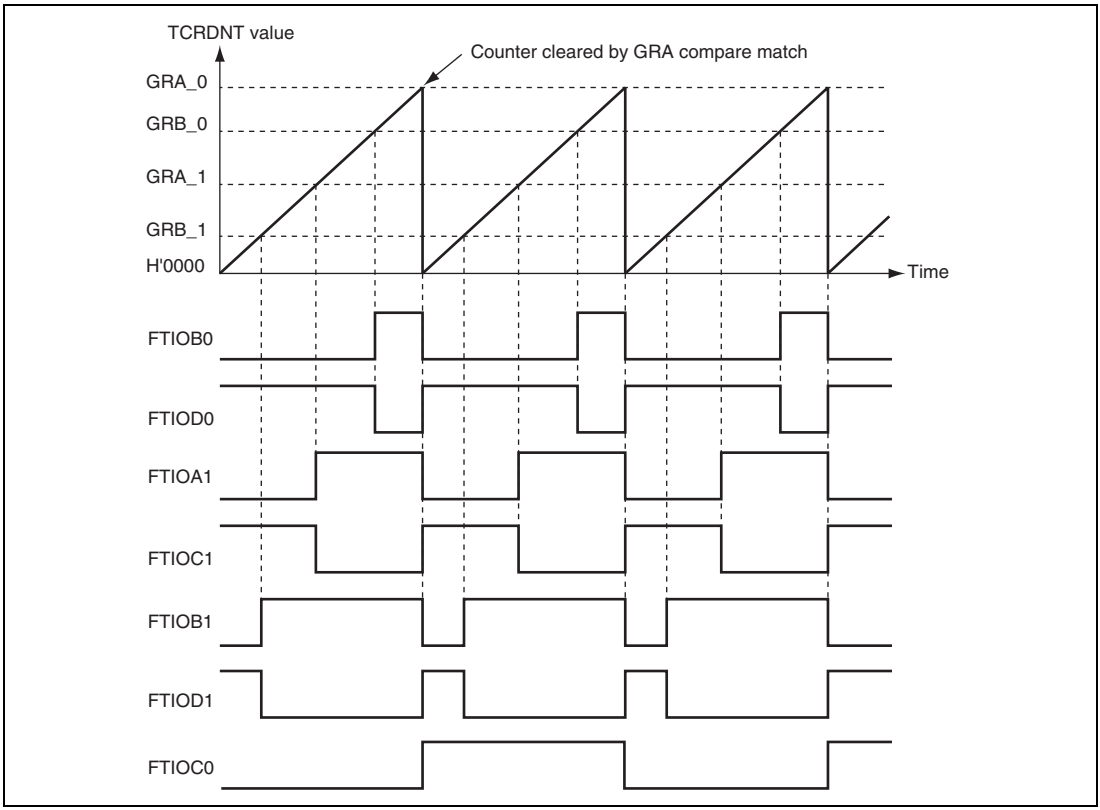



**Figure 14.30 Example of Reset Synchronous PWM Mode Operation (OLS0 = OLS1 = 1)**

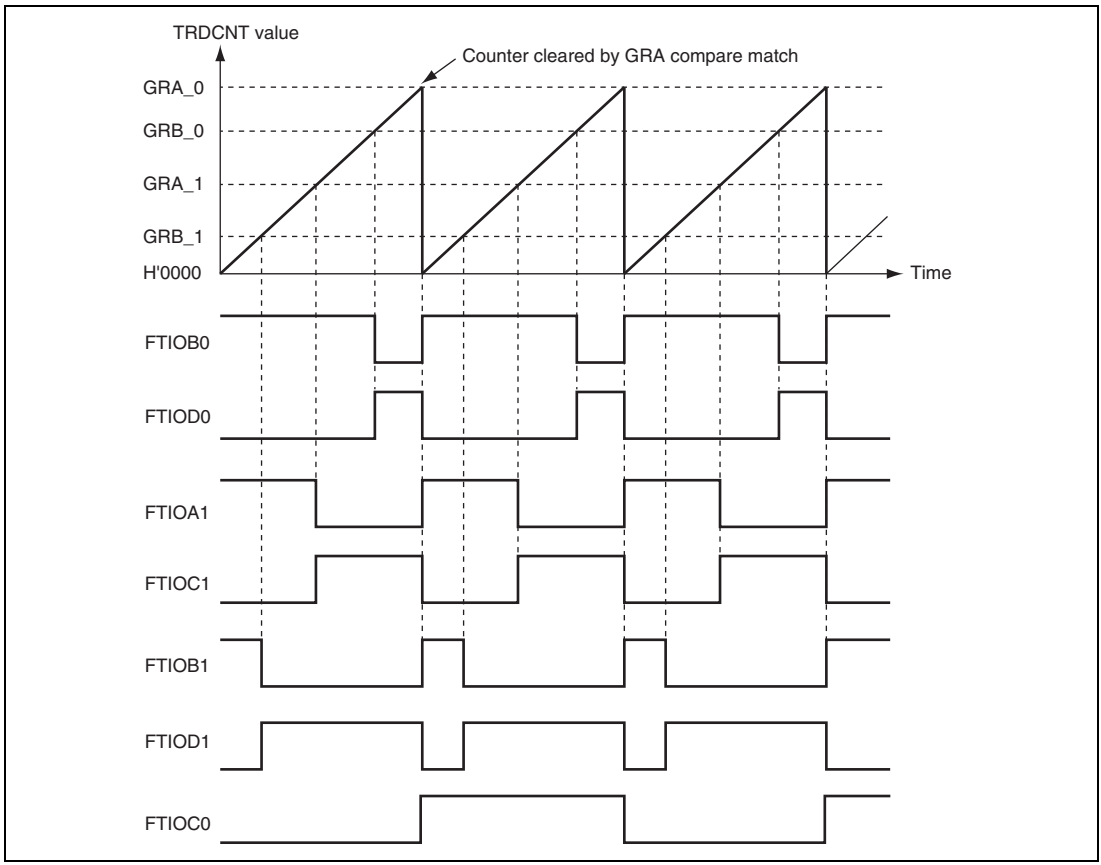



**Figure 14.31 Example of Reset Synchronous PWM Mode Operation (OLS0 = OLS1 = 0)**

In reset synchronous PWM mode, TRDCNT_0 and TRDCNT_1 perform increment and independent operations, respectively. However, GRA_1 and GRB_1 are separated from TRDCNT_1. When a compare match occurs between TRDCNT_0 and GRA_0, a counter is cleared and an increment operation is restarted from H'0000.

The PWM pin outputs 0 or 1 whenever a compare match between GRB_0, GRA_1, GRB_1 and TRDCNT_0 or counter clearing occur.

For details on operations when reset synchronous PWM mode and buffer operation are simultaneously set, refer to section 14.4.9, Buffer Operation.

### 14.4.7 Complementary PWM Mode

Three PWM waveforms for non-overlapped normal and counter phases are output by combining channels 0 and 1.

In complementary PWM mode, the FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 pins become PWM-output pins automatically. TRDCNT_0 and TRDCNT_1 perform an increment or decrement operation. Tables 14.7 and 14.8 show the output pins and register settings in complementary PWM mode, respectively.

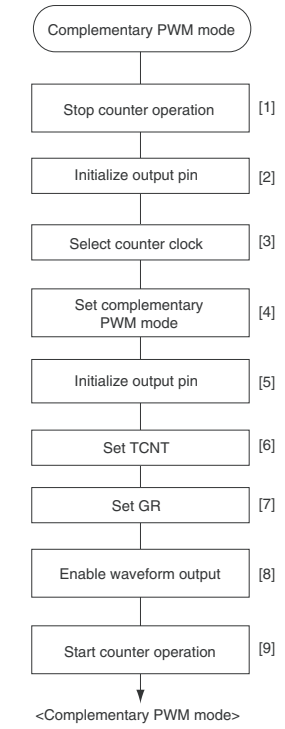
Figure 14.32 shows the example of complementary PWM mode setting procedure.

**Table 14.7 Output Pins in Complementary PWM Mode**

| Channel | Pin Name | Input/Output | Pin Function |
|---|---|---|---|
| 0 | FTIOC0 | Output | Toggle output in synchronous with PWM cycle |
| 0 | FTIOB0 | Output | PWM output 1 |
| 0 | FTIOD0 | Output | PWM output 1 (counter-phase waveform non-overlapped with PWM output 1) |
| 1 | FTIOA1 | Output | PWM output 2 |
| 1 | FTIOC1 | Output | PWM output 2 (counter-phase waveform non-overlapped with PWM output 2) |
| 1 | FTIOB1 | Output | PWM output 3 |
| 1 | FTIOD1 | Output | PWM output 3 (counter-phase waveform non-overlapped with PWM output 3) |

**Table 14.8 Register Settings in Complementary PWM Mode**

| Register | Description |
|---|---|
| TRDCNT_0 | Initial setting of non-overlapped periods (non-overlapped periods are differences with TRDCNT_1) |
| TRDCNT_1 | Initial setting of H'0000 |
| GRA_0 | Sets (upper limit value – 1) of TRDCNT_0 |
| GRB_0 | Set a changing point of the PWM waveform output from pins FTIOB0 and FTIOD0. |
| GRA_1 | Set a changing point of the PWM waveform output from pins FTIOA1 and FTIOC1. |
| GRB_1 | Set a changing point of the PWM waveform output from pins FTIOB1 and FTIOD1. |



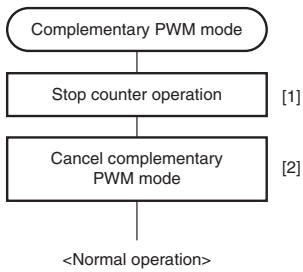

[1] Clear bits STR0 and STR1 in TSTR to 0, and stop the counter operation of TCNT_0. Stop TCNT_0 and TCNT_1 and set complementary PWM mode.
[2] Write H'00 to TRDOCR.
[3] Use bits TPSC2 to TPSC0 in TCR to select the same counter clock for channels 0 and 1. When an external clock is selected, select the edge of the external clock by bits CKEG1 and CKEG0 in TCR. Do not use bits CCLR1 and CCLR0 in TCR to clear the counter.
[4] Use bits CMD1 and CMD0 in TFCR to set complementary PWM mode. FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 automatically become PWM output pins.
[5] Set H'00 to TRDOCR.
[6] TCNT_1 must be H'0000. Set a non-overlapped period to TCNT_0.
[7] GRA_0 is a cycle register. Set the cycle to GRA_0. Set the timing to change the PWM output waveform to GRB_0, GRA_1, and GRB_1. Note that the timing must be set within the range of compare match carried out for TCNT_0 and TCNT_1. For GR settings, see 3. Setting GR Value in Complementary PWM Mode in section 13.4.7.
[8] Use TOER to enable or disable the timer output.
[9] Set the STR0 and STR1 bits in TSTR to 1 to start the count operation.

Note: To re-enter complementary PWM mode, first, enter a mode other than the complementary PWM mode. After that, repeat the setting procedures from step [1].
For settings of waveform outputs with a duty cycle of 0% and 100%, see the settings shown in 2. Examples of Complementary PWM Mode Operation and 3. Setting GR Value in Complementary PWM Mode in section 14.4.7.

**Figure 14.32 Example of Complementary PWM Mode Setting Procedure**

### (1) Canceling Procedure of Complementary PWM Mode

Figure 14.33 shows the complementary PWM mode canceling procedure.




[1] Clear bit CMD1 in TRDFCR to 0, and set channels 0 and 1 to normal operation.
[2] After setting channels 0 and 1 to normal operation, clear bits STR0 and STR1 in TRDSTR to 0 and stop TRDCNT_0 and TRDCNT_1.

**Figure 14.33 Canceling Procedure of Complementary PWM Mode**

### (2) Examples of Complementary PWM Mode Operation

Figure 14.34 shows an example of complementary PWM mode operation. In complementary PWM mode, TRDCNT_0 and TRDCNT_1 perform an increment or decrement operation. When TRDCNT_0 and GRA_0 are compared and their contents match, the counter is decremented, and when TRDCNT_1 underflows, the counter is incremented. In GRA_0, GRA_1, and GRB_1, compare match is carried out in the order of TRDCNT_0 → TRDCNT_1 → TRDCNT_1 → TRDCNT_0 and PWM waveform is output, during one cycle of a up/down counter. In this mode, the initial setting will be TRDCNT_0 > TRDCNT_1.

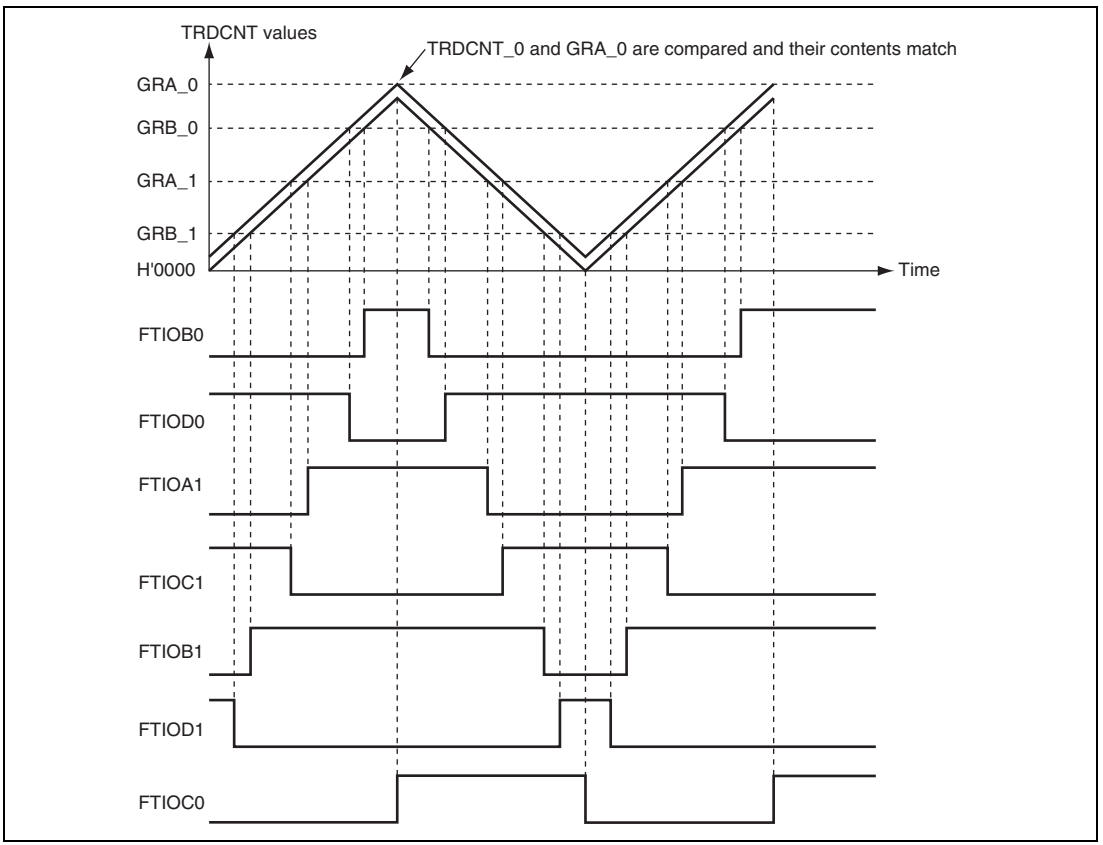



**Figure 14.34 Example of Complementary PWM Mode Operation (1)**

Figure 14.35 shows an example of PWM waveform output with 0% duty and 100% duty in complementary PWM mode (for one phase).

In this figure, GRB_0 is set to a value equal to or greater than GRA_0 and H'0000. The waveform with a duty cycle of 0% and 100% can be output. When buffer operation is used together, the duty cycles can easily be changed, including the above settings, during operation. For details on buffer operation, refer to section 14.4.9, Buffer Operation.

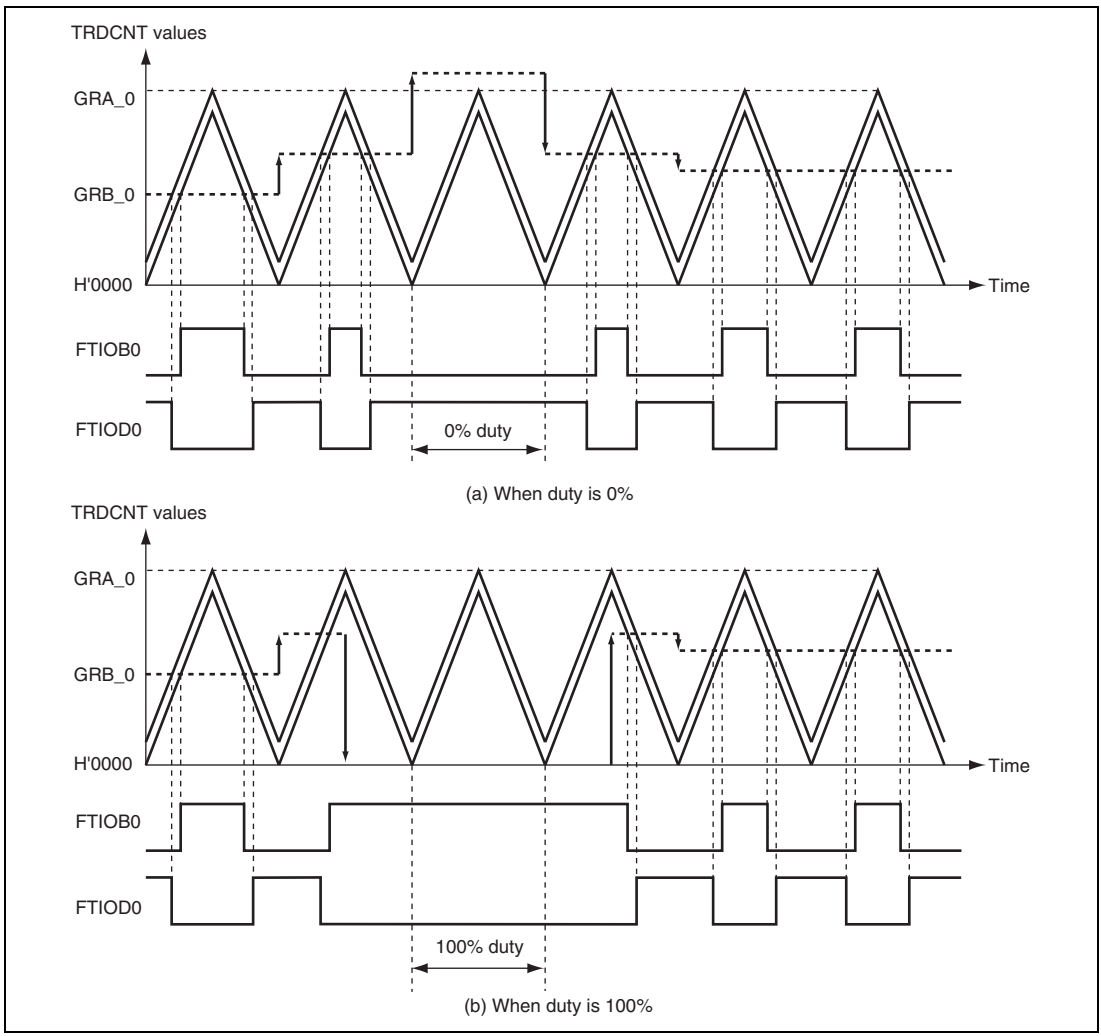



**Figure 14.35 Example of Complementary PWM Mode Operation (2)**

In complementary PWM mode, when the counter switches from up-counter to down-counter or vice versa, TRDCNT_0 and TRDCNT_1 overshoots or undershoots, respectively. In this case, the conditions to set the IMFA flag in channel 0 and the UDF flag in channel 1 differ from usual settings. Also, the transfer conditions in buffer operation differ from usual settings. Such timings are shown in figures 14.36 and 14.37.

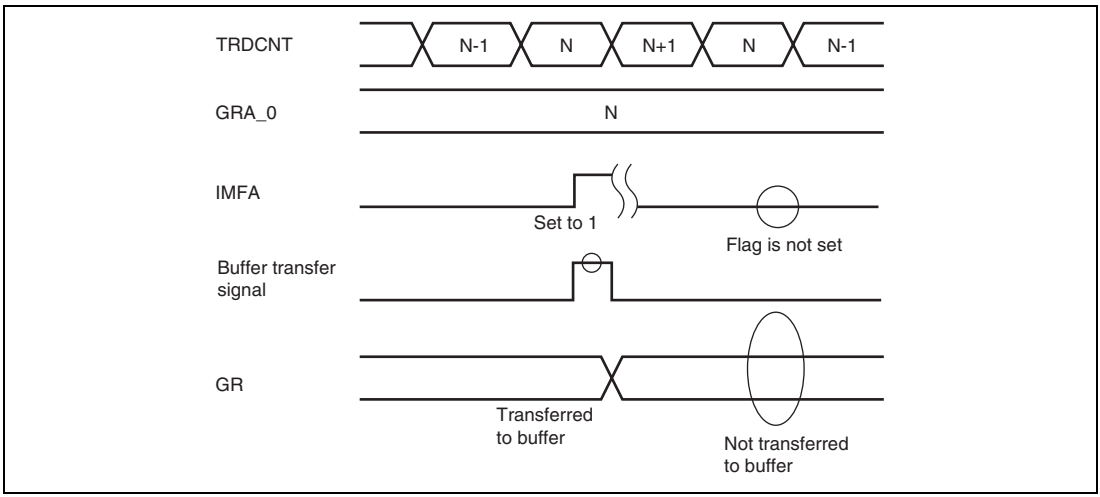



**Figure 14.36 Timing of Overshooting**

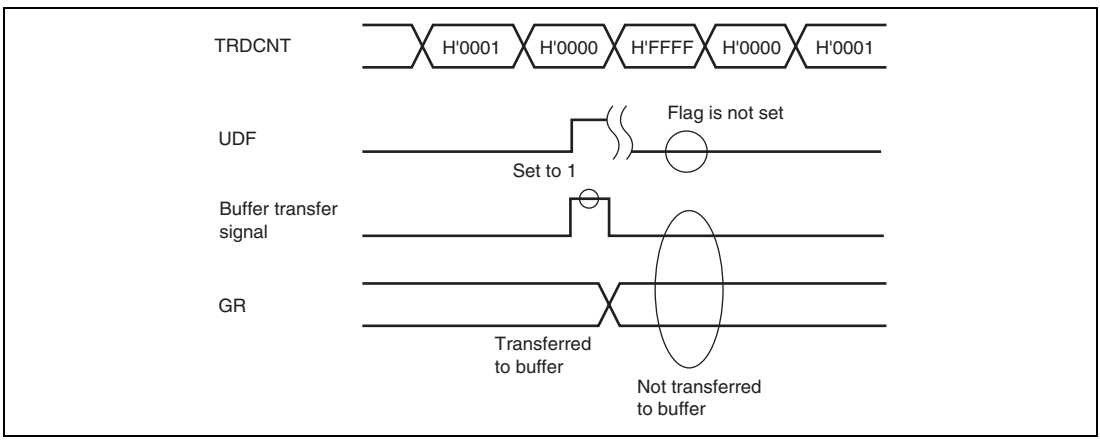



**Figure 14.37 Timing of Undershooting**

When the counter is incremented or decremented, the IMFA flag of channel 0 is set to 1, and when the register is underflowed, the UDF flag of channel 0 is set to 1. After buffer operation has been designated for GR, the value in the buffer registers is transferred to GR when the counter is incremented by compare match A0 or when TRDCNT_1 is underflowed. If the $\phi$ or $\phi/2$ clock is selected by TPSC2 to TPSC0 bits, the OVF flag is not set to 1 at the timing that the counter value changes from H'FFFF to H'0000.

### (3) Setting GR Value in Complementary PWM Mode

To set the general register (GR) or modify GR during operation in complementary PWM mode, refer to the following notes.

1. Initial value
   - H'0000 to T – 1 (T: Initial value of TRDCNT_0) must not be set for the initial value.
   - GRA_0 – (T – 1) or more must not be set for the initial value.
   - When using buffer operation, the same values must be set in the buffer registers and corresponding general registers.
2. Modifying the setting value
   - Use the buffer operation to change the GR value. If the GR value is changed by writing to it directly, the intended waveform may not be output.
   - Do not change settings of GRA_0 during operation.

### 14.4.8 PWM3 Mode Operation

In PWM3 mode, single-phase PWM waveforms can be output using TRDCNT_0. The waveform does not overlap its counter-phase waveform.

When the PWM3 mode is selected, the FTIOA0 and FTIOB0 pins are automatically set to output pins for the PWM function using TRDCNT_0 regardless of the TRDPMR value. The waveform is output on a GRA_0, GRA_1, GRB_0, or GRB_1 compare match according to bits TOA0 and TOB0 in TRDOCR.

- When TOA0 = 0, 1 is output on a compare match of GRA_1 and 0 is output on a compare match of GRA_0 on the FTIOA0 pin.
- When TOA0 = 1, 0 is output on a compare match of GRA_1 and 1 is output on a compare match of GRA_0 on the FTIOA0 pin.
- When TOB0 = 0, 1 is output on a compare match of GRB_1 and 0 is output on a compare match of GRB_0 on the FTIOB0 pin.
- When TOB0 = 1, 0 is output on a compare match of GRB_1 and 1 is output on a compare match of GRB_0 on the FTIOB0 pin.

Table 14.9 lists the correspondence between pin functions and GR registers, figure 14.38 shows a block diagram in PWM3 mode, and figure 14.39 shows a flowchart of setting in PWM3 mode.

When the buffer operation is used, set TRDMDR. The timer input/output pins, which are not used in PWM3 mode, can be used as general port pins. When the buffer operation is not set, since GRC or GRD is not used, a compare match interrupt can be generated when GRC or GRD matches with TRDCNT_1.

**Table 14.9 Pin Configuration in PWM3 Mode and GR Registers**

| Channel | Pin Name | Input/Output | Compare Match Register | Buffer Register |
|---|---|---|---|---|
| 0 | FTIOA0 | Output | GRA_0 | GRC_0 |
| | | | GRA_1 | GRC_1 |
| | FTIOB0 | | GRB_0 | GRD_0 |
| | | | GRB_1 | GRD_1 |
| | FTIOC0 | I/O | General I/O port | General I/O port |
| | FTIOD0 | | | |
| 1 | FTIOA1 | | | |
| | FTIOB1 | | | |
| | FTIOC1 | | | |
| | FTIOD1 | | | |

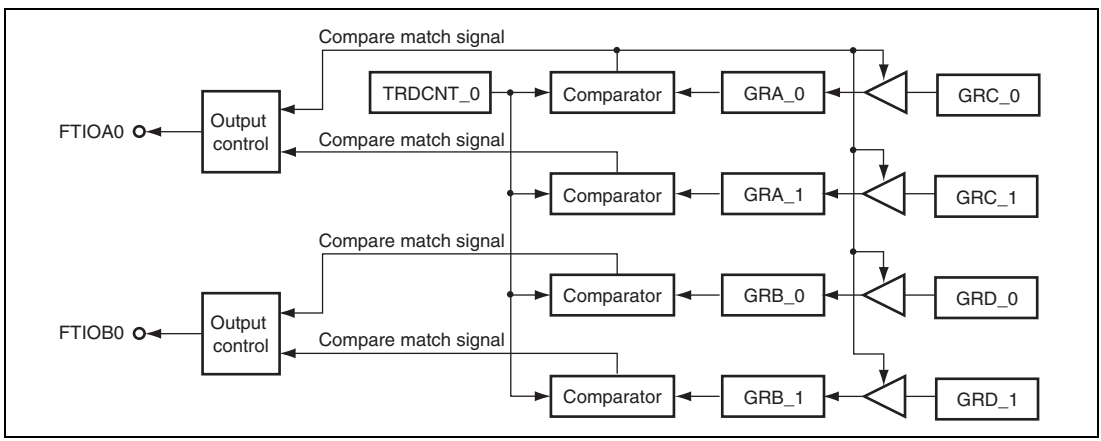



**Figure 14.38 Block Diagram in PWM3 Mode**

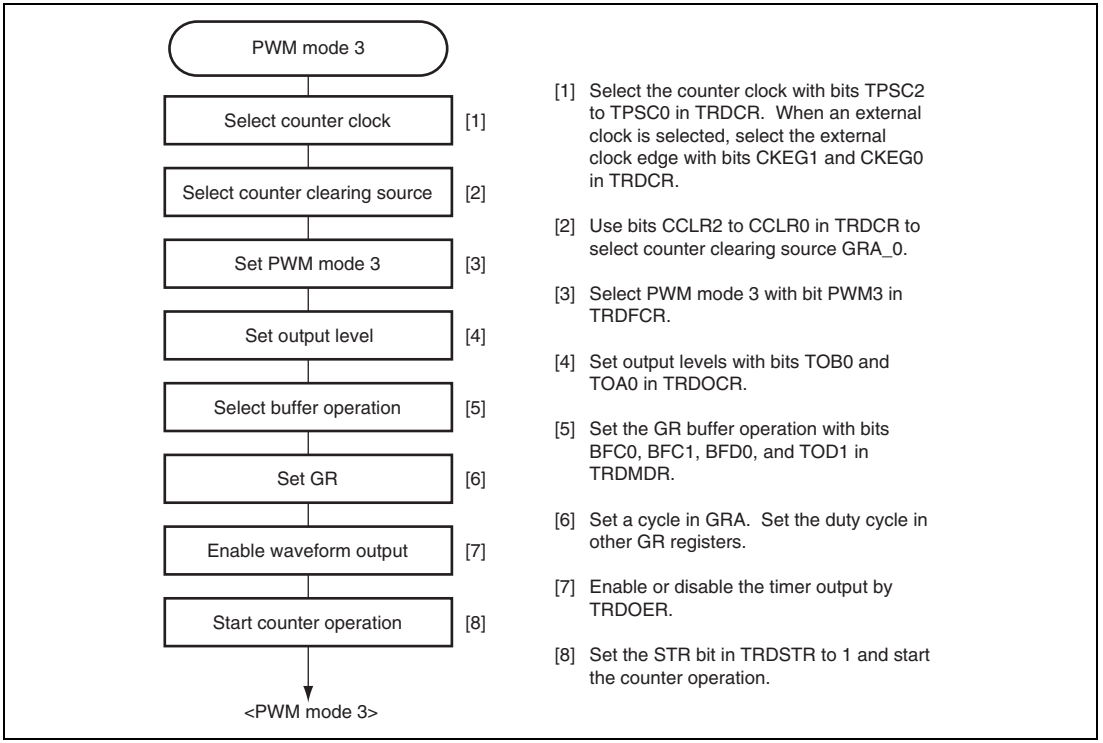



**Figure 14.39 Flowchart of Setting in PWM3 Mode**

Figure 14.40 is an example when non-overlapped pulses are output on pins FTIOA0 and FTIOB0. In this example, TRDCNT_0 functions as a periodic counter which is cleared on compare match A0 (bits CCLR2 to CCLR0 in TRDCR_0 are set to B'001), and PWM3 mode is selected (bit PWM3 in TRDFCR is cleared to 0). The cycle of the pulse is arbitrary.

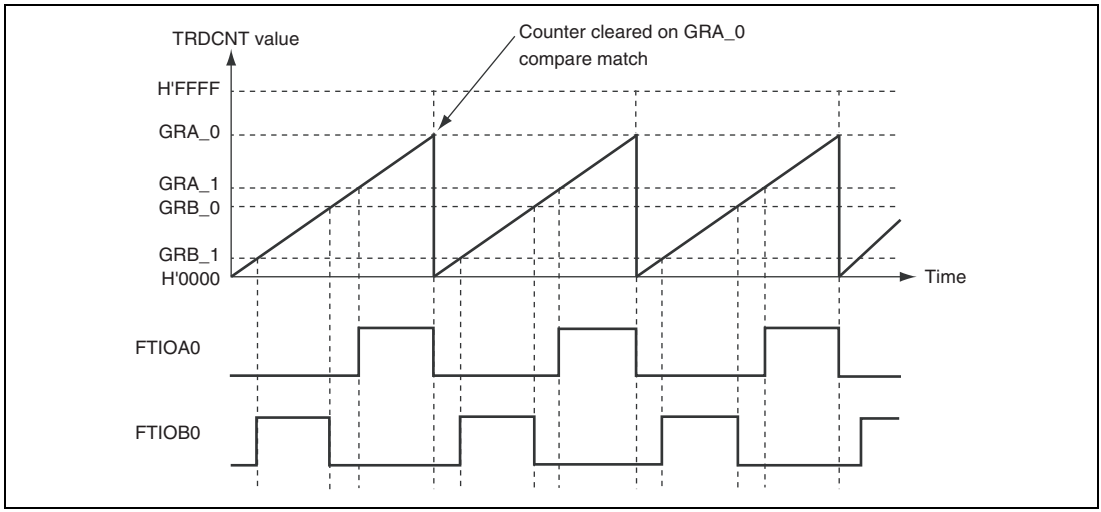



**Figure 14.40 Example of Non-Overlap Pulses**

### 14.4.9 Buffer Operation

Buffer operation differs depending on whether GR has been designated for an input capture register or an output compare register, or in reset synchronous PWM mode or complementary PWM mode.

Table 14.10 shows the register combinations used in buffer operation.

**Table 14.10 Register Combinations in Buffer Operation**

| General Register (GR) | Buffer Register |
|---|---|
| GRA | GRC |
| GRB | GRD |

#### (1) When GR is an Output Compare Register

When a compare match occurs, the value in GR of the corresponding channel is transferred to the general register.

This operation is illustrated in figure 14.41.

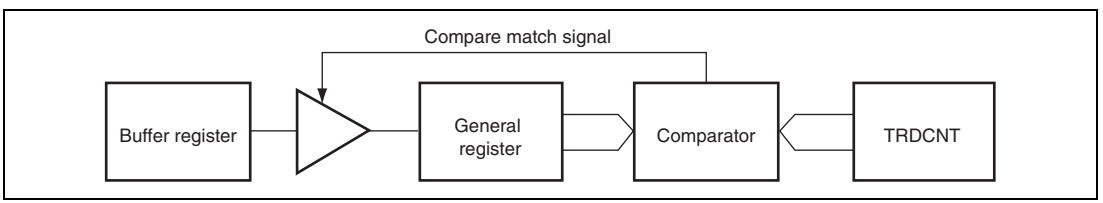



**Figure 14.41 Compare Match Buffer Operation**

### (2) When GR is an Input Capture Register

When an input capture occurs, the value in TRDCNT is transferred to GR and the value previously stored in the general register is transferred to the buffer register.

This operation is illustrated in figure 14.42.

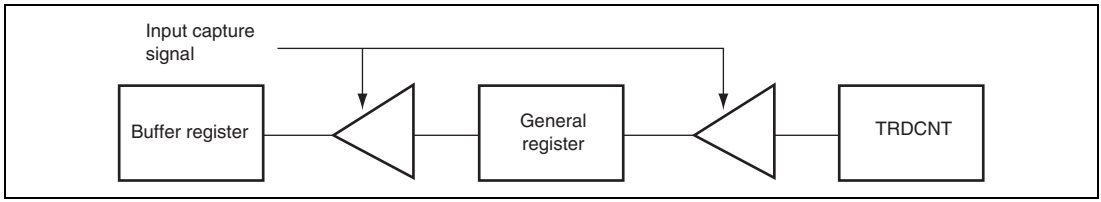



**Figure 14.42 Input Capture Buffer Operation**

### (3) PWM3 Mode

When compare match A0 occurs, the value of the buffer register is transferred to GR.

### (4) Complementary PWM Mode

When the counter switches from counting up to counting down or vice versa, the value of the buffer register is transferred to GR. Here, the value of the buffer register is transferred to GR in the following timing:

- When TRDCNT_0 and GRA_0 are compared and their contents match
- When TRDCNT_1 underflows

### (5) Reset Synchronous PWM Mode

When compare match A0 occurs, the value in the buffer register is transferred to GR.

### (6) Example of Buffer Operation Setting Procedure

Figure 14.43 shows an example of the buffer operation setting procedure.

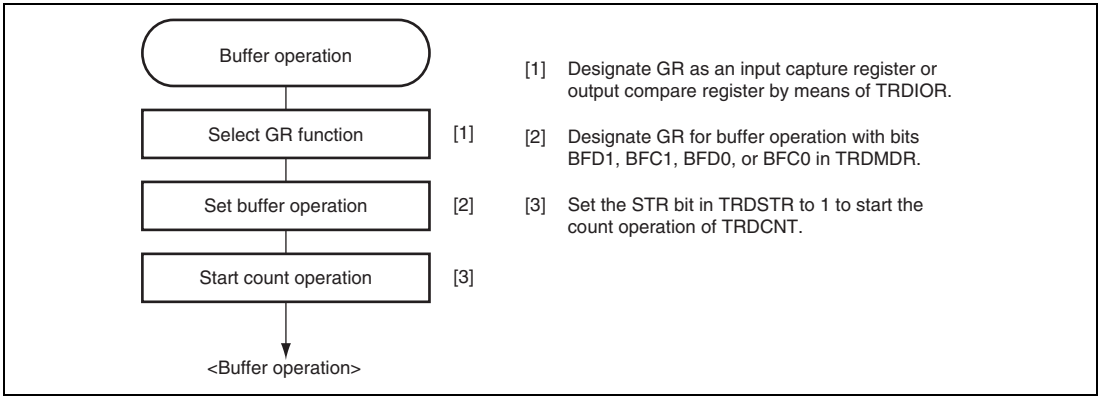



**Figure 14.43 Example of Buffer Operation Setting Procedure**

### (7) Examples of Buffer Operation

Figure 14.44 shows an operation example in which GRA has been designated as an output compare register, and buffer operation has been designated for GRA and GRC.

This is an example of TRDCNT operating as a periodic counter cleared by compare match B.

Pins FTIOA and FTIOB are set for toggle output by compare match A and B.

As buffer operation has been set, when compare match A occurs, the FTIOA pin performs toggle outputs and the value in buffer register is simultaneously transferred to the general register. This operation is repeated each time that compare match A occurs.

The timing to transfer data is shown in figure 14.45.

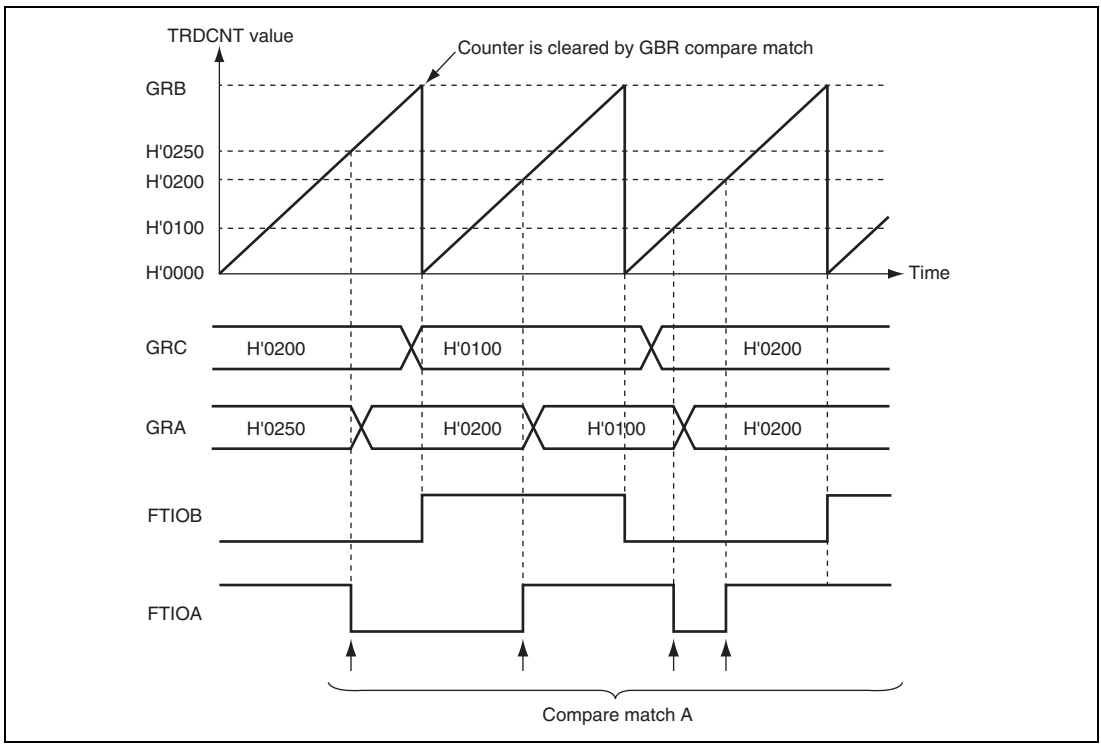



**Figure 14.44 Example of Buffer Operation (1)**
**(Buffer Operation for Output Compare Register)**

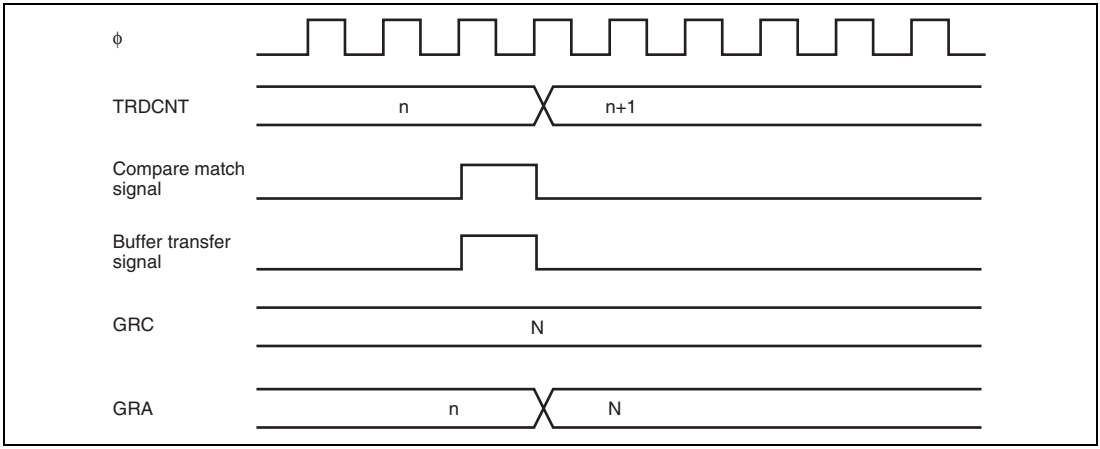



**Figure 14.45 Example of Compare Match Timing for Buffer Operation**

Figure 14.46 shows an operation example in which GRA has been designated as an input capture register, and buffer operation has been designated for GRA and GRC.

Counter clearing by input capture B has been set for TRDCNT, and falling edges have been selected as the FIOCB pin input capture input edge. And both rising and falling edges have been selected as the FIOCA pin input capture input edge.

As buffer operation has been set, when the TRDCNT value is stored in GRA upon the occurrence of input capture A, the value previously stored in GRA is simultaneously transferred to GRC. The transfer timing is shown in figure 14.47.

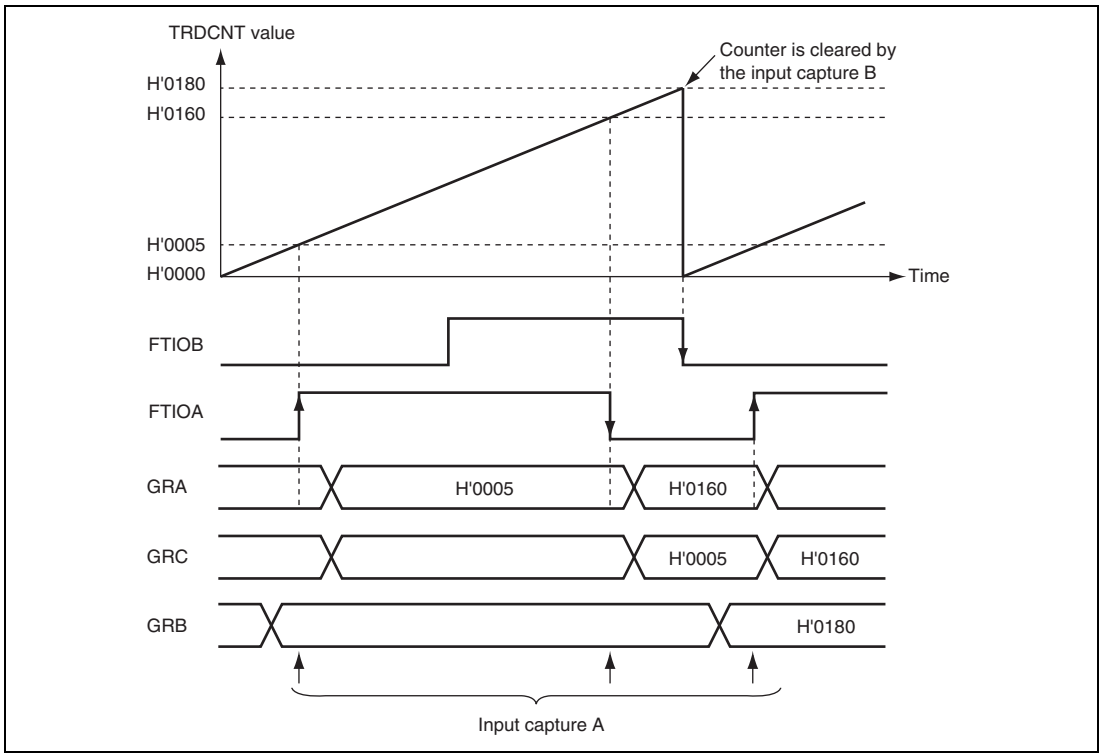



**Figure 14.46 Example of Buffer Operation (2)**
**(Buffer Operation for Input Capture Register)**

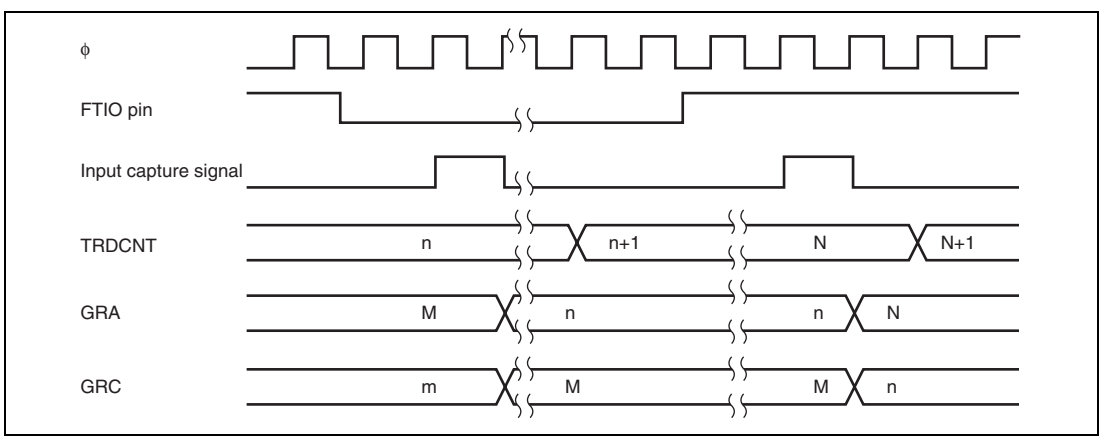



**Figure 14.47 Input Capture Timing of Buffer Operation**

Figures 14.48 and 14.49 show the operation examples when buffer operation has been designated for GRB_0 and GRD_0 in complementary PWM mode. These are examples when a PWM waveform of 0% duty is created by using the buffer operation and performing GRD_0 ≥ GRA_0. Data is transferred from GRD_0 to GRB_0 according to the settings of CMD0 and CMD1 when TRDCNT_0 and GRA_0 are compared and their contents match or when TRDCNT_1 underflows. However, when GRD_0 ≥ GRA_0, data is transferred from GRD_0 to GRB_0 when TRDCNT_1 underflows regardless of the setting of CMD0 and CMD1. When GRD_0 = H'0000, data is transferred from GRD_0 to GRB_0 when TRDCNT_0 and GRA_0 are compared and their contents match regardless of the settings of CMD0 and CMD1.

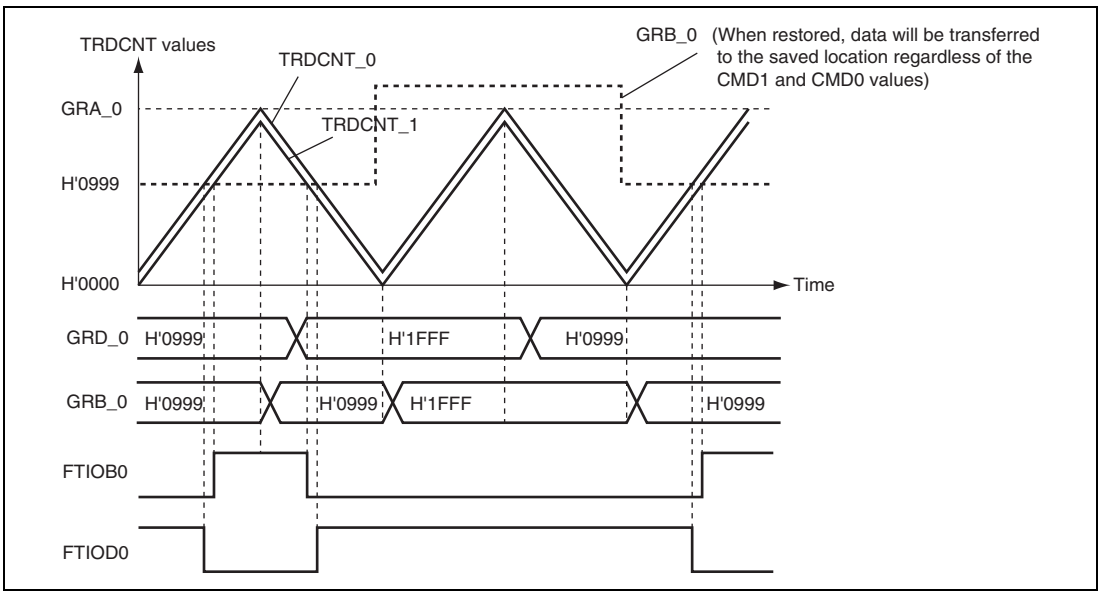



**Figure 14.48 Buffer Operation (3)**
**(Buffer Operation in Complementary PWM Mode CMD1 = CMD0 = 1)**

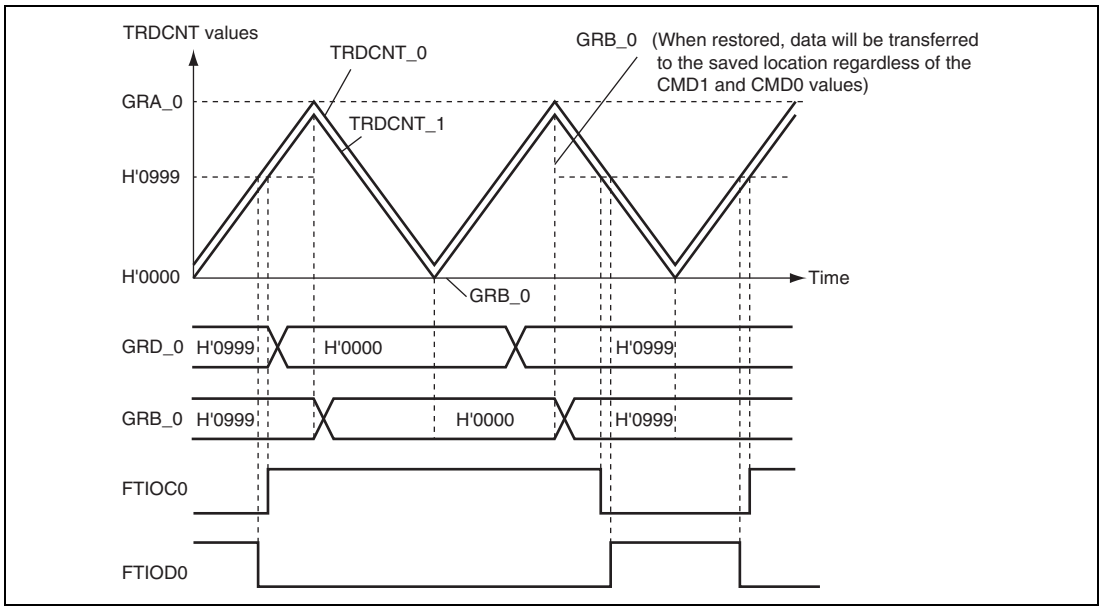



**Figure 14.49 Buffer Operation (4)**
**(Buffer Operation in Complementary PWM Mode CMD1 =1, CMD0 = 0)**

### 14.4.10 Timer RD Output Timing

The outputs of channels 0 and 1 can be disabled or inverted by the settings of TRDOER1 and TRDOCR and the external level.

#### (1) Output Disable/Enable Timing of Timer RD by TRDOER1

Setting the master enable bit in TRDOER1 to 1 disables the output of timer RD. By setting the PCR and PDR of the corresponding I/O port beforehand, any value can be output. Figure 14.50 shows the timing to enable or disable the output of timer RD by TRDOER1.

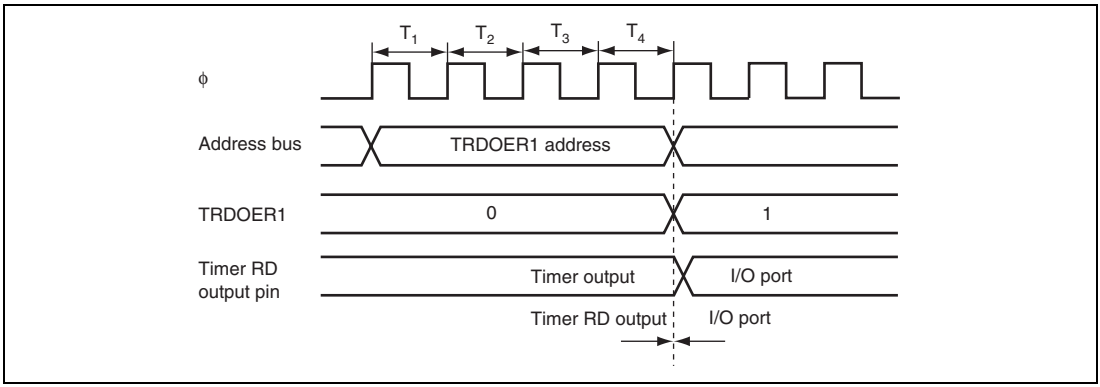



**Figure 14.50 Example of Output Disable Timing of Timer RD by Writing to TRDOER1**

#### (2) Output Disable Timing of Timer RD by External Trigger

When PH5/$\overline{\text{TRDOI_0}}$ (or PH6/$\overline{\text{TRDOI_1}}$) is set as a $\overline{\text{TRDOI}}$ input pin, and low level is input to $\overline{\text{TRDOI}}$, the master enable bit in TRDOER1 is set to 1 and the output of timer RD will be disabled.

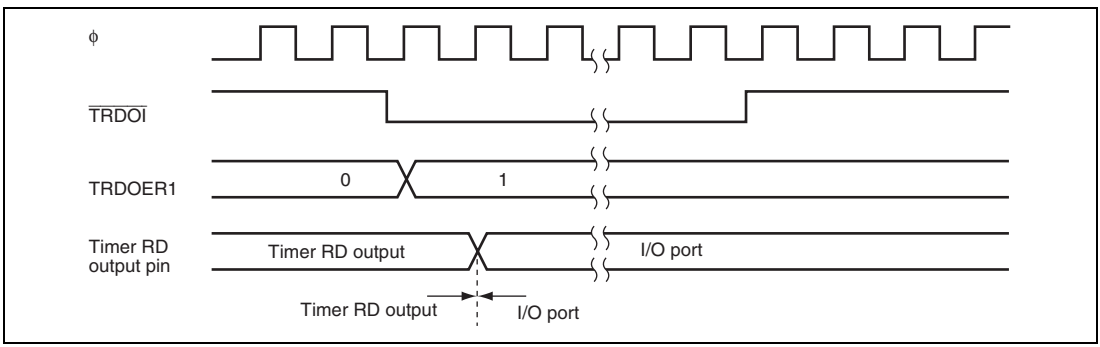



**Figure 14.51 Example of Output Disable Timing of Timer RD by External Trigger**

### (3) Output Inverse Timing by TRDFCR

The output level can be inverted by inverting the OLS1 and OLS0 bits in TRDFCR in reset synchronous PWM mode or complementary PWM mode. Figure 14.52 shows the timing.

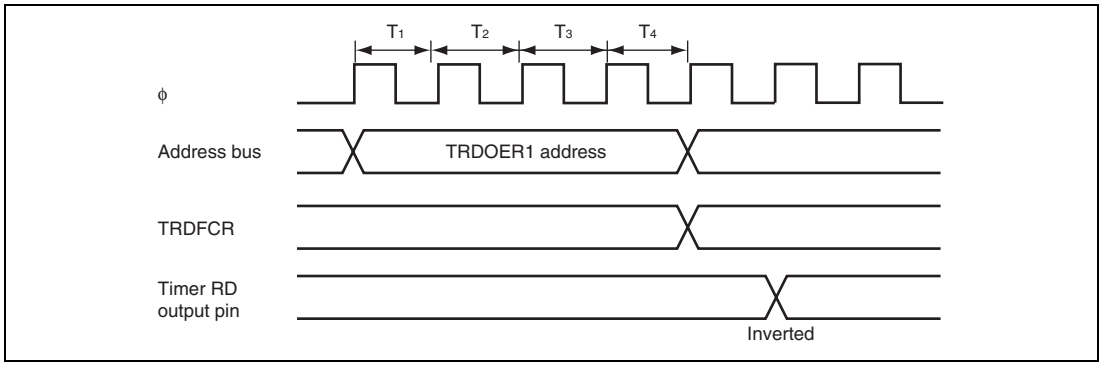



**Figure 14.52 Example of Output Inverse Timing of Timer RD by Writing to TRDFCR**

### (4) Output Inverse Timing by POCR

The output level can be inverted by inverting the POLD, POLC, and POLB bits in POCR in PWM mode. Figure 14.53 shows the timing.

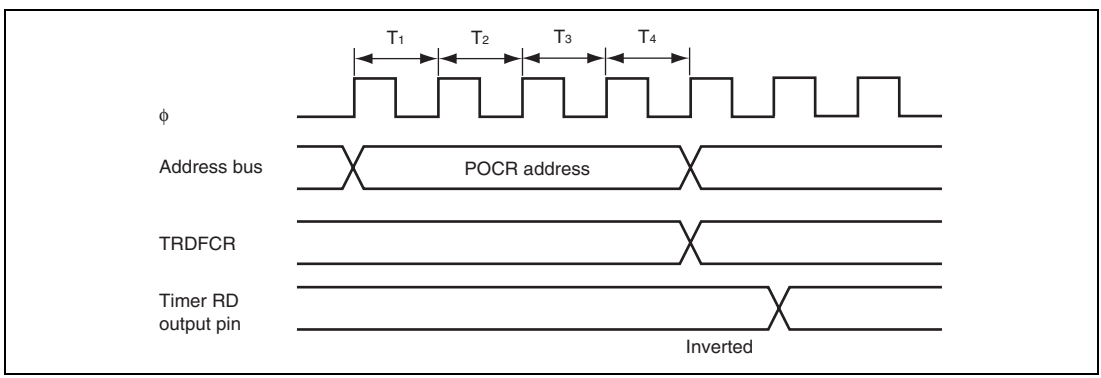



**Figure 14.53 Example of Output Inverse Timing of Timer RD by Writing to POCR**

### 14.4.11 Digital Filtering Function for Input Capture Inputs

Input signals on the FTIOA to FTIOD pins can be input via the digital filters. The digital filter includes three latches connected in series and a matching detecting circuit. The latches operate on the sampling clock specified by bits DFCK1 and DFCK0 in TRDDF and stores an input signal on the FTIOA to FTIOD pins. When outputs of the three latches match, the matching detecting circuit outputs the signal level of the input. Otherwise, the output remains unchanged. That is, when a pulse width is equal to or greater than three sampling clock cycles, the pulse is input as a signal. When a pulse width is less than three sampling clock cycles, the pulse is considered as a noise to be removed.

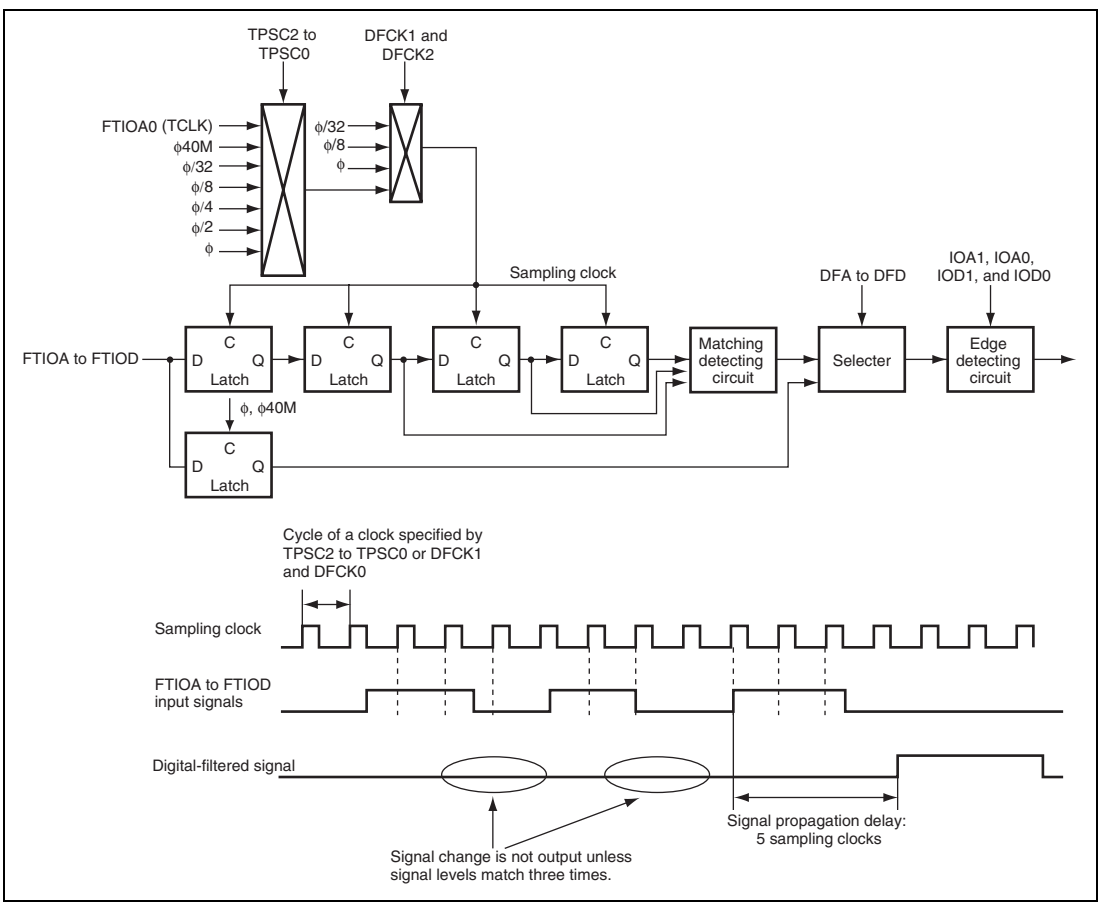



**Figure 14.54 Block Diagram of Digital Filter**

### 14.4.12 Function of Changing Output Pins for GR

With the settings of bits IOC3 and IOD3 in TRDIORC, pins for outputs of compare match signals for GRC and GRD can be changed from the FTIOC and FTIOD pins to the FTIOA and FTIOB pins. This means that the compare match A signal ORed with the compare match C signal can be output on the FTIOA pin. The compare match B ORed with the compare match D signal can be output on the FTIOB pin. Figure 14.55 is a block diagram of this function. The setting for channel 0 is independent of that for channel 1.

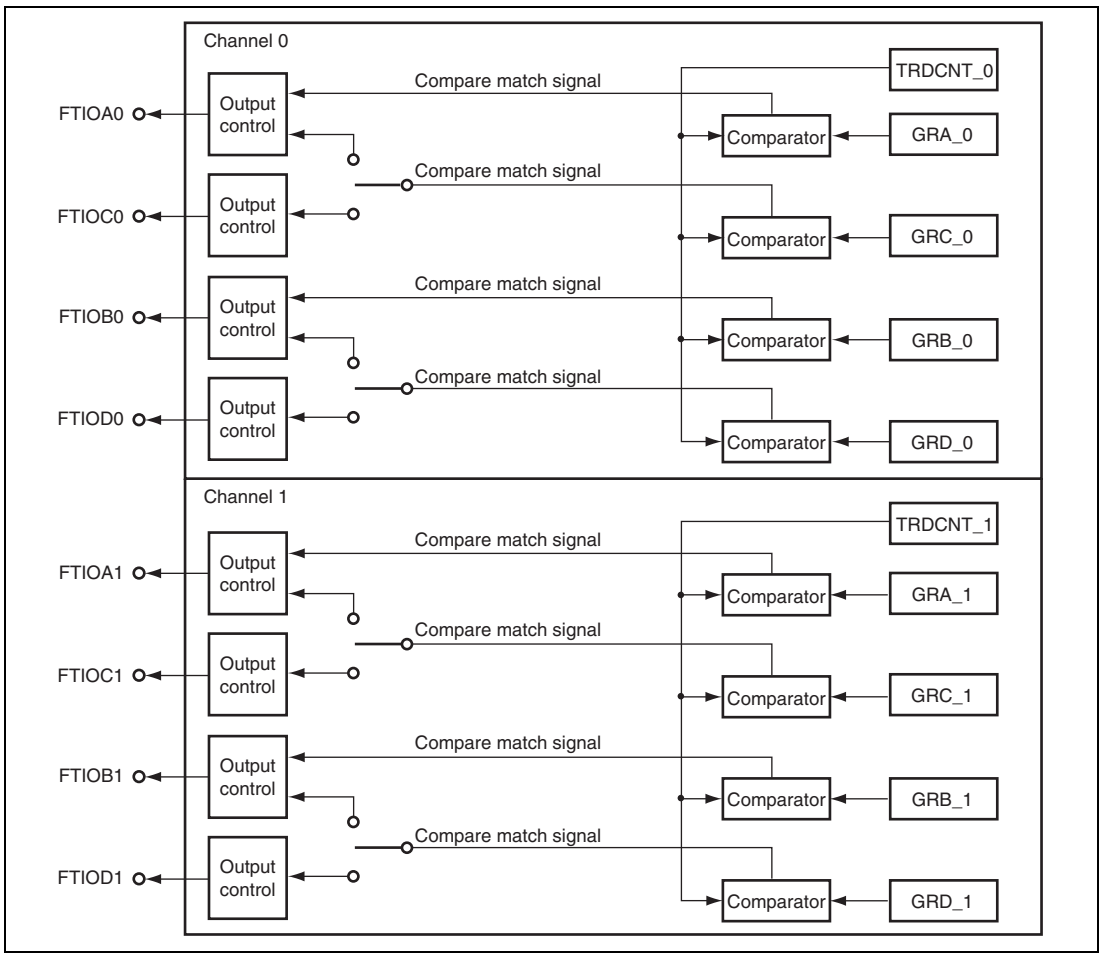



**Figure 14.55 Block Diagram of Output Pins for GR**

Figure 14.56 is an example when non-overlapped pulses are output on pins FTIOA0 and FTIOB0. In this example, TRDCNT_0 functions as a periodic counter which is cleared on compare match A0 (bits CCLR2 to CCLR0 in TRDCR_0 are set to B'001), an output signal is toggled on compare match A (bits IOA2 to IOA0 in TRDIORA_1 are set to B'011), the output signal on the FTIOA pin is toggled on compare match C (GRC_0) (bits IOC3 to IOC0 in TRDIORC_1 are set to B'0X11), an output signal is toggled on compare match B (GRB_0) (bits IOB2 to IOB0 in TRDIORA_1), and the output signal on the FTIOB pin is toggled on compare match D (GRD_0) (bits IOD3 to IOD0 in TRDIORC_1) are set to B'0X11). The cycle of the pulse is arbitrary.

Similarly, figure 14.57 is an example when non-overlapped pulses are output using TRDCNT_1.

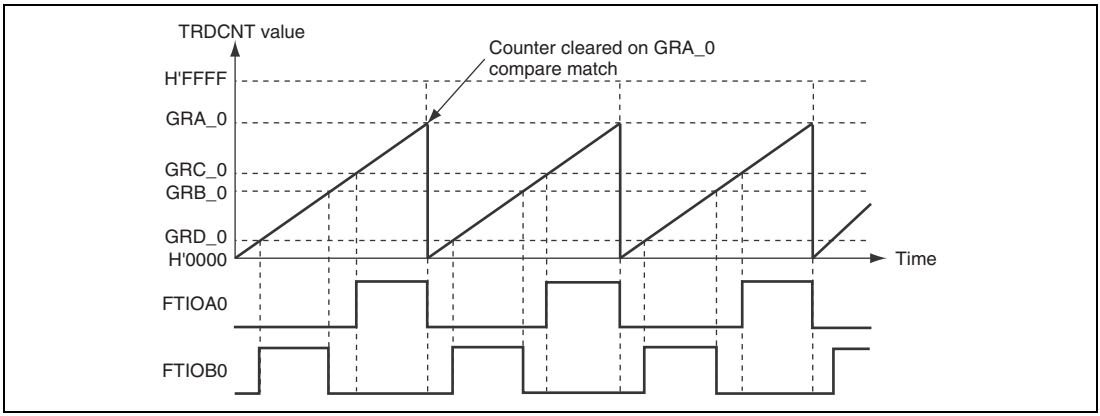



**Figure 14.56 Example of Non-Overlapped Pulses Output on Pins FTIOA0 and FTIOB0 (TRDCNT_0 Used)**

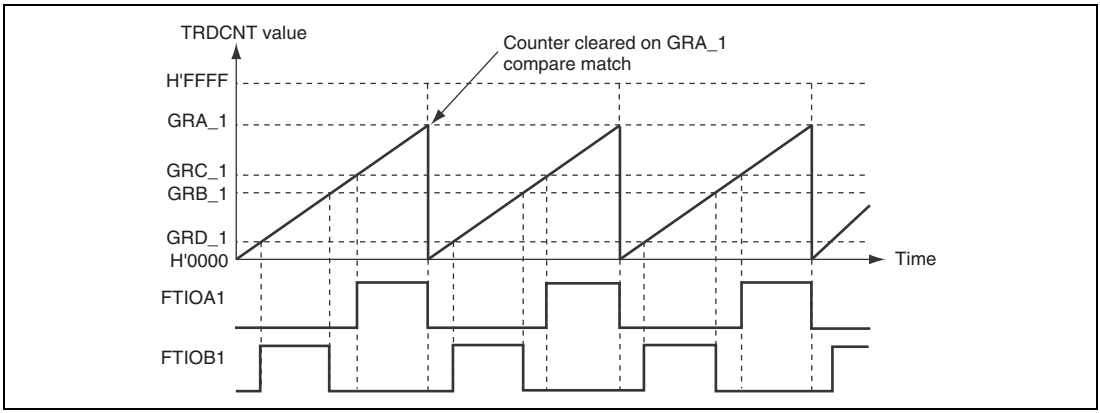



**Figure 14.57 Example of Non-Overlapped Pulses Output on Pins FTIOA1 and FTIOB1 (TRDCNT_1 Used)**

## 14.5 Interrupt Sources

There are three kinds of timer RD interrupt sources; input capture/compare match, overflow, and underflow. An interrupt is requested when the corresponding interrupt request flag is set to 1 while the corresponding interrupt enable bit is set to 1.

### 14.5.1 Status Flag Set Timing

#### (1) IMF Flag Set Timing

The IMF flag is set to 1 by the compare match signal that is generated when the GR matches with the TRDCNT. The compare match signal is generated at the last state of matching (timing to update the counter value when the GR and TRDCNT match). Therefore, when the TRDCNT and GR matches, the compare match signal will not be generated until the TRDCNT input clock is generated. Figure 14.58 shows the timing to set the IMF flag.

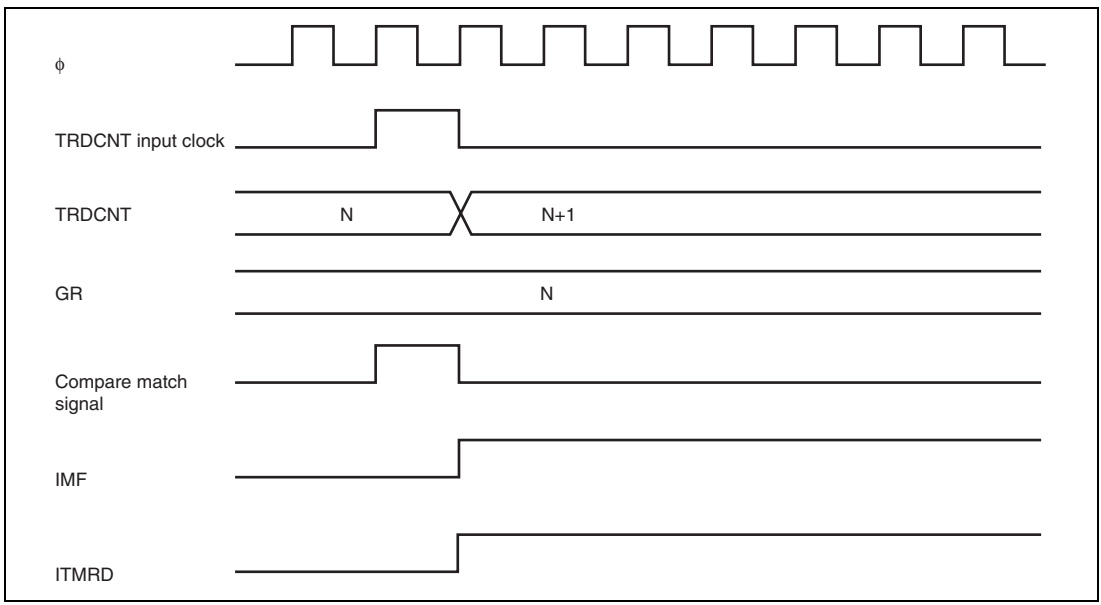



**Figure 14.58 IMF Flag Set Timing when Compare Match Occurs**

### (2) IMF Flag Set Timing at Input Capture

When an input capture signal is generated, the IMF flag is set to 1 and the value of TRDCNT is simultaneously transferred to corresponding GR. Figure 14.59 shows the timing.

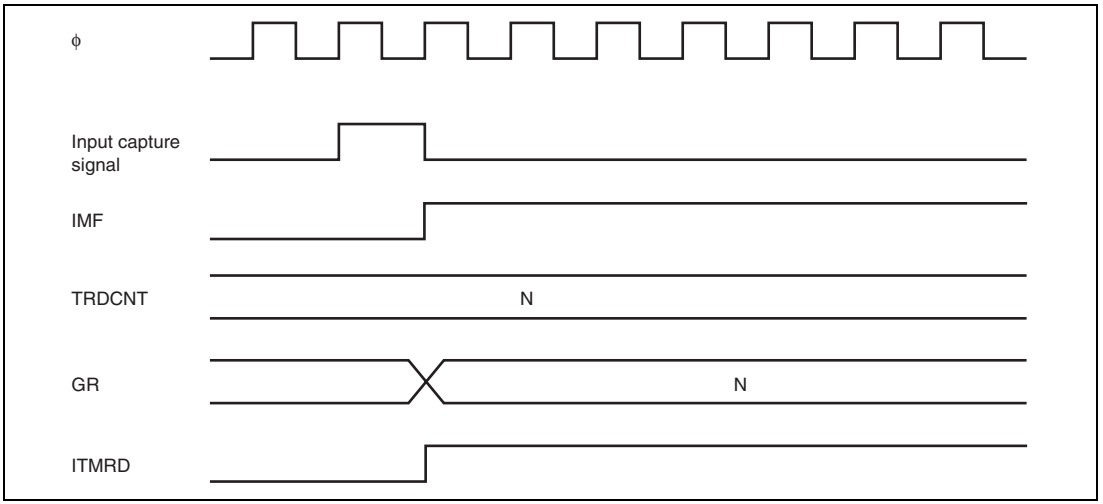



**Figure 14.59 IMF Flag Set Timing at Input Capture**

### (3) Overflow Flag (OVF) Set Timing

The overflow flag is set to 1 when the TRDCNT overflows. Figure 14.60 shows the timing.

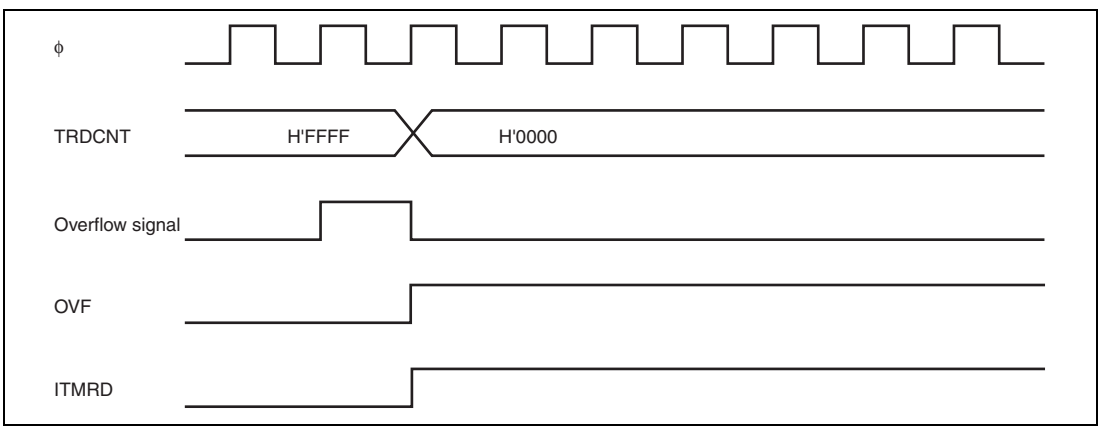



**Figure 14.60 OVF Flag Set Timing**

### 14.5.2 Status Flag Clearing Timing

The status flag can be cleared by writing 0 after reading 1 from the CPU. Figure 14.61 shows the timing in this case.

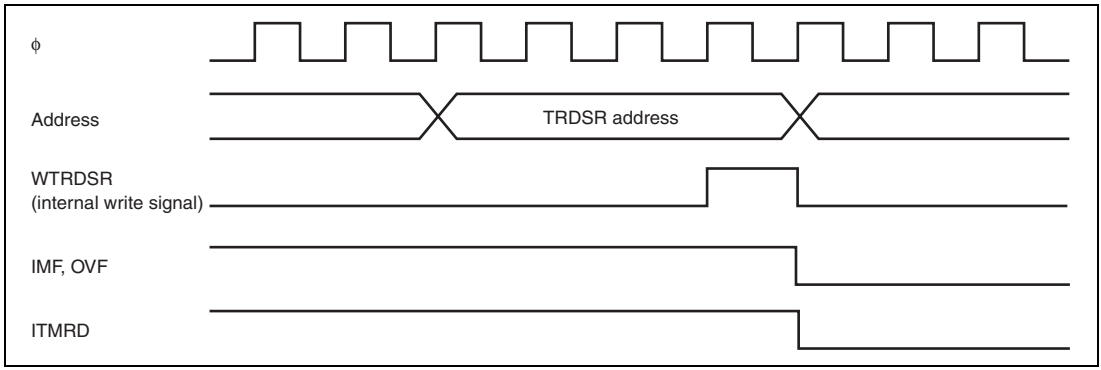



**Figure 14.61 Status Flag Clearing Timing**

## 14.6 Usage Notes

### (1) Input Pulse Width of Input Clock Signal and Input Capture Signal

The pulse width of the input clock signal and the input capture signal must be at least three system clock (ϕ) cycles when bits TPSC2 to TPSC0 in TRDCR = B'0XX or B'10X, or at least three on-chip oscillator clock (ϕ40M) cycles when B'110; shorter pulses will not be detected correctly.

### (2) Conflict between TRDCNT Write and Clear Operations

If a counter clear signal is generated in the $T_4$ state of a TRDCNT write cycle, TRDCNT clearing has priority and the TRDCNT write is not performed. Figure 14.62 shows the timing in this case.

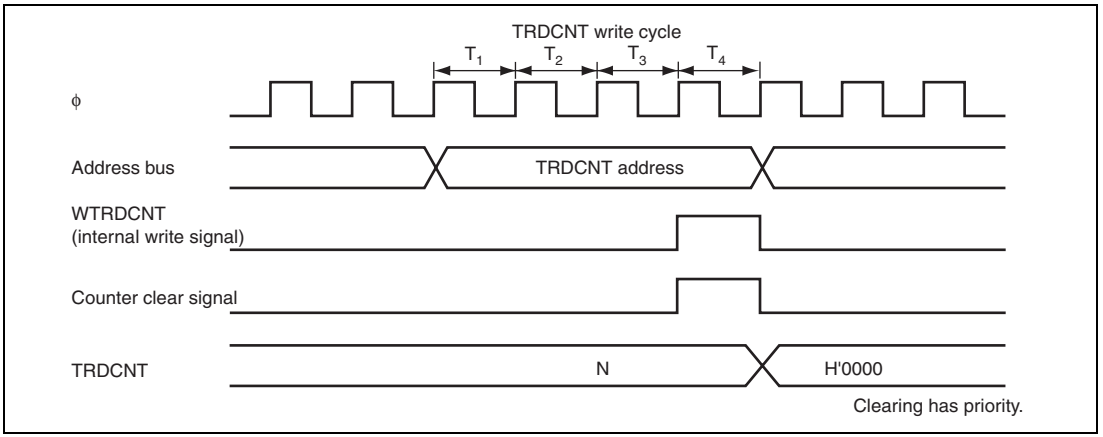



**Figure 14.62 Conflict between TRDCNT Write and Clear Operations**

### (3) Conflict between TRDCNT Write and Increment Operations

If TRDCNT is incremented in the $T_4$ state of a TRDCNT write cycle, writing has priority. Figure 14.63 shows the timing in this case.

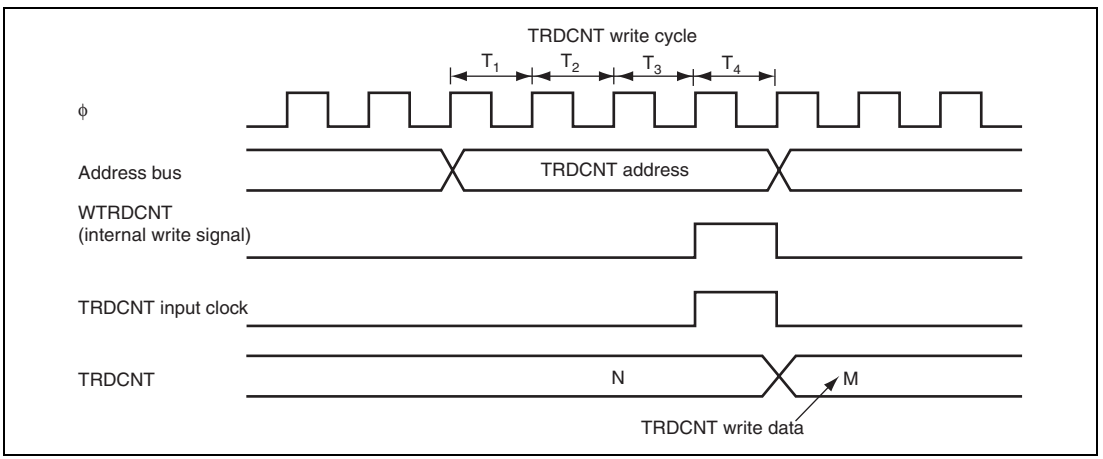



**Figure 14.63 Conflict between TRDCNT Write and Increment Operations**

### (4) Conflict between GR Write and Compare Match

If a compare match occurs in the $T_4$ state of a GR write cycle, GR write has priority and the compare match signal is disabled. Figure 14.64 shows the timing in this case.

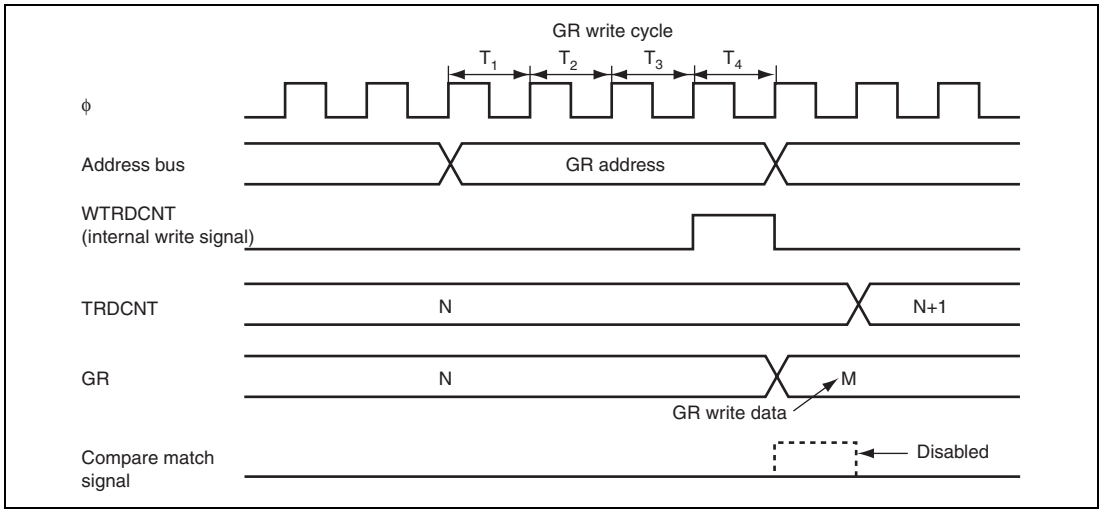



**Figure 14.64 Conflict between GR Write and Compare Match**

### (5) Conflict between TRDCNT Write and Overflow/Underflow

If overflow/underflow occurs in the $T_4$ state of a TRDCNT write cycle, TRDCNT write has priority without an increment operation. At this time, the OVF flag is set to 1. Figure 14.65 shows the timing in this case.

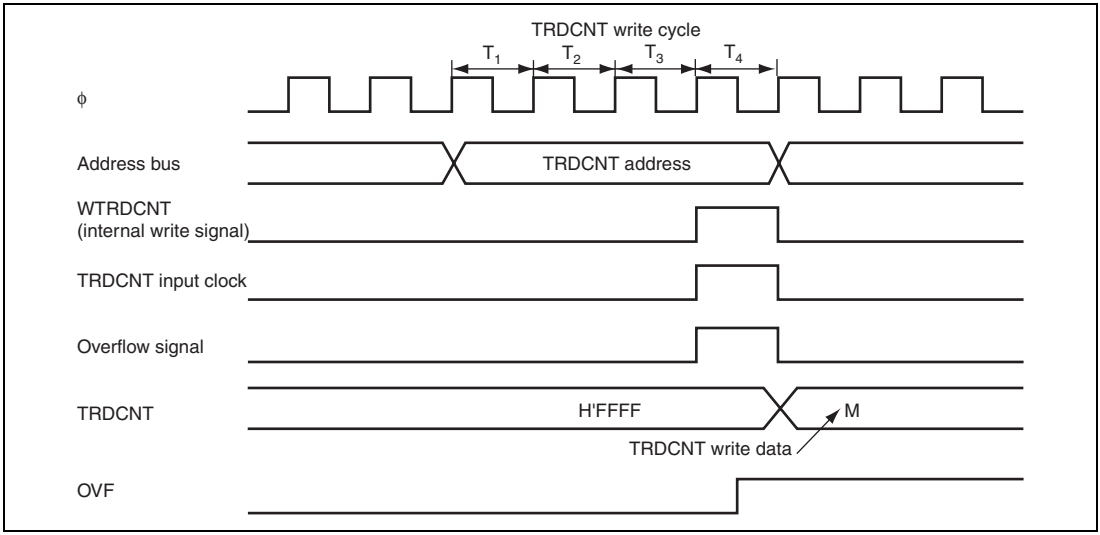



**Figure 14.65 Conflict between TRDCNT Write and Overflow**

### (6) Conflict between GR Read and Input Capture

If an input capture signal is generated in the $T_4$ state of a GR read cycle, the data that is read will be transferred before input capture transfer. Figure 14.66 shows the timing in this case.

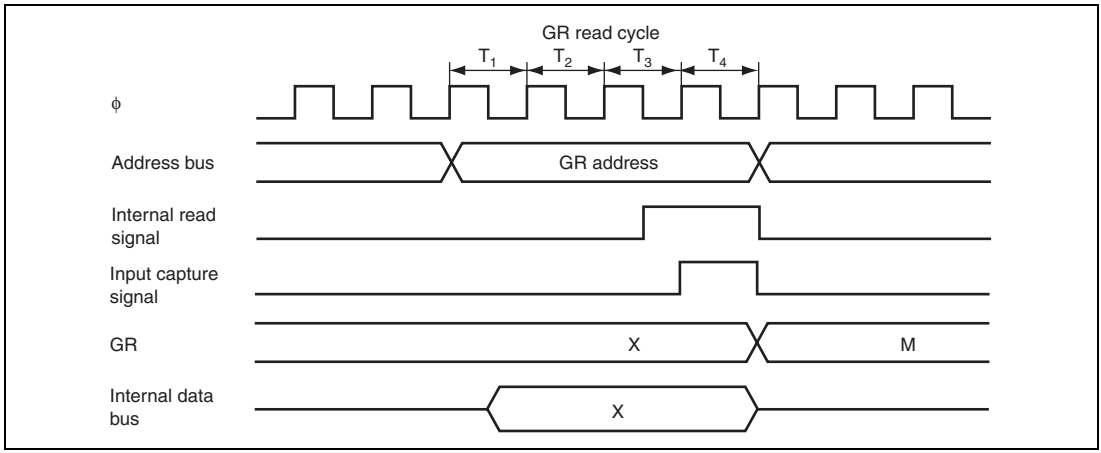



**Figure 14.66 Conflict between GR Read and Input Capture**

### (7) Conflict between Count Clearing and Increment Operations by Input Capture

If an input capture and increment signals are simultaneously generated, count clearing by the input capture operation has priority without an increment operation. The TRDCNT contents before clearing counter are transferred to GR. Figure 14.67 shows the timing in this case.

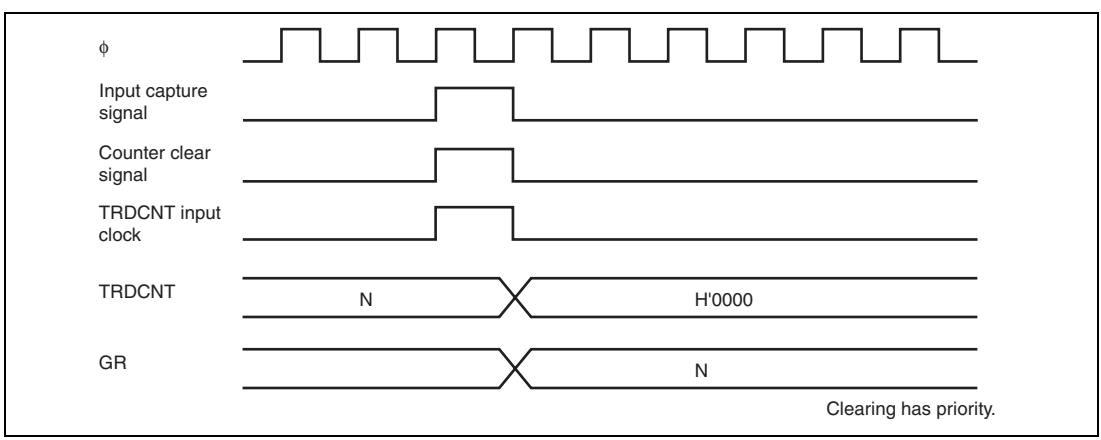



**Figure 14.67 Conflict between Count Clearing and Increment Operations by Input Capture**

### (8) Conflict between GR Write and Input Capture

If an input capture signal is generated in the $T_4$ state of a GR write cycle, the input capture operation has priority and the write to GR is not performed. Figure 14.68 shows the timing in this case.

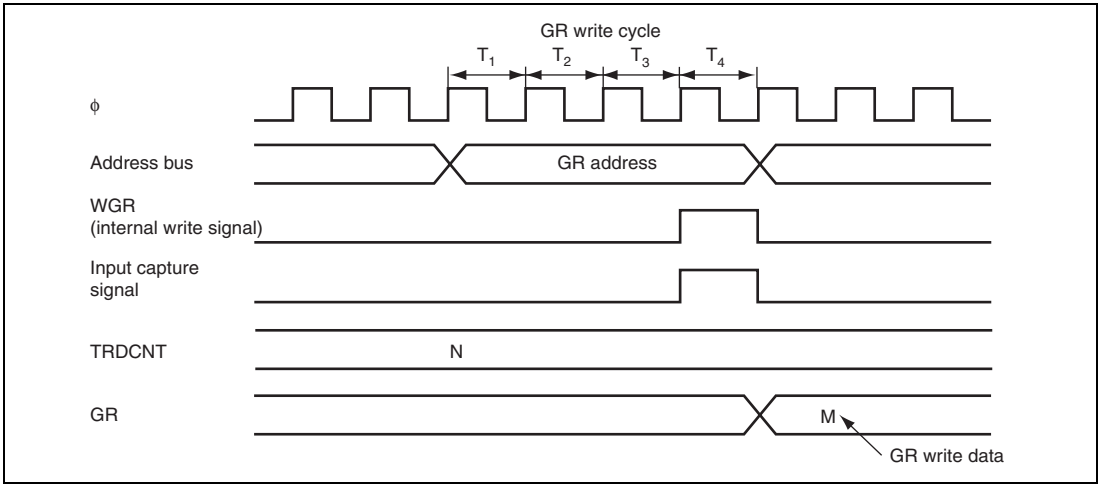



**Figure 14.68 Conflict between GR Write and Input Capture**

### (9) Notes on Setting Reset Synchronous PWM Mode/Complementary PWM Mode

When bits CMD1 and CMD0 in TRDFCR are set, note the following:

- Write bits CMD1 and CMD0 while TRDCNT_1 and TRDCNT_0 are halted.
- Changing the settings of reset synchronous PWM mode to complementary PWM mode or vice versa is disabled. Set reset synchronous PWM mode or complementary PWM mode after the normal operation (bits CMD1 and CMD0 are cleared to 0) has been set.

### (10) Note on Writing to the TOA0 to TOD0 Bits and the TOA1 to TOD1 Bits in TRDOCR

The TOA0 to TOD0 bits and the TOA1 to TOD1 bits in TRDOCR decide the value of the FTIO pin, which is output until the first compare match occurs. Once a compare match occurs and this compare match changes the values of FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 output, the values of the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pin output and the values read from the TOA0 to TOD0 and TOA1 to TOD1 bits may differ. Moreover, when the writing to TRDOCR and the generation of the compare match A0 to D0 and A1 to D1 occur at the same timing, the writing to TRDOCR has the priority. Thus, output change due to the compare match is not reflected to the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pins. Therefore, when bit manipulation instruction is used to write to TRDOCR, the values of the FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 pin output may result in an unexpected result. When TRDOCR is to be written to while compare match is operating, stop the counter once before accessing to TRDOCR, read the port 6 state to reflect the values of FTIOA0 to FTIOD0 and FTIOA1 to FTIOD1 output, to TOA0 to TOD0 and TOA1 to TOD1, and then restart the counter. Figure 14.69 shows an example when the compare match and the bit manipulation instruction to TRDOCR occur at the same timing.

TRDOCR has been set to H'06. Compare match B0 and compare match C0 are used. The FTIOB0 pin is in the 1 output state, and is set to the toggle output or the 0 output by compare match B0.
When BCLR#2, @TRDOCR is executed to clear the TOC0 bit (the FTIOC0 signal is low) and compare match B0 occurs at the same timing as shown below, the H'02 writing to TRDOCR has priority and compare match B0 does not drive the FTIOB0 signal low; the FTIOB0 signal remains high.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDOCR | TOD1 | TOC1 | TOB1 | TOA1 | TOD0 | TOC0 | TOB0 | TOA0 |
| Set value | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

BCLR#2, @TRDOCR
(1) TRDOCR read operation: Read H'06
(2) Modify operation: Modify H'06 to H'02
(3) Write operation to TRDOCR: Write H'02

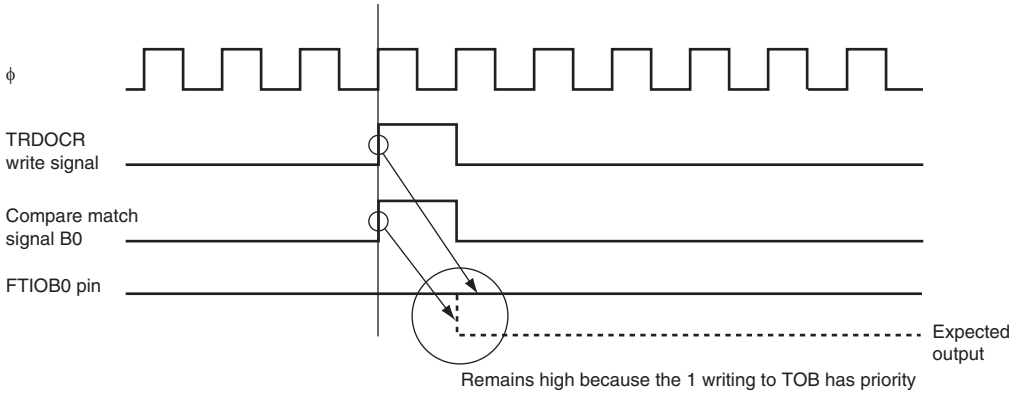



**Figure 14.69 When Compare Match and Bit Manipulation Instruction to TRDOCR Occur at the Same Timing**

# Section 15 Watchdog Timer

The watchdog timer is an 8-bit timer that can generate an internal reset signal for this LSI if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow.

The block diagram of the watchdog timer is shown in figure 15.1.

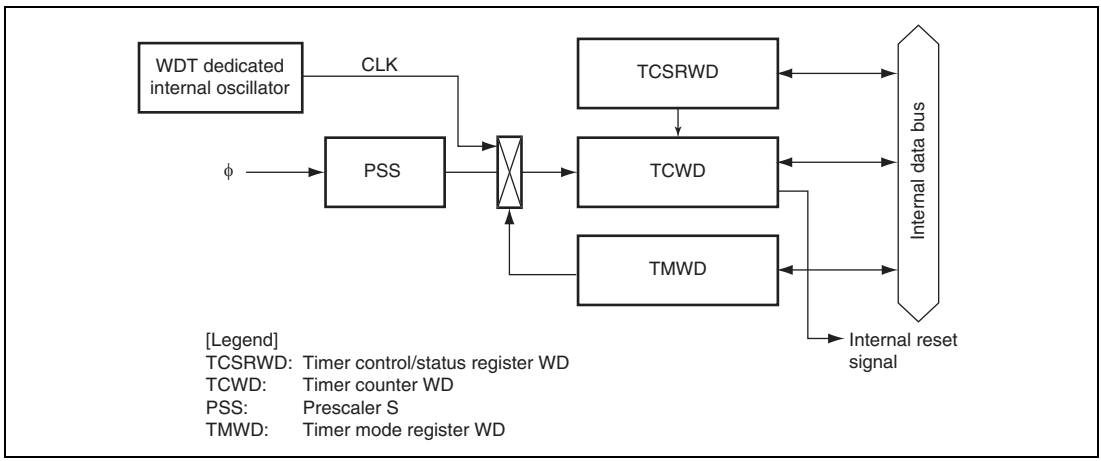



**Figure 15.1 Block Diagram of Watchdog Timer**

## 15.1 Features

- Selectable from nine counter input clocks.
  Eight clock sources (ϕ/64, ϕ/128, ϕ/256, ϕ/512, ϕ/1024, ϕ/2048, ϕ/4096, and ϕ/8192) or the WDT dedicated internal oscillator can be selected as the timer-counter clock. When the WDT dedicated internal oscillator is selected, it can operate as the watchdog timer in any operating mode.
- Reset signal generated on counter overflow
  An overflow period of 1 to 256 times the selected clock can be set.
- The watchdog timer is enabled in the initial state.
  It starts operating after the reset state is lifted.

## 15.2 Register Descriptions

The watchdog timer has the following registers.

- Timer control/status register WD (TCSRWD)
- Timer counter WD (TCWD)
- Timer mode register WD (TMWD)

### 15.2.1 Timer Control/Status Register WD (TCSRWD)

TCSRWD performs the TCSRWD and TCWD write control. TCSRWD also controls the watchdog timer operation and indicates the operating state. TCSRWD must be rewritten by using the MOV instruction. The bit manipulation instruction cannot be used to change the setting value.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | B6WI | 1 | R/W | Bit 6 Write Inhibit<br>The TCWE bit can be written only when the write value of the B6WI bit is 0.<br>This bit is always read as 1. |
| 6 | TCWE | 0 | R/W | Timer Counter WD Write Enable<br>TCWD can be written when the TCWE bit is set to 1.<br>When writing data to this bit, the value for bit 7 must be 0. |
| 5 | B4WI | 1 | R/W | Bit 4 Write Inhibit<br>The TCSRWE bit can be written only when the write value of the B4WI bit is 0. This bit is always read as 1. |
| 4 | TCSRWE | 0 | R/W | Timer Control/Status Register WD Write Enable<br>The WDON and WRST bits can be written when the TCSRWE bit is set to 1.<br>When writing data to this bit, the value for bit 5 must be 0. |
| 3 | B2WI | 1 | R/W | Bit 2 Write Inhibit<br>This bit can be written to the WDON bit only when the write value of the B2WI bit is 0.<br>This bit is always read as 1. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | WDON | 1 | R/W | Watchdog Timer On<br>TCWD starts counting up when the WDON bit is set to 1 and halts when the WDON bit is cleared to 0. The watchdog timer is enabled in the initial state. When the watchdog timer is not used, clear the WDON bit to 0.<br>[Setting conditions]<br>• Reset<br>• When 1 is written to the WDON bit and 0 is written to the B2WI bit while the TCSRWE bit = 1<br>[Clearing condition]<br>• When 0 is written to the WDON bit and 0 is written to the B2WI bit while the TCSRWE bit = 1 |
| 1 | B0WI | 1 | R/W | Bit 0 Write Inhibit<br>This bit can be written to the WRST bit only when the write value of the B0WI bit is 0. This bit is always read as 1. |
| 0 | WRST | 0 | R/W | Watchdog Timer Reset<br>[Setting condition]<br>• When TCWD overflows and an internal reset signal is generated<br>[Clearing conditions]<br>• Reset by the $\overline{\text{RES}}$ pin<br>• When 0 is written to the WRST bit and 0 is written to the B0WI bit while the TCSRWE bit = 1 |

### 15.2.2 Timer Counter WD (TCWD)

TCWD is an 8-bit readable/writable up-counter. When TCWD overflows from H'FF to H'00, the internal reset signal is generated and the WRST bit in TCSRWD is set to 1. TCWD is initialized to H'00.

### 15.2.3 Timer Mode Register WD (TMWD)

TMWD selects the input clock.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 4 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 3 | CKS3 | 1 | R/W | Clock Select 3 to 0 |
| 2 | CKS2 | 1 | R/W | Select the clock to be input to TCWD. |
| 1 | CKS1 | 1 | R/W | 1000: Internal clock: counts on $\phi$/64 |
| 0 | CKS0 | 1 | R/W | 1001: Internal clock: counts on $\phi$/128 |
| | | | | 1010: Internal clock: counts on $\phi$/256 |
| | | | | 1011: Internal clock: counts on $\phi$/512 |
| | | | | 1100: Internal clock: counts on $\phi$/1024 |
| | | | | 1101: Internal clock: counts on $\phi$/2048 |
| | | | | 1110: Internal clock: counts on $\phi$/4096 |
| | | | | 1111: Internal clock: counts on $\phi$8192 |
| | | | | 0XXX: WDT dedicated internal oscillator |
| | | | | For the overflow periods of the WDT dedicated internal oscillator, see section 23, Electrical Characteristics. |

[Legend]
X: Don't care

## 15.3 Operation

The watchdog timer is provided with an 8-bit counter. After the reset state is released, TCWD starts counting up. When the TCWD count value overflows H'FF, an internal reset signal is generated. The internal reset signal is output for a period of 256 $\phi_{RC}$ clock cycles. As TCWD is a writable counter, it starts counting from the value set in TCWD. An overflow period in the range of 1 to 256 input clock cycles can therefore be set, according to the TCWD set value. When the watchdog timer is not used, stop TCWD counting by writing 0 to B2WI and WDON simultaneously while the TCSRWE bit in TCSRWD is set to 1. (To stop the watchdog timer, two write accesses to TCSRWD are required.)

Figure 15.2 shows an example of watchdog timer operation.

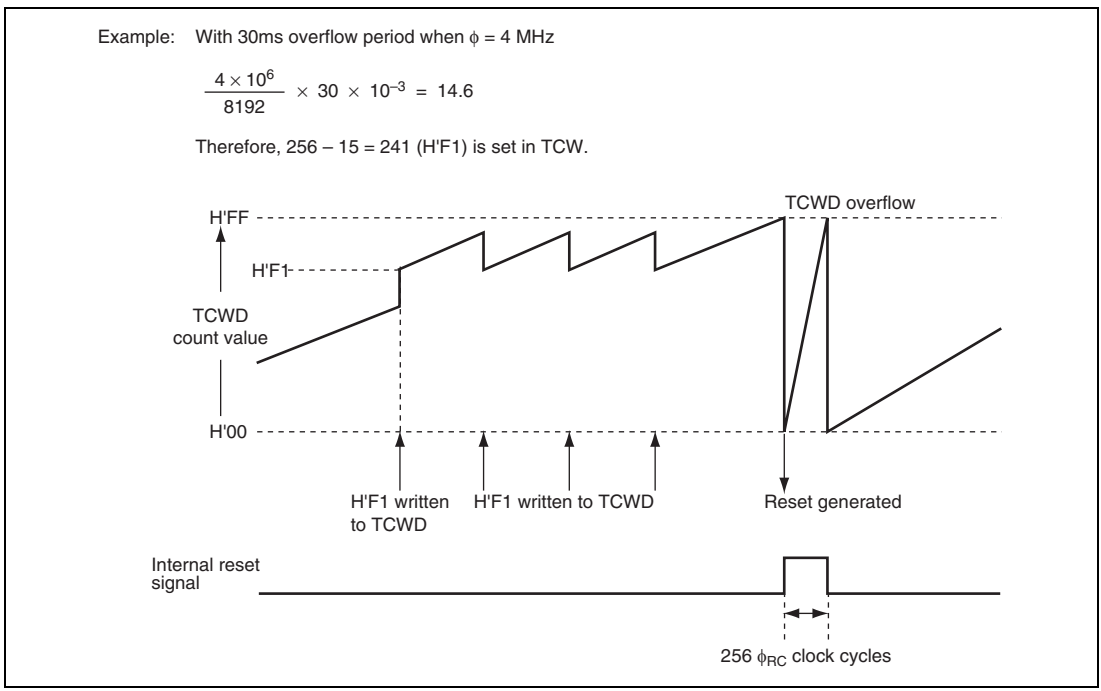



**Figure 15.2 Watchdog Timer Operation Example**

# Section 16 14-Bit PWM

The 14-bit PWM is a pulse division type PWM that can be used for electronic tuner control, etc. Figure 16.1 shows a block diagram of the 14-bit PWM.

## 16.1 Features

- Choice of two conversion periods
  A conversion period of 32768/ϕ with a minimum modulation width of 2/ϕ, or a conversion period of 16384/ϕ with a minimum modulation width of 1/ϕ, can be selected.
- Pulse division method for less ripple

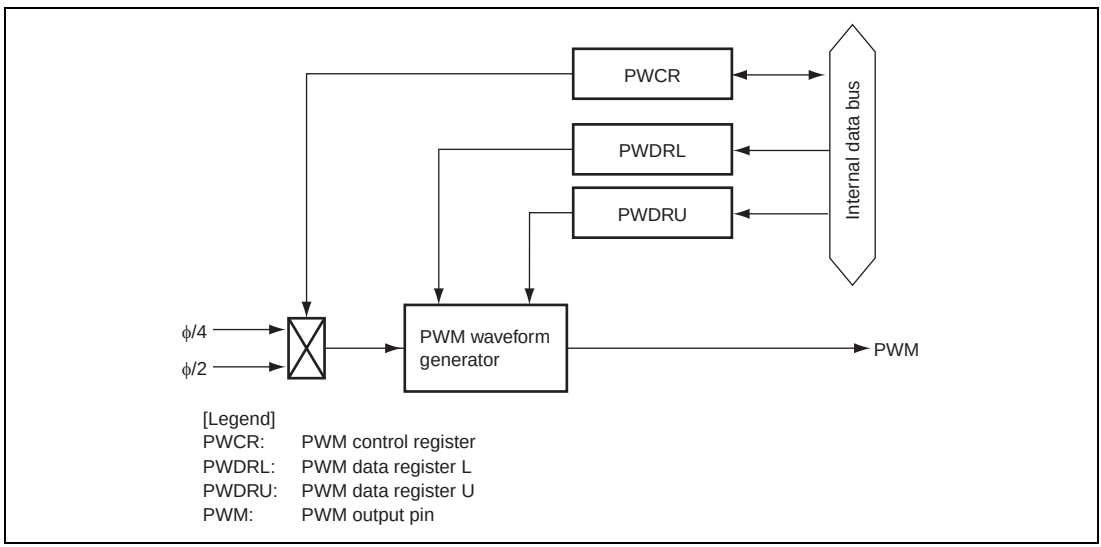



**Figure 16.1 Block Diagram of 14-Bit PWM**

## 16.2 Input/Output Pin

Table 16.1 shows the 14-bit PWM pin configuration.

**Table 16.1 Pin Configuration**

| Name | Abbreviation | I/O | Function |
|---|---|---|---|
| 14-bit PWM square-wave output | PWM | Output | 14-bit PWM square-wave output pin |

## 16.3 Register Descriptions

The 14-bit PWM has the following registers.

- PWM control register (PWCR)
- PWM data register U (PWDRU)
- PWM data register L (PWDRL)

### 16.3.1 PWM Control Register (PWCR)

PWCR selects the conversion period.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 1 | — | Reserved<br>These bits are always read as 1, and cannot be modified. |
| 0 | PWCR0 | 0 | R/W | Clock Select<br>0: The input clock is $\phi/2$ ($t\phi = 2/\phi$)<br>— The conversion period is $16384/\phi$, with a minimum modulation width of $1/\phi$<br>1: The input clock is $\phi/4$ ($t\phi = 4/\phi$)<br>— The conversion period is $32768/\phi$, with a minimum modulation width of $2/\phi$ |

[Legend] $t\phi$: Period of PWM clock input

### 16.3.2 PWM Data Registers U, L (PWDRU, PWDRL)

PWDRU and PWDRL indicate high level width in one PWM waveform cycle. PWDRU and PWDRL are 14-bit write-only registers, with the upper 6 bits assigned to PWDRU and the lower 8 bits to PWDRL. When read, all bits are always read as 1.

Both PWDRU and PWDRL are accessible only in bytes. Note that the operation is not guaranteed if word access is performed. When 14-bit data is written in PWDRU and PWDRL, the contents are latched in the PWM waveform generator and the PWM waveform generation data is updated. When writing the 14-bit data, the order is as follows: PWDRL to PWDRU.

PWDRU and PWDRL are initialized to H'C000.

## 16.4 Operation

When using the 14-bit PWM, set the registers in this sequence:

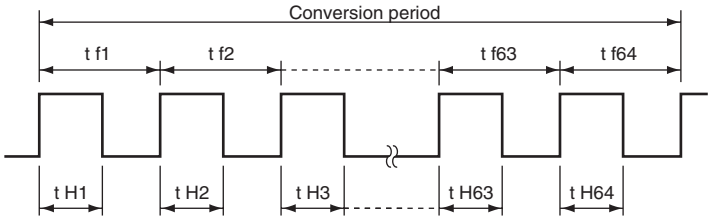
1. Set the PWM bit in the port mode register 1 (PMR1) to set the P11/PWM pin to function as a PWM output pin.
2. Set the PWCR0 bit in PWCR to select a conversion period of either.
3. Set the output waveform data in PWDRU and PWDRL. Be sure to write byte data first to PWDRL and then to PWDRU. When the data is written in PWDRU, the contents of these registers are latched in the PWM waveform generator, and the PWM waveform generation data is updated in synchronization with internal signals.

One conversion period consists of 64 pulses, as shown in figure 16.2. The total high-level width during this period ($T_H$) corresponds to the data in PWDRU and PWDRL. This relation can be expressed as follows:

$$T_H = (\text{data value in PWDRU and PWDRL} + 64) \times t\phi/2$$

where $t\phi$ is the period of PWM clock input: $2/\phi$ (bit PWCR0 = 0) or $4/\phi$ (bit PWCR0 = 1).
If the data value in PWDRU and PWDRL is from H'FFC0 to H'FFFF, the PWM output stays high. When the data value is H'C000, $T_H$ is calculated as follows:

$$T_H = 64 \times t\phi/2 = 32\ t\phi$$




T H = t H1 + t H2 + t H3 + ··· + t H64

t f1 = t f2 = t f3 = ··· = t f64

**Figure 16.2 Waveform Output by 14-Bit PWM**

# Section 17 Serial Communication Interface 3 (SCI3)

This LSI includes a serial communication interface 3 (SCI3), which has independent three channels. The SCI3 can handle both asynchronous and clock synchronous serial communication. In asynchronous mode, serial data communication can be carried out using standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or an Asynchronous Communication Interface Adapter (ACIA). A function for serial communication between multiple processors (multiprocessor communication function) is also provided.

Table 17.1 shows the SCI3 channel configuration and figure 17.1 shows a block diagram of the SCI3. Since basic functions are identical for each of the three channels (SCI3, SCI3_2, and SCI3_3), separate explanations are not given in this section.

## 17.1 Features

- Choice of asynchronous or clock synchronous serial communication mode
- Full-duplex communication capability

  The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously.

  Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected
- External clock or on-chip baud rate generator can be selected as a transfer clock source.
- Six interrupt sources

  Transmit-end, transmit-data-empty, receive-data-full, overrun error, framing error, and parity error.
- Noise canceller (only for SCI3_3)

Asynchronous mode:

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RXD pin level directly in the case of a framing error

Clock synchronous mode:

- Data length: 8 bits
- Receive error detection: Overrun errors

**Table 17.1 Channel Configuration**

| Channel | Abbreviation | Pin | Register | Register Address | Noise Canceller |
|---|---|---|---|---|---|
| Channel 1 | SCI3*[2] | SCK3<br>RXD<br>TXD | SMR | H'FFFFA8 | None |
| | | | BRR | H'FFFFA9 | |
| | | | SCR3 | H'FFFFAA | |
| | | | TDR | H'FFFFAB | |
| | | | SSR | H'FFFFAC | |
| | | | RDR | H'FFFFAD | |
| | | | RSR | — | |
| | | | TSR | — | |
| Channel 2 | SCI3_2 | SCK3_2<br>RXD_2<br>TXD_2 | SMR_2 | H'FFF740 | None |
| | | | BRR_2 | H'FFF741 | |
| | | | SCR3_2 | H'FFF742 | |
| | | | TDR_2 | H'FFF743 | |
| | | | SSR_2 | H'FFF744 | |
| | | | RDR_2 | H'FFF745 | |
| | | | RSR_2 | — | |
| | | | TSR_2 | — | |
| Channel 3 | SCI3_3 | SCK3_3<br>RXD_3<br>TXD_3 | SMR_3 | H'FFF600 | Yes |
| | | | BRR_3 | H'FFF601 | |
| | | | SCR3_3 | H'FFF602 | |
| | | | TDR_3 | H'FFF603 | |
| | | | SSR_3 | H'FFF604 | |
| | | | RDR_3 | H'FFF605 | |
| | | | RSR_3 | — | |
| | | | TSR_3 | — | |
| | | | SMCR_3*[1] | H'FFF608 | |

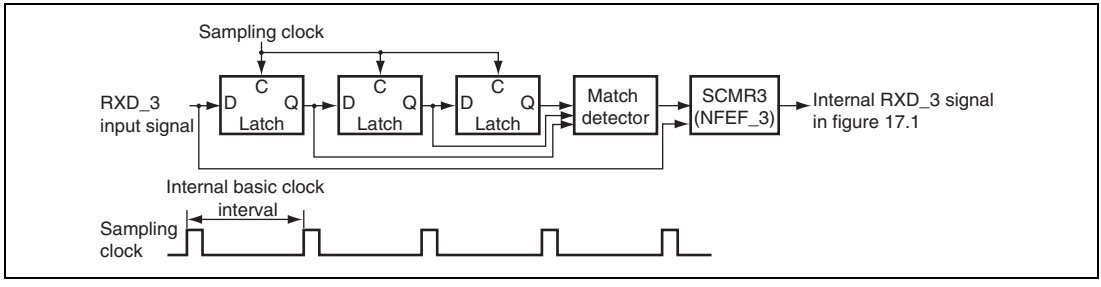
Note: 1. In addition to basic functions common in SCI3 and SCI3_2, SCI3_3 has the serial mode control register (SMCR). SMCR controls noise canceling on the RXD_3 input signal, PH2/TXD_3 pin function, and SCI3_3 module standby function.
2. The channel 1 of the SCI3 is used in on-board programming mode by boot mode.

- Serial mode control register (SMCR)

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 3 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 2 | NFEN_3 | 0 | R/W | Noise Cancel Function Select<br>When COM in SMR is cleared to 0 and this bit is set to 1, noise in the RXD_3 input signal is taken. |
| 1 | TXD_3 | 0 | R/W | TXD_3 Pin Select<br>Selects PH2/TXD_3 pin function.<br>0: General input pin is selected<br>1: TXD_3 output pin is selected |
| 0 | MSTS3_3 | 0 | R/W | SCI3_3 Module Standby<br>When this bit is set to 1, SCI3_3 enters in the standby state. |

- Noise canceller

The RXD_3 input signal is loaded internally via the noise canceller. The noise canceller consists of three latch circuits and match detection circuit connected in series. The RXD_3 input signal is sampled on the basic clock with a frequency 16 times the transfer rate, and the level is passed forward to the next circuit when outputs of three latches match. When the outputs are not match, previous value is retained. In other word, when the same level is retained more than three clocks, the input signal is acknowledged as a signal. When the level is changed within three clocks, the change is acknowledged as not a signal change but noise.




**Block Diagram of Noise Canceller**

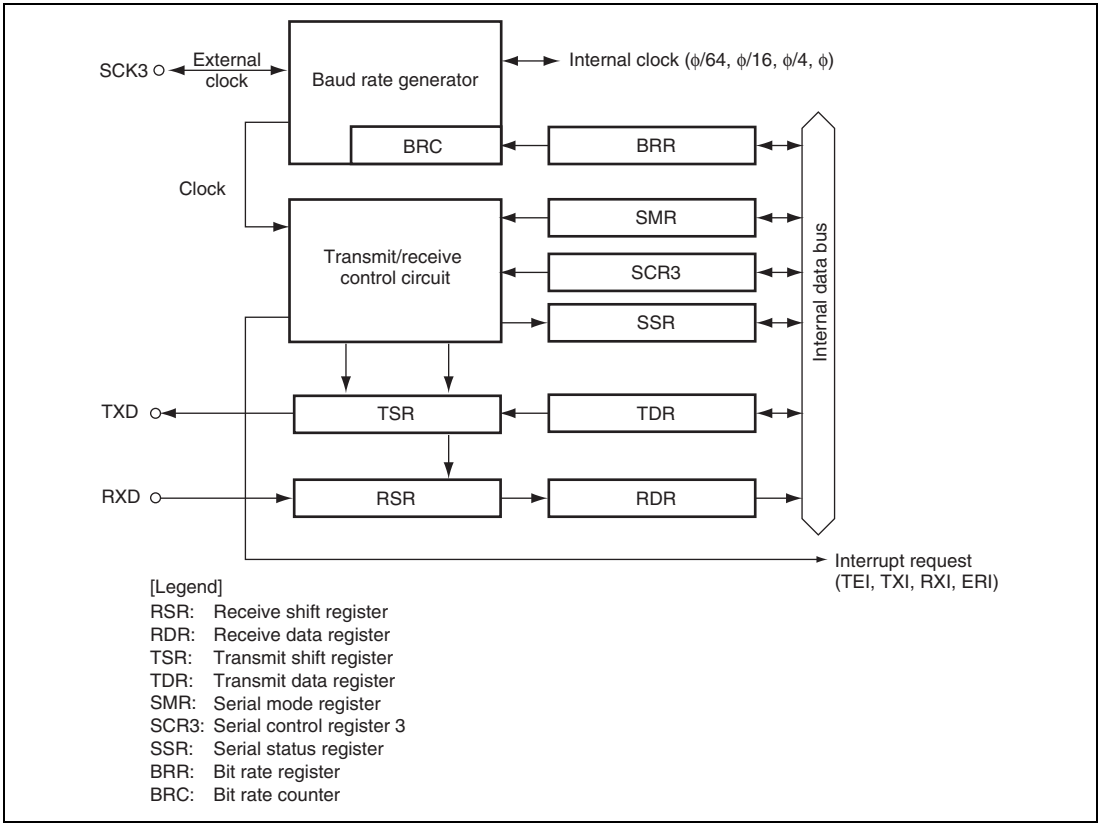



**Figure 17.1 Block Diagram of SCI3**

## 17.2 Input/Output Pins

Table 17.2 shows the SCI3 pin configuration.

**Table 17.2 Pin Configuration**

| Pin Name | Abbreviation | I/O | Function |
|---|---|---|---|
| SCI3 clock | SCK3 | I/O | SCI3 clock input/output |
| SCI3 receive data input | RXD | Input | SCI3 receive data input |
| SCI3 transmit data output | TXD | Output | SCI3 transmit data output |

## 17.3 Register Descriptions

The SCI3 has the following registers for each channel.

- Receive shift register (RSR)
- Receive data register (RDR)
- Transmit shift register (TSR)
- Transmit data register (TDR)
- Serial mode register (SMR)
- Serial control register 3 (SCR3)
- Serial status register (SSR)
- Bit rate register (BRR)
- Serial mode control register 3 (SMCR3)

### 17.3.1 Receive Shift Register (RSR)

RSR is a shift register that is used to receive serial data input from the RXD pin and convert it into parallel data. When one frame of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

### 17.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores received data. When the SCI3 has received one frame of serial data, it transfers the received serial data from RSR to RDR, where it is stored. After this, RSR is receive-enabled. As RSR and RDR function as a double buffer in this way, continuous receive operations are possible. After confirming that the RDRF bit in SSR is set to 1, read RDR only once. RDR cannot be written to by the CPU. RDR is initialized to H'00.

### 17.3.3 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI3 first transfers transmit data from TDR to TSR automatically, then sends the data that starts from the LSB to the TXD pin. TSR cannot be directly accessed by the CPU.

### 17.3.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for transmission. When the SCI3 detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structure of TDR and TSR enables continuous serial transmission. If the next transmit data has already been written to TDR during transmission of one-frame data, the SCI3 transfers the written data to TSR to continue transmission. To achieve reliable serial transmission, write transmit data to TDR only once after confirming that the TDRE bit in SSR is set to 1. TDR is initialized to H'FF.

### 17.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI3's serial transfer format and select the baud rate generator clock source.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | COM | 0 | R/W | Communication Mode<br>0: Asynchronous mode<br>1: Clock synchronous mode |
| 6 | CHR | 0 | R/W | Character Length (enabled only in asynchronous mode)<br>0: Selects 8 bits as the data length.<br>1: Selects 7 bits as the data length. |
| 5 | PE | 0 | R/W | Parity Enable (enabled only in asynchronous mode)<br>When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. |
| 4 | PM | 0 | R/W | Parity Mode (enabled only when the PE bit is 1 in asynchronous mode)<br>0: Selects even parity.<br>1: Selects odd parity. |
| 3 | STOP | 0 | R/W | Stop Bit Length (enabled only in asynchronous mode)<br>Selects the stop bit length in transmission.<br>0: 1 stop bit<br>1: 2 stop bits<br>For reception, only the first stop bit is checked, regardless of the value in the bit. If the second stop bit is 0, it is treated as the start bit of the next transmit character. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | MP | 0 | R/W | Multiprocessor Mode<br>When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and PM bit settings are invalid in multiprocessor mode. In clock synchronous mode, clear this bit to 0. |
| 1 | CKS1 | 0 | R/W | Clock Select 0 and 1 |
| 0 | CKS0 | 0 | R/W | These bits select the clock source for the baud rate generator.<br>00: $\phi$ clock (n = 0)<br>01: $\phi$/4 clock (n = 1)<br>10: $\phi$/16 clock (n = 2)<br>11: $\phi$/64 clock (n = 3)<br>For the relationship between the bit rate register setting and the baud rate, see section 17.3.8, Bit Rate Register (BRR). n is the decimal representation of the value of n in BRR (see section 17.3.8, Bit Rate Register (BRR)). |

### 17.3.6 Serial Control Register 3 (SCR3)

SCR3 is a register that enables or disables SCI3 transfer operations and interrupt requests, and is also used to select the transfer clock source. For details on interrupt requests, see section 17.7, Interrupt Requests.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TIE | 0 | R/W | Transmit Interrupt Enable<br>When this bit is set to 1, the TXI interrupt request is enabled. |
| 6 | RIE | 0 | R/W | Receive Interrupt Enable<br>When this bit is set to 1, RXI and ERI interrupt requests are enabled. |
| 5 | TE | 0 | R/W | Transmit Enable<br>When this bit s set to 1, transmission is enabled. |
| 4 | RE | 0 | R/W | Receive Enable<br>When this bit is set to 1, reception is enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | MPIE | 0 | R/W | Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode)<br>When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and OER status flags in SSR is disabled. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, see section 17.6, Multiprocessor Communication Function. |
| 2 | TEIE | 0 | R/W | Transmit End Interrupt Enable<br>When this bit is set to 1, TEI interrupt request is enabled. |
| 1 | CKE1 | 0 | R/W | Clock Enable 0 and 1 |
| 0 | CKE0 | 0 | R/W | Selects the clock source.<br>• Asynchronous mode<br>00: On-chip baud rate generator<br>01: On-chip baud rate generator<br>Outputs a clock of the same frequency as the bit rate from the SCK3 pin.<br>10: External clock<br>Inputs a clock with a frequency 16 times the bit rate from the SCK3 pin.<br>11: Reserved<br>• Clock synchronous mode<br>00: On-chip clock (SCK3 pin functions as clock output)<br>01: Reserved<br>10: External clock (SCK3 pin functions as clock input)<br>11: Reserved |

### 17.3.7 Serial Status Register (SSR)

SSR is a register containing status flags of the SCI3 and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, OER, PER, and FER; they can only be cleared.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TDRE | 1 | R/W | Transmit Data Register Empty<br>Indicates whether TDR contains transmit data.<br>[Setting conditions]<br>• When the TE bit in SCR3 is 0<br>• When data is transferred from TDR to TSR<br>[Clearing conditions]<br>• When 0 is written to TDRE after reading TDRE = 1<br>• When the transmit data is written to TDR |
| 6 | RDRF | 0 | R/W | Receive Data Register Full<br>Indicates that the received data is stored in RDR.<br>[Setting condition]<br>• When serial reception ends normally and receive data is transferred from RSR to RDR<br>[Clearing conditions]<br>• When 0 is written to RDRF after reading RDRF = 1<br>• When data is read from RDR |
| 5 | OER | 0 | R/W | Overrun Error<br>[Setting condition]<br>• When an overrun error occurs in reception<br>[Clearing condition]<br>• When 0 is written to OER after reading OER = 1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | FER | 0 | R/W | Framing Error<br>[Setting condition]<br>• When a framing error occurs in reception<br>[Clearing condition]<br>• When 0 is written to FER after reading FER = 1 |
| 3 | PER | 0 | R/W | Parity Error<br>[Setting condition]<br>• When a parity error is detected during reception<br>[Clearing condition]<br>• When 0 is written to PER after reading PER = 1 |
| 2 | TEND | 1 | R | Transmit End<br>[Setting conditions]<br>• When the TE bit in SCR3 is 0<br>• When TDRE = 1 at transmission of the last bit of a 1-frame serial transmit character<br>[Clearing conditions]<br>• When 0 is written to TDRE after reading TDRE = 1<br>• When the transmit data is written to TDR |
| 1 | MPBR | 0 | R | Multiprocessor Bit Receive<br>MPBR stores the multiprocessor bit in the receive character data. When the RE bit in SCR3 is cleared to 0, its state is retained. |
| 0 | MPBT | 0 | R/W | Multiprocessor Bit Transfer<br>MPBT stores the multiprocessor bit to be added to the transmit character data. |

### 17.3.8 Bit Rate Register (BRR)

BRR is an 8-bit register that adjusts the bit rate. The initial value of BRR is H'FF. Table 17.3 shows the relationship between the N setting in BRR and the n setting in bits CKS1 and CKS0 of SMR in asynchronous mode. Table 17.4 shows the maximum bit rate for each frequency in asynchronous mode. The values shown in both tables 17.3 and 17.4 are values in active (high-speed) mode. Table 17.5 shows the relationship between the N setting in BRR and the n setting in bits CKS1 and CKS0 of SMR in clock synchronous mode. The values shown in table 17.5 are values in active (high-speed) mode. The N setting in BRR and error for other operating frequencies and bit rates can be obtained by the following formulas:

**[Asynchronous Mode]**

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

**[Clock Synchronous Mode]**

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

[Legend]
B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
$\phi$: Operating frequency (MHz)
n: CSK1 and CSK0 settings in SMR ($0 \leq n \leq 3$)

**Table 17.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)**

| | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4 | | | 4.9152 | | | 5 | | | 6 | | |
| Bit Rate (bit/s) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 70 | 0.03 | 2 | 86 | 0.31 | 2 | 88 | –0.25 | 2 | 106 | –0.44 |
| 150 | 1 | 207 | 0.16 | 1 | 255 | 0.00 | 2 | 64 | 0.16 | 2 | 77 | 0.16 |
| 300 | 1 | 103 | 0.16 | 1 | 127 | 0.00 | 1 | 129 | 0.16 | 1 | 155 | 0.16 |
| 600 | 0 | 207 | 0.16 | 0 | 255 | 0.00 | 1 | 64 | 0.16 | 1 | 77 | 0.16 |
| 1200 | 0 | 103 | 0.16 | 0 | 127 | 0.00 | 0 | 129 | 0.16 | 0 | 155 | 0.16 |
| 2400 | 0 | 51 | 0.16 | 0 | 63 | 0.00 | 0 | 64 | 0.16 | 0 | 77 | 0.16 |
| 4800 | 0 | 25 | 0.16 | 0 | 31 | 0.00 | 0 | 32 | –1.36 | 0 | 38 | 0.16 |
| 9600 | 0 | 12 | 0.16 | 0 | 15 | 0.00 | 0 | 15 | 1.73 | 0 | 19 | –2.34 |
| 19200 | 0 | 6 | –6.99 | 0 | 7 | 0.00 | 0 | 7 | 1.73 | 0 | 9 | –2.34 |
| 31250 | 0 | 3 | 0.00 | 0 | 4 | –1.70 | 0 | 4 | 0.00 | 0 | 5 | 0.00 |
| 38400 | 0 | 2 | 8.51 | 0 | 3 | 0.00 | 0 | 3 | 1.73 | 0 | 4 | –2.34 |

| | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 6.144 | | | 7.3728 | | | 8 | | | 9.8304 | | |
| Bit Rate (bit/s) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 108 | 0.08 | 2 | 130 | –0.07 | 2 | 141 | 0.03 | 2 | 174 | –0.26 |
| 150 | 2 | 79 | 0.00 | 2 | 95 | 0.00 | 2 | 103 | 0.16 | 2 | 127 | 0.00 |
| 300 | 1 | 159 | 0.00 | 1 | 191 | 0.00 | 1 | 207 | 0.16 | 1 | 255 | 0.00 |
| 600 | 1 | 79 | 0.00 | 1 | 95 | 0.00 | 1 | 103 | 0.16 | 1 | 127 | 0.00 |
| 1200 | 0 | 159 | 0.00 | 0 | 191 | 0.00 | 0 | 207 | 0.16 | 0 | 255 | 0.00 |
| 2400 | 0 | 79 | 0.00 | 0 | 95 | 0.00 | 0 | 103 | 0.16 | 0 | 127 | 0.00 |
| 4800 | 0 | 39 | 0.00 | 0 | 47 | 0.00 | 0 | 51 | 0.16 | 0 | 63 | 0.00 |
| 9600 | 0 | 19 | 0.00 | 0 | 23 | 0.00 | 0 | 25 | 0.16 | 0 | 31 | 0.00 |
| 19200 | 0 | 9 | 0.00 | 0 | 11 | 0.00 | 0 | 12 | 0.16 | 0 | 15 | 0.00 |
| 31250 | 0 | 5 | 2.40 | 0 | 6 | 5.33 | 0 | 7 | 0.00 | 0 | 9 | –1.70 |
| 38400 | 0 | 4 | 0.00 | 0 | 5 | 0.00 | 0 | 6 | -6.99 | 0 | 7 | 0.00 |

[Legend]

—: A setting is available but error occurs

**Table 17.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2)**

| | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | | | 12 | | | 12.288 | | | 14 | | |
| Bit Rate (bit/s) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 2 | 177 | –0.25 | 2 | 212 | 0.03 | 2 | 217 | 0.08 | 2 | 248 | –0.17 |
| 150 | 2 | 129 | 0.16 | 2 | 155 | 0.16 | 2 | 159 | 0.00 | 2 | 181 | 0.16 |
| 300 | 2 | 64 | 0.16 | 2 | 77 | 0.16 | 2 | 79 | 0.00 | 2 | 90 | 0.16 |
| 600 | 1 | 129 | 0.16 | 1 | 155 | 0.16 | 1 | 159 | 0.00 | 1 | 181 | 0.16 |
| 1200 | 1 | 64 | 0.16 | 1 | 77 | 0.16 | 1 | 79 | 0.00 | 1 | 90 | 0.16 |
| 2400 | 0 | 129 | 0.16 | 0 | 155 | 0.16 | 0 | 159 | 0.00 | 0 | 181 | 0.16 |
| 4800 | 0 | 64 | 0.16 | 0 | 77 | 0.16 | 0 | 79 | 0.00 | 0 | 90 | 0.16 |
| 9600 | 0 | 32 | –1.36 | 0 | 38 | 0.16 | 0 | 39 | 0.00 | 0 | 45 | –0.93 |
| 19200 | 0 | 15 | 1.73 | 0 | 19 | –2.34 | 0 | 19 | 0.00 | 0 | 22 | –0.93 |
| 31250 | 0 | 9 | 0.00 | 0 | 11 | 0.00 | 0 | 11 | 2.40 | 0 | 13 | 0.00 |
| 38400 | 0 | 7 | 1.73 | 0 | 9 | –2.34 | 0 | 9 | 0.00 | — | — | — |

| | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 14.7456 | | | 16 | | | 18 | | | 20 | | |
| Bit Rate (bit/s) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |
| 110 | 3 | 64 | 0.70 | 3 | 70 | 0.03 | 3 | 79 | –0.12 | 3 | 88 | –0.25 |
| 150 | 2 | 191 | 0.00 | 2 | 207 | 0.16 | 2 | 233 | 0.16 | 3 | 64 | 0.16 |
| 300 | 2 | 95 | 0.00 | 2 | 103 | 0.16 | 2 | 116 | 0.16 | 2 | 129 | 0.16 |
| 600 | 1 | 191 | 0.00 | 1 | 207 | 0.16 | 1 | 233 | 0.16 | 2 | 64 | 0.16 |
| 1200 | 1 | 95 | 0.00 | 1 | 103 | 0.16 | 1 | 116 | 0.16 | 1 | 129 | 0.16 |
| 2400 | 0 | 191 | 0.00 | 0 | 207 | 0.16 | 0 | 233 | 0.16 | 1 | 64 | 0.16 |
| 4800 | 0 | 95 | 0.00 | 0 | 103 | 0.16 | 0 | 116 | 0.16 | 0 | 129 | 0.16 |
| 9600 | 0 | 47 | 0.00 | 0 | 51 | 0.16 | 0 | 58 | –0.96 | 0 | 64 | 0.16 |
| 19200 | 0 | 23 | 0.00 | 0 | 25 | 0.16 | 0 | 28 | 1.02 | 0 | 32 | –1.36 |
| 31250 | 0 | 14 | –1.70 | 0 | 15 | 0.00 | 0 | 17 | 0.00 | 0 | 19 | 0.00 |
| 38400 | 0 | 11 | 0.00 | 0 | 12 | 0.16 | 0 | 14 | –2.34 | 0 | 15 | 1.73 |

[Legend]

—: A setting is available but error occurs

**Table 17.4 Maximum Bit Rate for Each Frequency (Asynchronous Mode)**

| ϕ (MHz) | Maximum Bit Rate (bit/s) | n | N | ϕ (MHz) | Maximum Bit Rate (bit/s) | n | N |
|---|---|---|---|---|---|---|---|
| 4 | 125000 | 0 | 0 | 12 | 375000 | 0 | 0 |
| 4.9152 | 153600 | 0 | 0 | 12.288 | 384000 | 0 | 0 |
| 5 | 156250 | 0 | 0 | 14 | 437500 | 0 | 0 |
| 6 | 187500 | 0 | 0 | 14.7456 | 460800 | 0 | 0 |
| 6.144 | 192000 | 0 | 0 | 16 | 500000 | 0 | 0 |
| 8 | 250000 | 0 | 0 | 17.2032 | 537600 | 0 | 0 |
| 9.8304 | 307200 | 0 | 0 | 18 | 562500 | 0 | 0 |
| 10 | 312500 | 0 | 0 | 20 | 625000 | 0 | 0 |

**Table 17.5 Examples of BRR Settings for Various Bit Rates (Clock Synchronous Mode)**

| Bit Rate (bit/s) | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4 | | 8 | | 10 | | 16 | | 18 | | 20 | |
| | n | N | n | N | n | N | n | N | n | N | n | N |
| 110 | — | — | — | — | — | — | | | — | — | — | — |
| 250 | 2 | 249 | 3 | 124 | — | — | 3 | 249 | — | — | — | — |
| 500 | 2 | 124 | 2 | 249 | — | — | 3 | 124 | 3 | 140 | 3 | 155 |
| 1k | 1 | 249 | 2 | 124 | — | — | 2 | 249 | 3 | 69 | 3 | 77 |
| 2.5k | 1 | 99 | 1 | 199 | 1 | 249 | 2 | 99 | 2 | 112 | 2 | 124 |
| 5k | 0 | 199 | 1 | 99 | 1 | 124 | 1 | 199 | 1 | 224 | 1 | 249 |
| 10k | 0 | 99 | 0 | 199 | 0 | 249 | 1 | 99 | 1 | 112 | 1 | 124 |
| 25k | 0 | 39 | 0 | 79 | 0 | 99 | 0 | 159 | 0 | 179 | 0 | 199 |
| 50k | 0 | 19 | 0 | 39 | 0 | 49 | 0 | 79 | 0 | 89 | 0 | 99 |
| 100k | 0 | 9 | 0 | 19 | 0 | 24 | 0 | 39 | 0 | 44 | 0 | 49 |
| 250k | 0 | 3 | 0 | 7 | 0 | 9 | 0 | 15 | 0 | 17 | 0 | 19 |
| 500k | 0 | 1 | 0 | 3 | 0 | 4 | 0 | 7 | 0 | 8 | 0 | 9 |
| 1M | 0 | 0* | 0 | 1 | — | — | 0 | 3 | 0 | 4 | 0 | 4 |
| 2M | | | 0 | 0* | — | — | 0 | 1 | — | — | — | — |
| 2.5M | | | | | 0 | 0* | — | — | — | — | 0 | 1 |
| 4M | | | | | | | 0 | 0* | — | — | — | — |

[Legend]

Blank: No setting is available.

—: A setting is available but error occurs.

*: Continuous transfer is not possible.

## 17.4 Operation in Asynchronous Mode

Figure 17.2 shows the general format for asynchronous serial communication. One character (or frame) consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally stop bits (high level). Inside the SCI3, the transmitter and receiver are independent units, enabling full-duplex. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

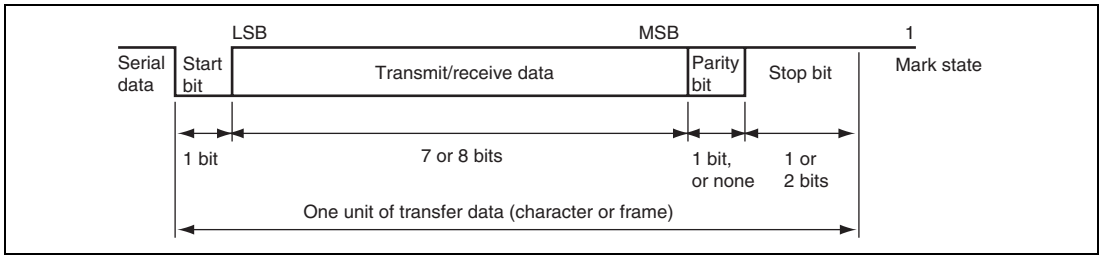



**Figure 17.2 Data Format in Asynchronous Communication**

### 17.4.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK3 pin can be selected as the SCI3's serial clock, according to the setting of the COM bit in SMR and the CKE0 and CKE1 bits in SCR3. When an external clock is input at the SCK3 pin, the clock frequency should be 16 times the bit rate used.

When the SCI3 is operated on an internal clock, the clock can be output from the SCK3 pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 17.3.

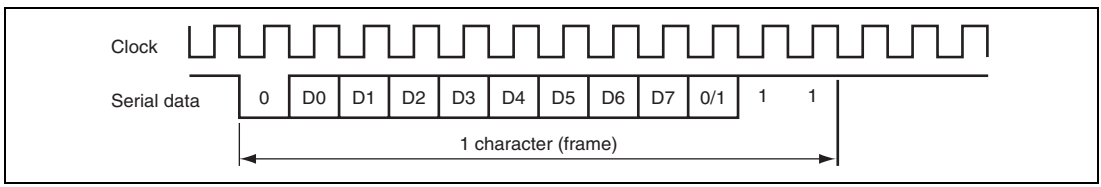
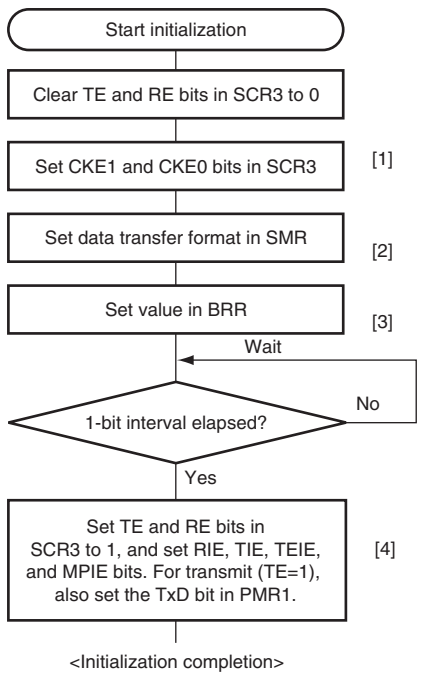



**Figure 17.3 Relationship between Output Clock and Transfer Data Phase (Asynchronous Mode)(Example with 8-Bit Data, Parity, Two Stop Bits)**

### 17.4.2 SCI3 Initialization

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR3 to 0, then initialize the SCI3 as described below. When the operating mode, or transfer format, is changed for example, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and OER flags, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.




[1] Set the clock selection in SCR3. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock output is selected in asynchronous mode, clock is output immediately after CKE1 and CKE0 settings are made. When the clock output is selected at reception in clocked synchronous mode, clock is output immediately after CKE1, CKE0, and RE are set to 1.

[2] Set the data transfer format in SMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR3 to 1. RE settings enable the RXD pin to be used. For transmission, set the TXD bit in PMR1 to 1 to enable the TXD output pin to be used. Also set the RIE, TIE, TEIE, and MPIE bits, depending on whether interrupts are required. In asynchronous mode, the output pin level is in the mark state for transmission and the input in level is in the idle state for reception. SCI3 waites for a start bit in the idle state. SCI3 is ready for transmission after 1 is output for a signal frame.

**Figure 17.4 Sample SCI3 Initialization Flowchart**

### 17.4.3 Data Transmission

Figure 17.5 shows an example of operation for transmission in asynchronous mode. In transmission, the SCI3 operates as described below.

1. The SCI3 monitors the TDRE flag in SSR. If the flag is cleared to 0, the SCI3 recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is possible because the TXI interrupt routine writes next transmit data to TDR before transmission of the current transmit data has been completed.
3. The SCI3 checks the TDRE flag at the timing for sending the stop bit.
4. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.
5. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the “mark state” is entered, in which 1 is output. If the TEIE bit in SCR3 is set to 1 at this time, a TEI interrupt request is generated.
6. Figure 17.6 shows a sample flowchart for transmission in asynchronous mode.

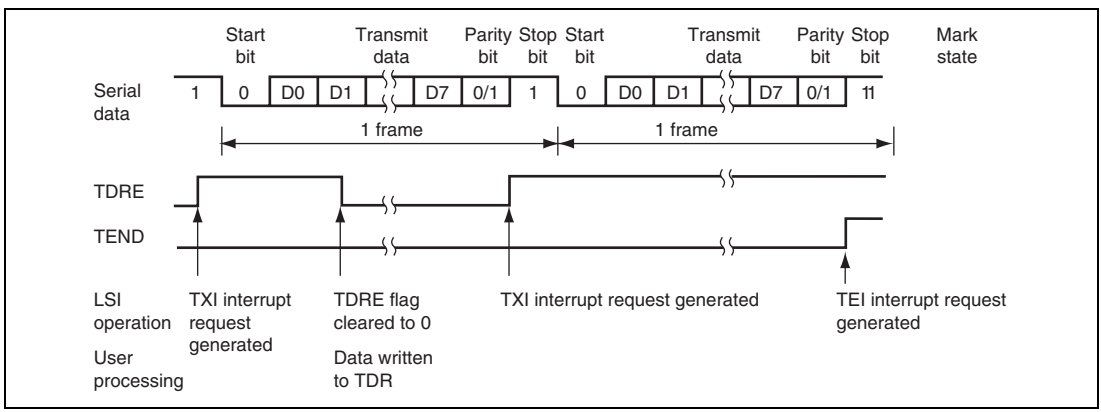



**Figure 17.5 Example of SCI3 Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)**

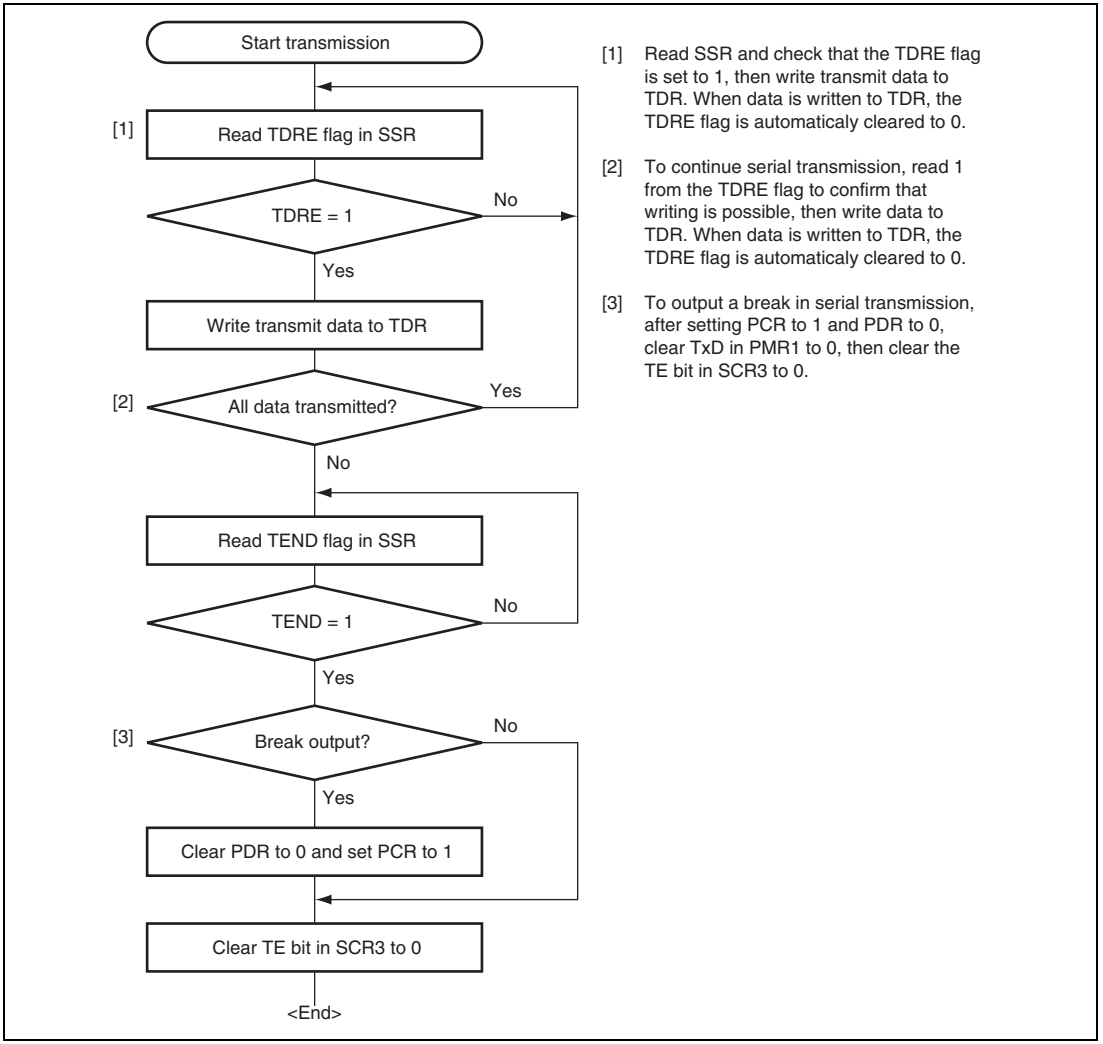



**Figure 17.6 Sample Serial Transmission Data Flowchart (Asynchronous Mode)**

### 17.4.4 Serial Data Reception

Figure 17.7 shows an example of operation for reception in asynchronous mode. In serial reception, the SCI3 operates as described below.

1. The SCI3 monitors the communication line. If a start bit is detected, the SCI3 performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error occurs (when reception of the next data is completed while the RDRF flag is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error is detected (when the stop bit is 0), the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated.
5. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt request is generated. Continuous reception is possible because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has been completed.

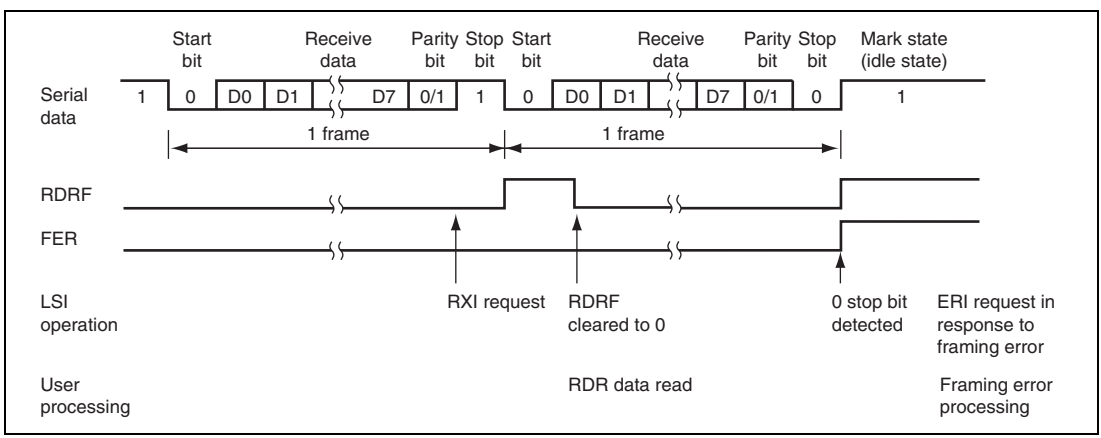


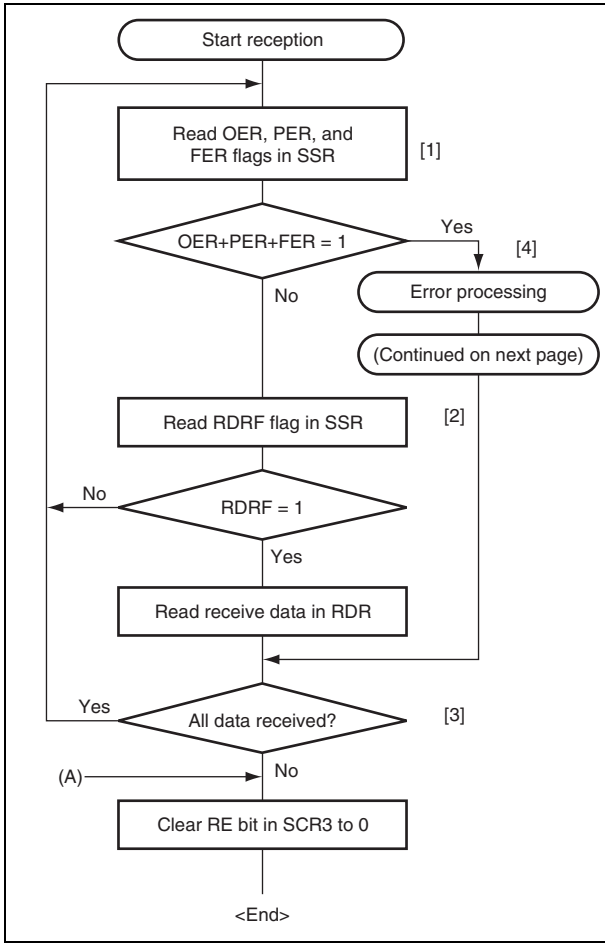

**Figure 17.7 Example of SCI3 Reception in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)**

Table 17.6 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 17.8 shows a sample flow chart for serial data reception.

**Table 17.6 SSR Status Flags and Receive Data Handling**

| | SSR Status Flag | | | | |
|---|---|---|---|---|---|
| **RDRF*** | **OER** | **FER** | **PER** | **Receive Data** | **Receive Error Type** |
| 1 | 1 | 0 | 0 | Lost | Overrun error |
| 0 | 0 | 1 | 0 | Transferred to RDR | Framing error |
| 0 | 0 | 0 | 1 | Transferred to RDR | Parity error |
| 1 | 1 | 1 | 0 | Lost | Overrun error + framing error |
| 1 | 1 | 0 | 1 | Lost | Overrun error + parity error |
| 0 | 0 | 1 | 1 | Transferred to RDR | Framing error + parity error |
| 1 | 1 | 1 | 1 | Lost | Overrun error + framing error + parity error |

Note: * The RDRF flag retains the state it had before data reception.




[1] Read the OER, PER, and FER flags in SSR to identify the error. If a receive error occurs, performs the appropriate error processing.

[2] Read SSR and check that RDRF = 1, then read the receive data in RDR. The RDRF flag is cleared automatically.

[3] To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag and read RDR.
The RDRF flag is cleared automatically.

[4] If a receive error occurs, read the OER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the OER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

**Figure 17.8 Sample Serial Reception Data Flowchart (Asynchronous Mode) (1)**

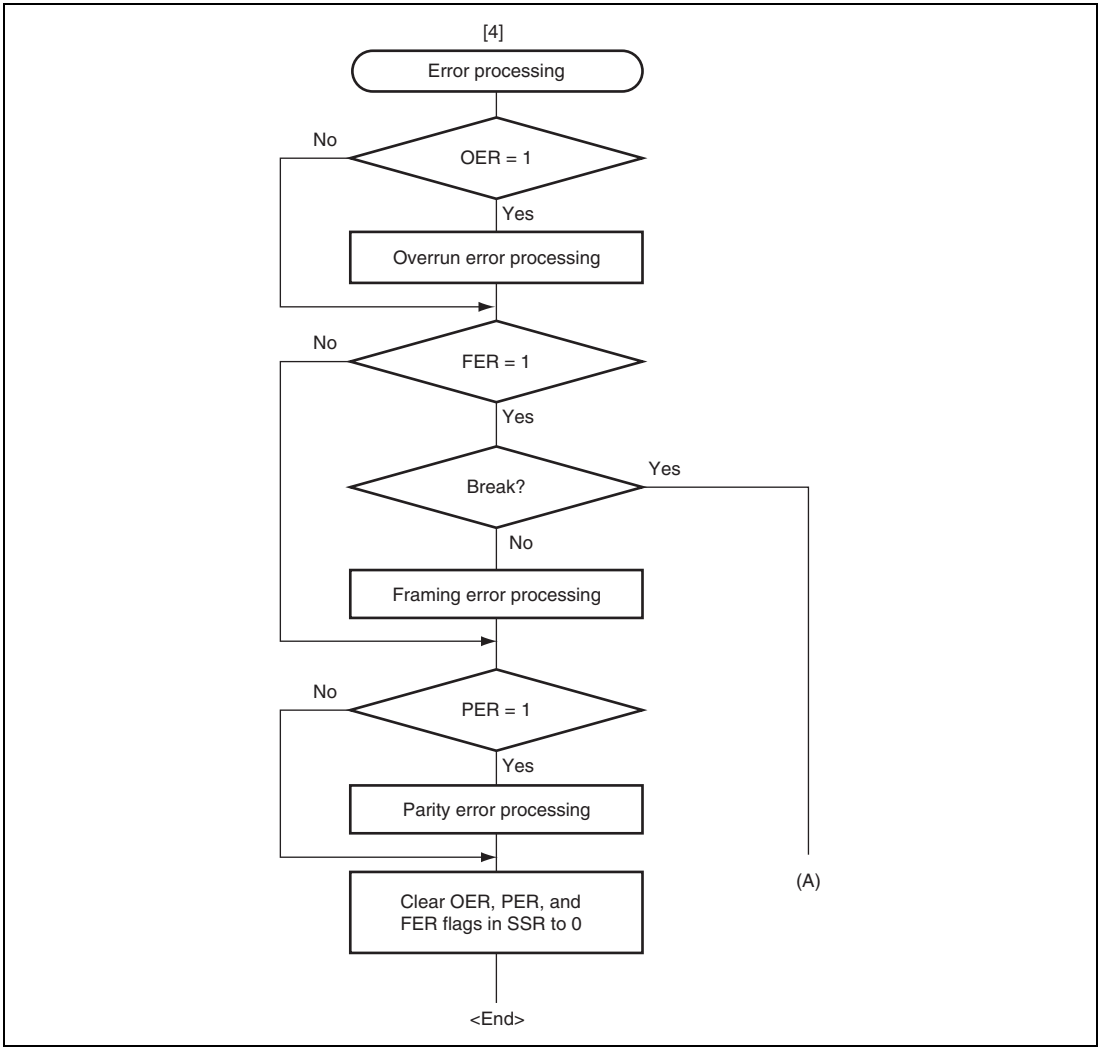



**Figure 17.8 Sample Serial Reception Data Flowchart (Asynchronous Mode) (2)**

## 17.5 Operation in Clock Synchronous Mode

Figure 17.9 shows the general format for clock synchronous communication. In clock synchronous mode, data is transmitted or received synchronous with clock pulses. A single character in the transmit data consists of the 8-bit data starting from the LSB. In clock synchronous serial communication, data on the transmission line is output from one falling edge of the synchronization clock to the next. In clock synchronous mode, the SCI3 receives data in synchronous with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the MSB state. In clock synchronous mode, no parity or multiprocessor bit is added. Inside the SCI3, the transmitter and receiver are independent units, enabling full-duplex communication through the use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so data can be read or written during transmission or reception, enabling continuous data transfer.

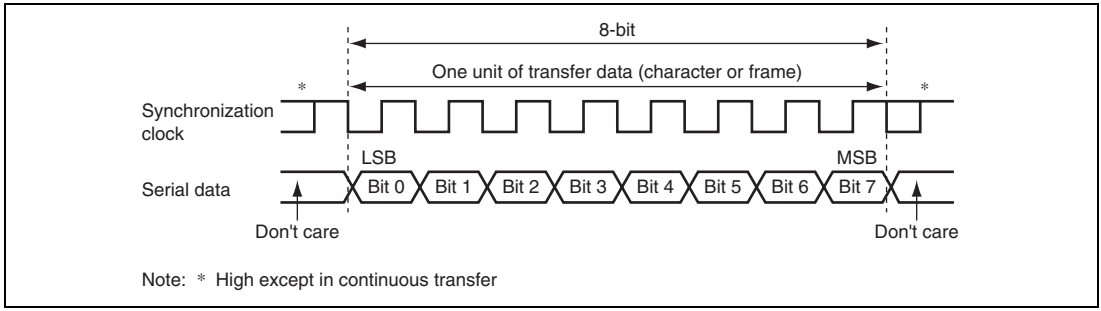



**Figure 17.9 Data Format in Clock Synchronous Communication**

### 17.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK3 pin can be selected, according to the setting of the COM bit in SMR and CKE0 and CKE1 bits in SCR3. When the SCI3 is operated on an internal clock, the synchronization clock is output from the SCK3 pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

### 17.5.2 SCI3 Initialization

Before transmitting and receiving data, the SCI3 should be initialized as described in a sample flowchart in figure 17.4.

### 17.5.3 Serial Data Transmission

Figure 17.10 shows an example of SCI3 operation for transmission in clock synchronous mode. In serial transmission, the SCI3 operates as described below.

1. The SCI3 monitors the TDRE flag in SSR, and if the flag is 0, the SCI3 recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. The SCI3 sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR3 is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.
3. 8-bit data is sent from the TXD pin synchronized with the output clock when output clock mode has been specified, and synchronized with the input clock when use of an external clock has been specified. Serial data is transmitted sequentially from the LSB (bit 0), from the TXD pin.
4. The SCI3 checks the TDRE flag at the timing for sending the MSB (bit 7).
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TDRE flag maintains the output state of the last bit. If the TEIE bit in SCR3 is set to 1 at this time, a TEI interrupt request is generated.
7. The SCK3 pin is fixed high at the end of transmission.

Figure 17.11 shows a sample flow chart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (OER, FER, or PER) is set to 1. Make sure that the receive error flags are cleared to 0 before starting transmission.

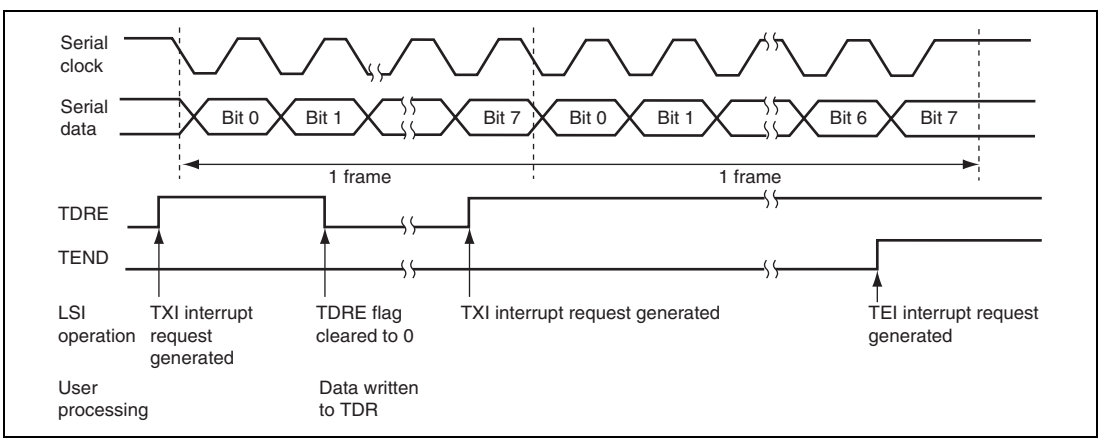
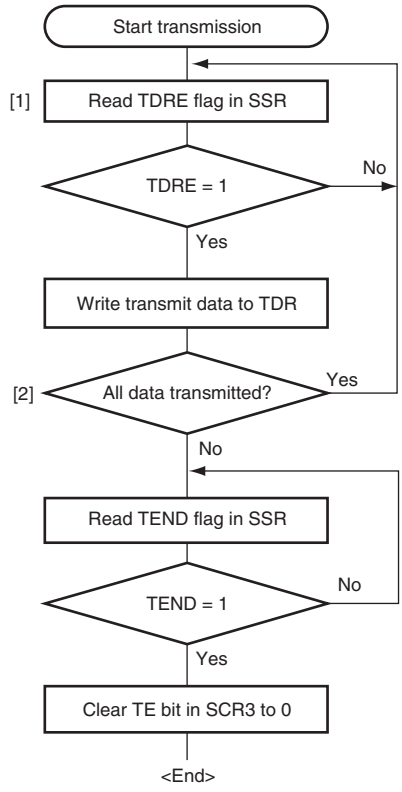



**Figure 17.10 Example of SCI3 Transmission in Clock Synchronous Mode**




[1] Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. When data is written to TDR, the TDRE flag is automatically cleared to 0 and clocks are output to start the data transmission.

[2] To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR. When data is written to TDR, the TDRE flag is automatically cleared to 0.

**Figure 17.11 Sample Serial Transmission Flowchart (Clock Synchronous Mode)**

### 17.5.4 Serial Data Reception (Clock Synchronous Mode)

Figure 17.12 shows an example of SCI3 operation for reception in clock synchronous mode. In serial reception, the SCI3 operates as described below.

1. The SCI3 performs internal initialization synchronous with a synchronization clock input or output, starts receiving data.
2. The SCI3 stores the receive data in RSR.
3. If an overrun error occurs (when reception of the next data is completed while the RDRF flag in SSR is still set to 1), the OER bit in SSR is set to 1. If the RIE bit in SCR3 is set to 1 at this time, an ERI interrupt request is generated, receive data is not transferred to RDR, and the RDRF flag remains to be set to 1.
4. If reception is completed successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt request is generated.

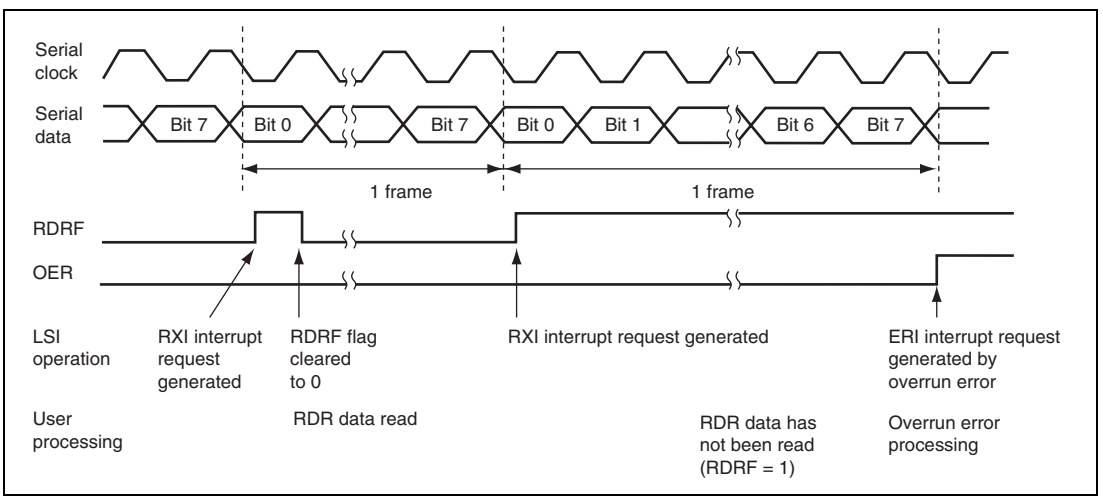



**Figure 17.12 Example of SCI3 Reception in Clock Synchronous Mode**

Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the OER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 17.13 shows a sample flow chart for serial data reception.

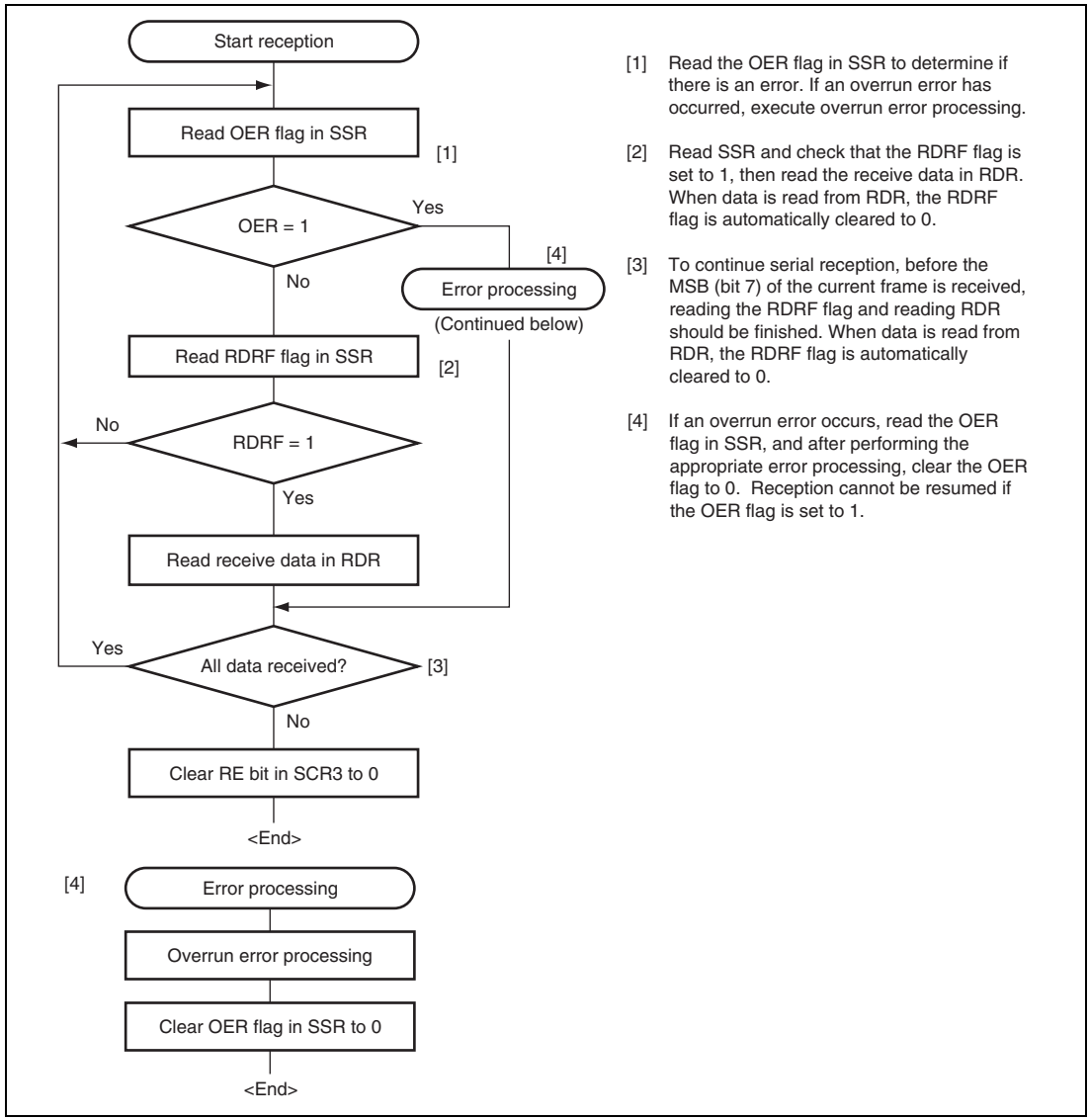
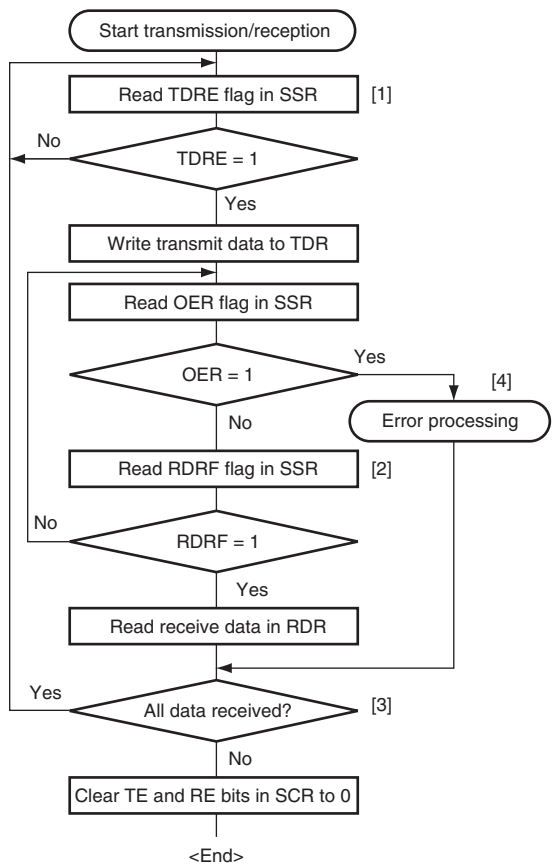



**Figure 17.13 Sample Serial Reception Flowchart (Clock Synchronous Mode)**

### 17.5.5 Simultaneous Serial Data Transmission and Reception

Figure 17.14 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI3 has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI3 has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (OER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.




[1] Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR.
When data is written to TDR, the TDRE flag is automatically cleared to 0.

[2] Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR.
When data is read from RDR, the RDRF flag is automatically cleared to 0.

[3] To continue serial transmission/ reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR.
When data is written to TDR, the TDRE flag is automatically cleared to 0. When data is read from RDR, the RDRF flag is automatically cleared to 0.

[4] If an overrun error occurs, read the OER flag in SSR, and after performing the appropriate error processing, clear the OER flag to 0. Transmission/reception cannot be resumed if the OER flag is set to 1.
For overrun error processing, see figure 17.13.

**Figure 17.14 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations (Clock Synchronous Mode)**

## 17.6 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer between a number of processors sharing communication lines by asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is performed, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles; an ID transmission cycle that specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle; if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 17.15 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose IDs do not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI3 uses the MPIE bit in SCR3 to implement this function. When the MPIE bit is set to 1, transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status flags, RDRF, FER, and OER, to 1, are inhibited until data with a 1 multiprocessor bit is received. On reception of a receive character with a 1 multiprocessor bit, the MPBR bit in SSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCR3 is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is rendered invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

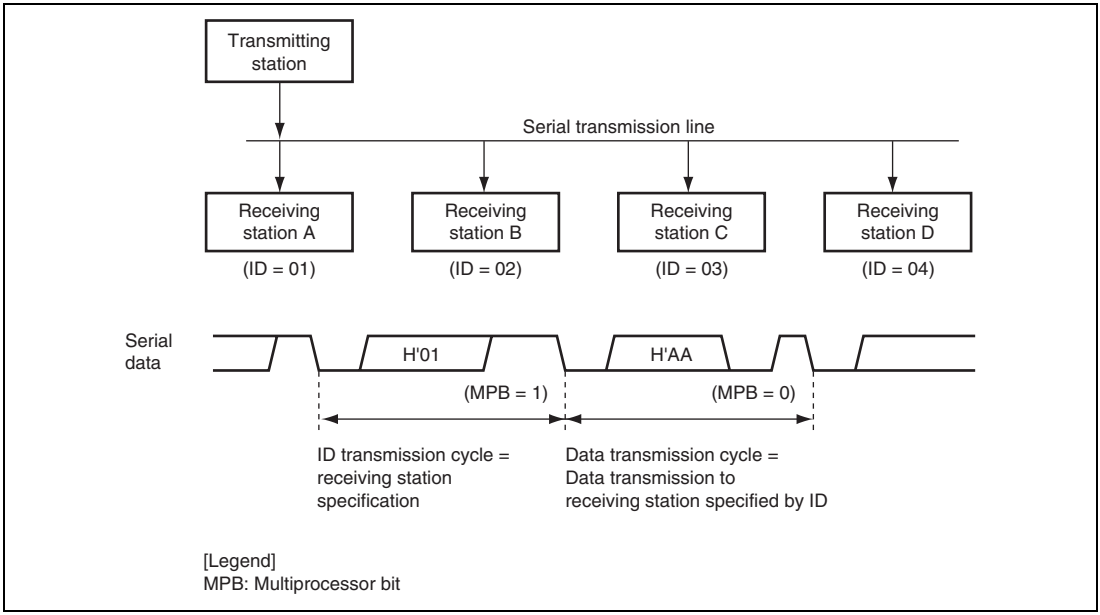



**Figure 17.15 Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)**

### 17.6.1 Multiprocessor Serial Data Transmission

Figure 17.16 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI3 operations are the same as those in asynchronous mode.

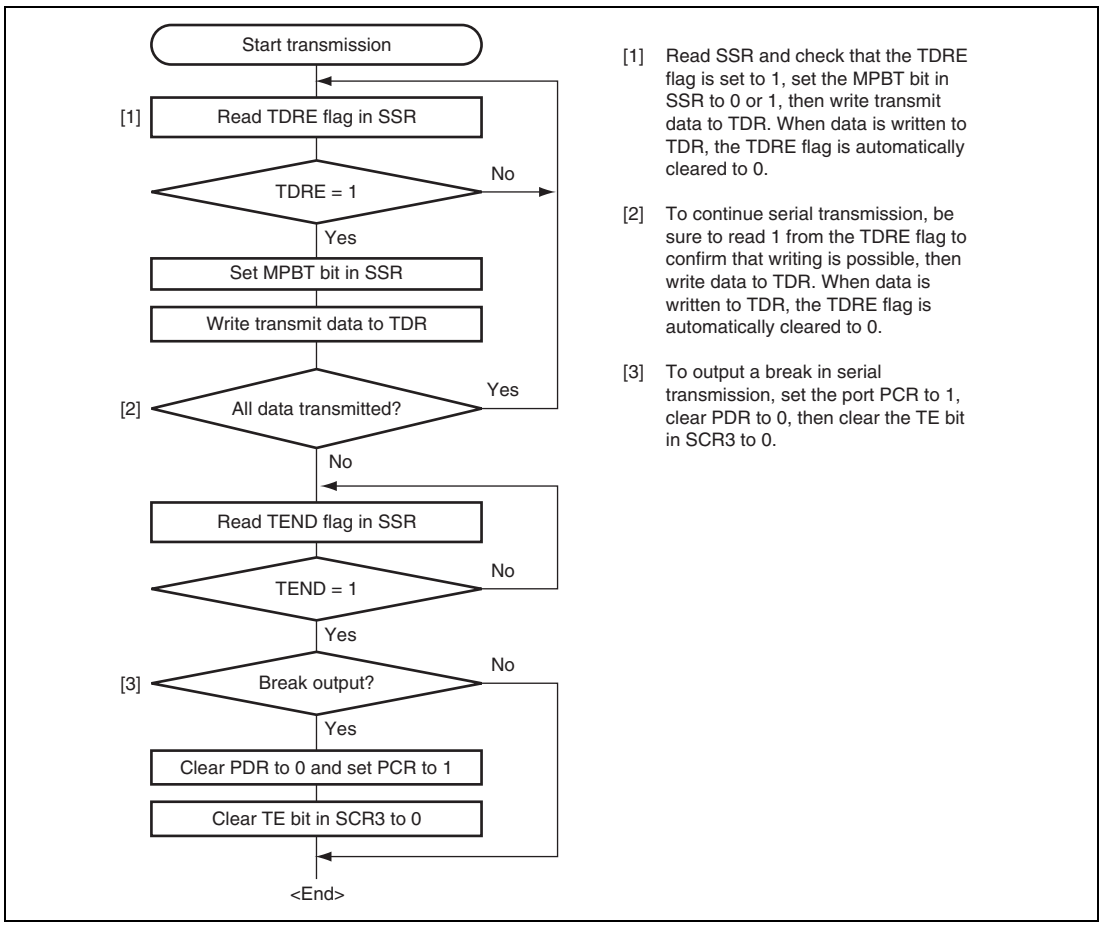



**Figure 17.16 Sample Multiprocessor Serial Transmission Flowchart**

### 17.6.2 Multiprocessor Serial Data Reception

Figure 17.17 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR3 is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI3 operations are the same as those in asynchronous mode. Figure 17.18 shows an example of SCI3 operation for multiprocessor format reception.

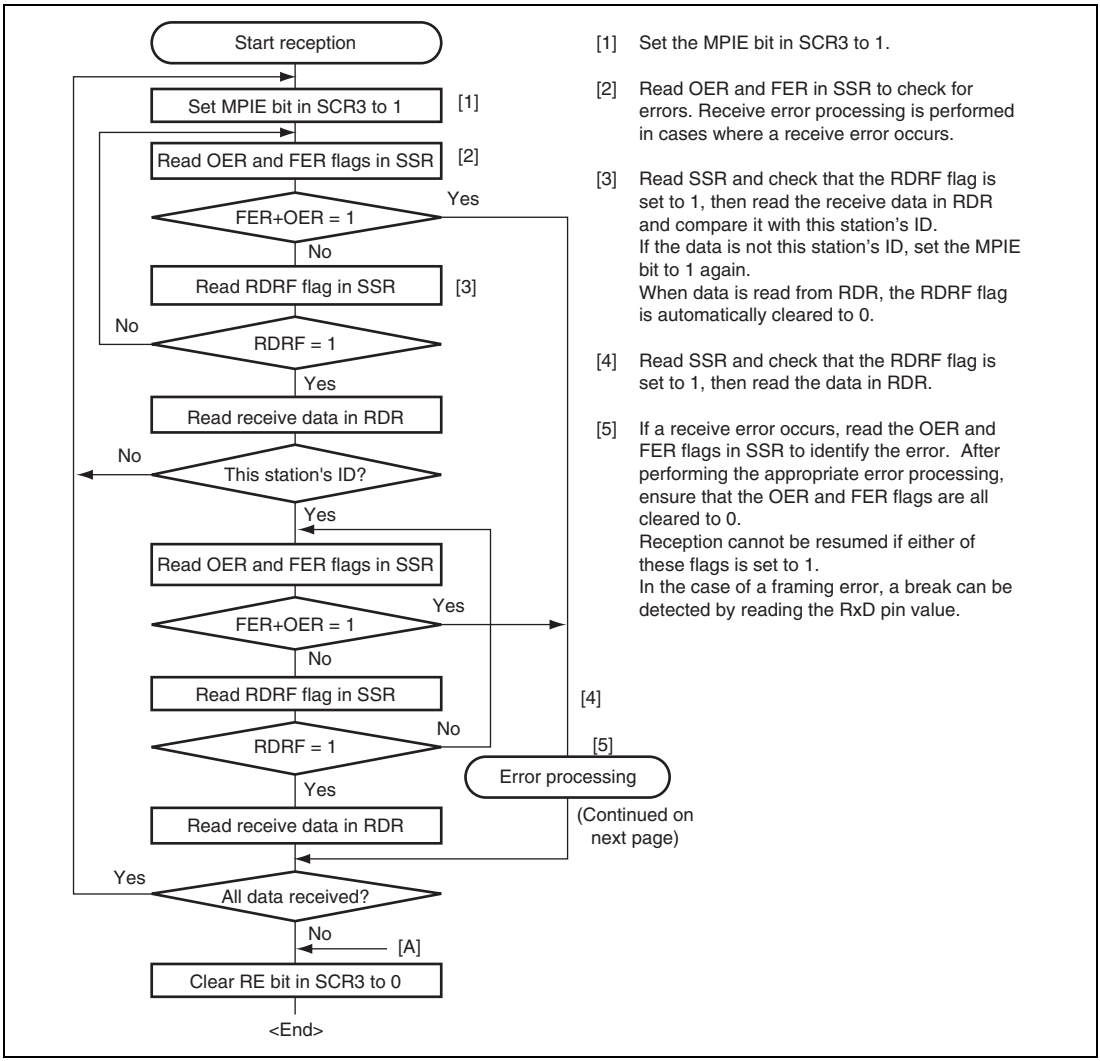



**Figure 17.17 Sample Multiprocessor Serial Reception Flowchart (1)**

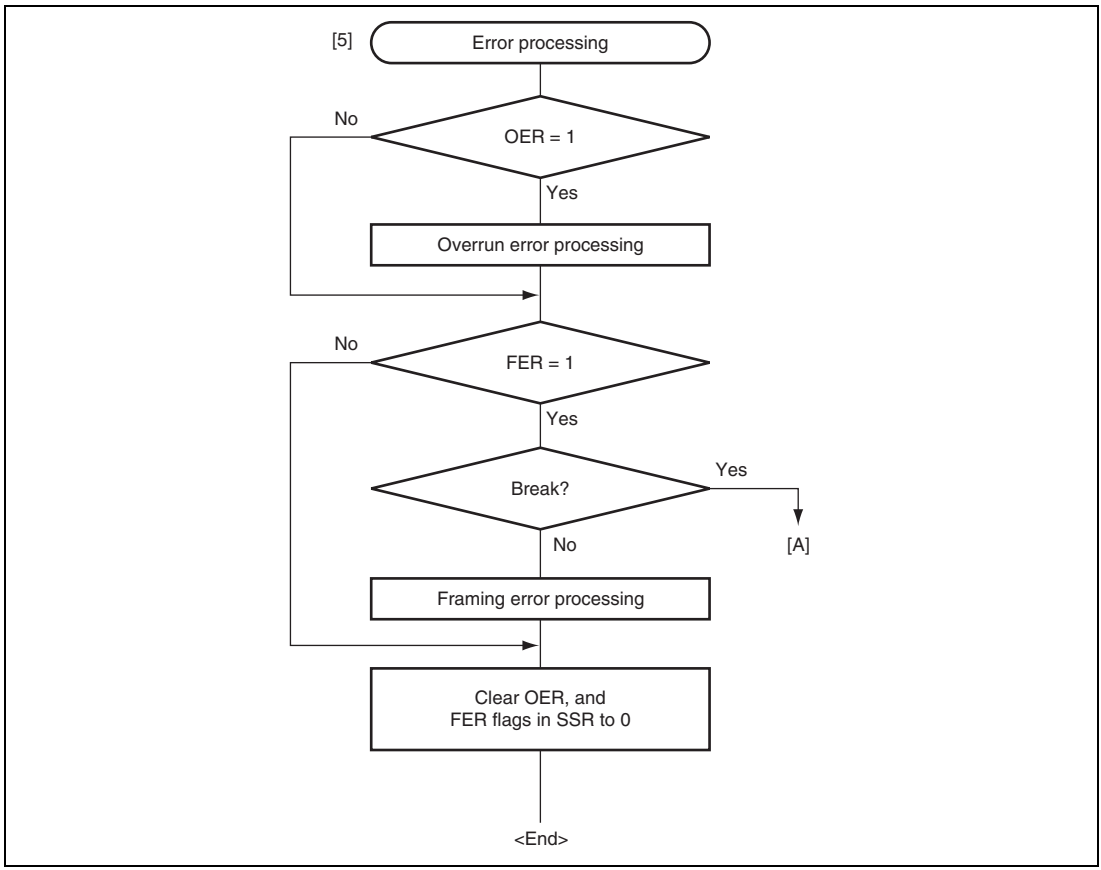



**Figure 17.17 Sample Multiprocessor Serial Reception Flowchart (2)**

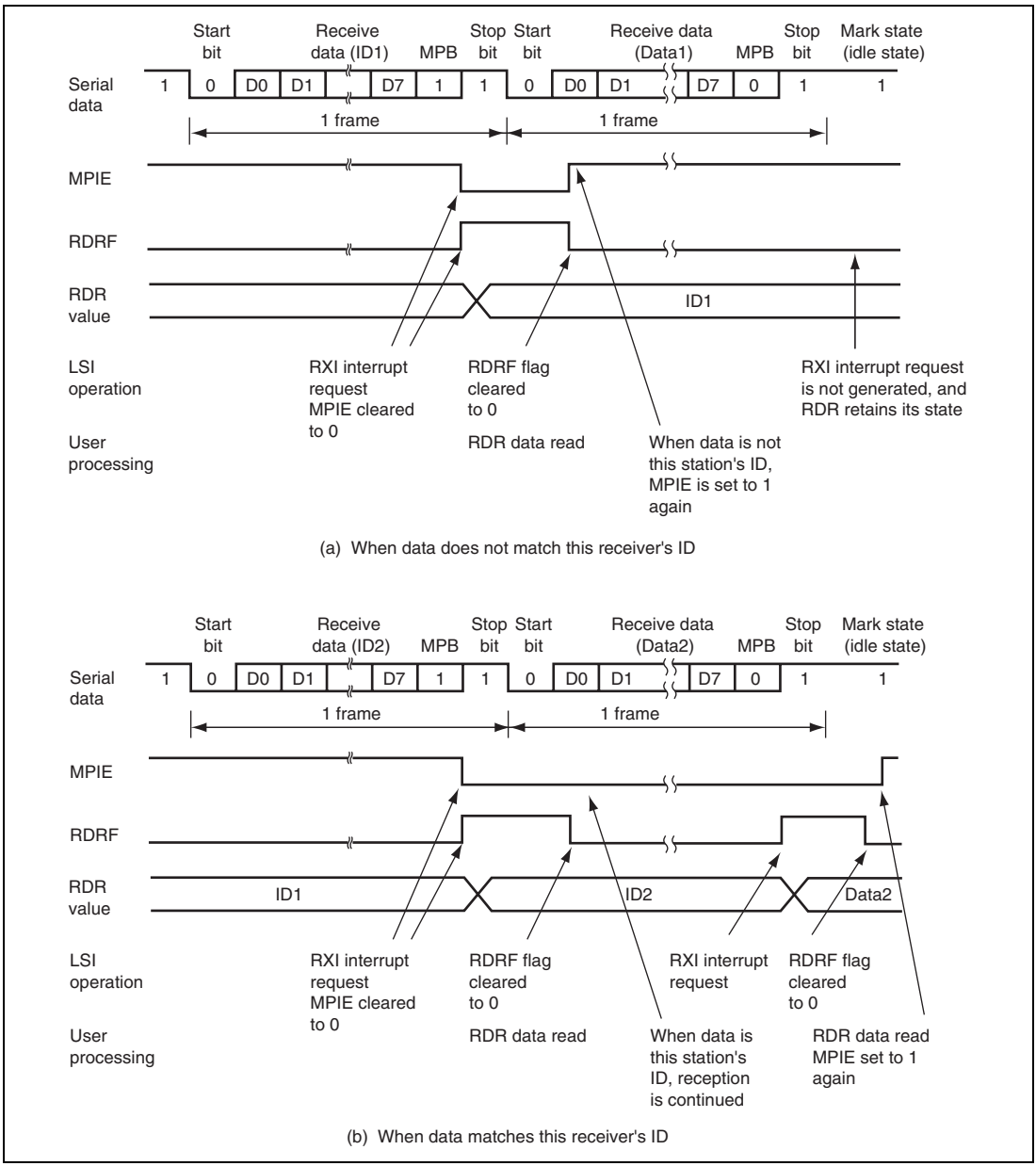



**Figure 17.18 Example of SCI3 Reception Using Multiprocessor Format (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)**

## 17.7 Interrupt Requests

SCI3 creates the following six interrupt requests: transmission end, transmit data empty, receive data full, and receive errors (overrun error, framing error, and parity error). Table 17.7 shows the interrupt sources.

**Table 17.7 SCI3 Interrupt Requests**

| Interrupt Requests | Abbreviation | Interrupt Sources |
|---|---|---|
| Receive Data Full | RXI | Setting RDRF in SSR |
| Transmit Data Empty | TXI | Setting TDRE in SSR |
| Transmission End | TEI | Setting TEND in SSR |
| Receive Error | ERI | Setting OER, FER, and PER in SSR |

The initial value of the TDRE flag in SSR is 1. Thus, when the TIE bit in SCR3 is set to 1 before transferring the transmit data to TDR, a TXI interrupt request is generated even if the transmit data is not ready. The initial value of the TEND flag in SSR is 1. Thus, when the TEIE bit in SCR3 is set to 1 before transferring the transmit data to TDR, a TEI interrupt request is generated even if the transmit data has not been sent. It is possible to make use of the most of these interrupt requests efficiently by transferring the transmit data to TDR in the interrupt routine. To prevent the generation of these interrupt requests (TXI and TEI), set the enable bits (TIE and TEIE) that correspond to these interrupt requests to 1, after transferring the transmit data to TDR.

## 17.8 Usage Notes

### 17.8.1 Break Detection and Processing

When framing error detection is performed, a break can be detected by reading the RXD pin value directly. In a break, the input from the RXD pin becomes all 0s, setting the FER flag, and possibly the PER flag. Note that as the SCI3 continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

### 17.8.2 Mark State and Break Sending

When the TXD or TXD2 bit in PMR1 or the TXD_3 bit in SMCR is 0, the TXD pin is used as an I/O port whose direction (input or output) and level are determined by PCR and PDR. This can be used to set the TXD pin to mark state (high level) or send a break during serial data transmission. To maintain the communication line at mark state until TE is set to 1, set both PCR and PDR to 1 and also cleared the TXD bit to 0. Then, the TXD pin becomes an I/O port, and 1 is output from the TXD pin. To send a break during serial transmission, first set PCR to 1 and clear PDR to 0, and then clear the TXD bit to 0. At this time, regardless of the current transmission state, the TXD pin becomes an I/O port, and 0 is output from the TXD pin.

### 17.8.3 Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only)

Transmission cannot be started when a receive error flag (OER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

### 17.8.4 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI3 operates on a basic clock with a frequency of 16 times the transfer rate. In reception, the SCI3 samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the basic clock as shown in figure 17.19. Thus, the reception margin in asynchronous mode is given by formula (1) below.

$$M = \left\{ (0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} - (L - 0.5)\,F \right\} \times 100(\%)$$

... Formula (1)

[Legend]

N: Ratio of bit rate to clock (N = 16)

D: Clock duty (D = 0.5 to 1.0)

L: Frame length (L = 9 to 12)

F: Absolute value of clock rate deviation

Assuming values of F (absolute value of clock rate deviation) = 0 and D (clock duty) = 0.5 in formula (1), the reception margin can be given by the formula.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100\ [\%] = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed for in system design.

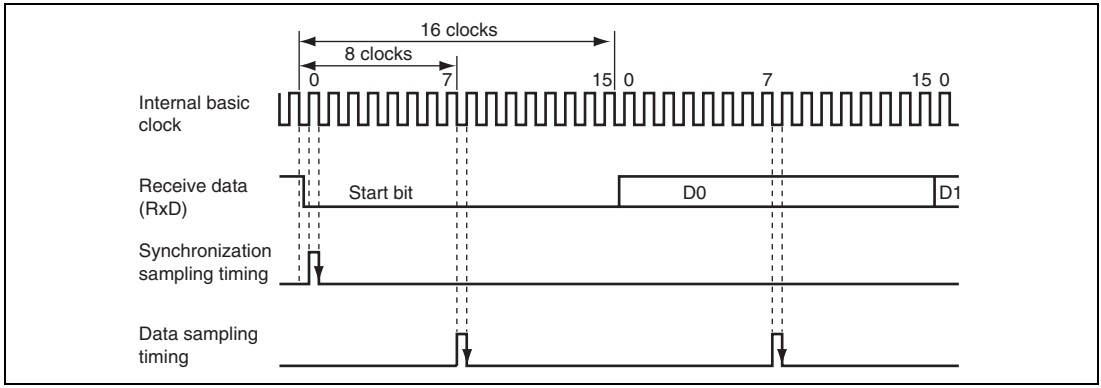



**Figure 17.19 Receive Data Sampling Timing in Asynchronous Mode**

# Section 18 I²C Bus Interface 2 (IIC2)

The I²C bus interface 2 conforms to and provides a subset of the Philips I²C bus (inter-IC bus) interface functions. The register configuration that controls the I²C bus differs partly from the Philips configuration, however.

Figure 18.1 shows a block diagram of the I²C bus interface 2.

Figure 18.2 shows an example of I/O pin connections to external circuits.

## 18.1 Features

- Selection of I²C format or clocked synchronous serial format
- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.

I²C bus format:

- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically.

  If transmission/reception is not yet possible, set the SCL to low until preparations are completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full (including slave-address match), arbitration lost, NACK detection, and stop condition detection
- Direct bus drive

  Two pins, SCL and SDA pins, function as NMOS open-drain outputs when the bus drive function is selected.

Clocked synchronous format:

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error

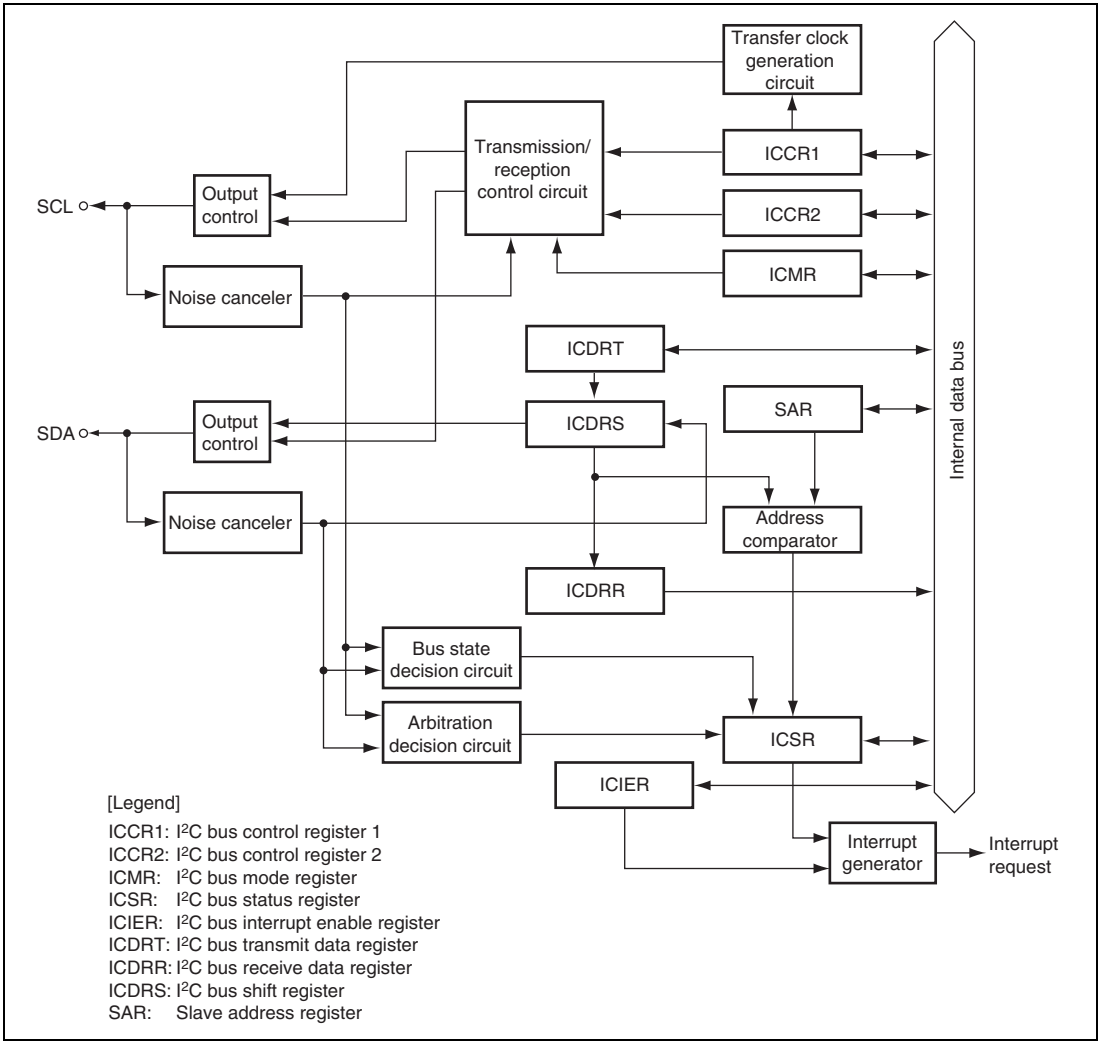



**Figure 18.1 Block Diagram of I²C Bus Interface 2**

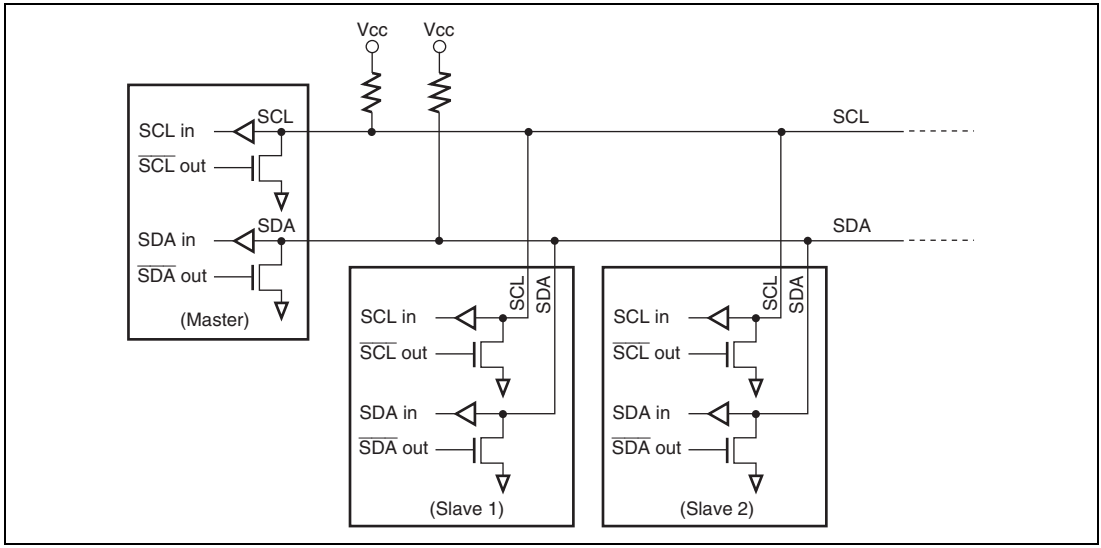



**Figure 18.2 External Circuit Connections of I/O Pins**

## 18.2 Input/Output Pins

Table 18.1 summarizes the input/output pins used by the I²C bus interface 2.

**Table 18.1 Pin Configuration**

| Name | Abbreviation | I/O | Function |
|---|---|---|---|
| Serial clock | SCL | I/O | IIC serial clock input/output |
| Serial data | SDA | I/O | IIC serial data input/output |

## 18.3 Register Descriptions

The I²C bus interface 2 has the following registers.

- I²C bus control register 1 (ICCR1)
- I²C bus control register 2 (ICCR2)
- I²C bus mode register (ICMR)
- I²C bus interrupt enable register (ICIER)
- I²C bus status register (ICSR)
- I²C bus slave address register (SAR)
- I²C bus transmit data register (ICDRT)
- I²C bus receive data register (ICDRR)
- I²C bus shift register (ICDRS)

### 18.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 enables or disables the I²C bus interface 2, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | ICE | 0 | R/W | I²C Bus Interface Enable<br>0: This module is halted. (SCL and SDA pins are set to port function.)<br>1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.) |
| 6 | RCVD | 0 | R/W | Reception Disable<br>This bit enables or disables the next operation when TRS is 0 and ICDRR is read.<br>0: Enables next reception<br>1: Disables next reception |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | MST | 0 | R/W | Master/Slave Select |
| 4 | TRS | 0 | R/W | Transmit/Receive Select<br>In master mode with the I²C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. Modification of the TRS bit should be made between transfer frames.<br>After data receive has been started in slave receive mode, when the first seven bits of the receive data agree with the slave address that is set to SAR and the eighth bit is 1, TRS is automatically set to 1. If an overrun error occurs in master mode with the clock synchronous serial format, MST is cleared to 0 and slave receive mode is entered.<br>Operating modes are described below according to MST and TRS combination. When clocked synchronous serial format is selected and MST is 1, clock is output.<br>00: Slave receive mode<br>01: Slave transmit mode<br>10: Master receive mode<br>11: Master transmit mode |
| 3 | CKS3 | 0 | R/W | Transfer Clock Select 3 to 0<br>These bits should be set according to the necessary transfer rate (see table 18.2) in master mode. In slave mode, these bits are used for reservation of the setup time in transmit mode. The time is 10 $t_{cyc}$ when CKS3 = 0 and 20 $t_{cyc}$ when CKS3 = 1. |
| 2 | CKS2 | 0 | R/W | |
| 1 | CKS1 | 0 | R/W | |
| 0 | CKS0 | 0 | R/W | |

**Table 18.2 Transfer Rate**

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | | Transfer Rate | | | | |
|---|---|---|---|---|---|---|---|---|---|
| CKS3 | CKS2 | CKS1 | CKS0 | Clock | φ=5 MHz | φ=8 MHz | φ=10 MHz | φ=16 MHz | φ=20 MHz |
| 0 | 0 | 0 | 0 | φ/28 | 179 kHz | 286 kHz | 357 kHz | 571 kHz | 714 kHz |
| | | | 1 | φ/40 | 125 kHz | 200 kHz | 250 kHz | 400 kHz | 500 kHz |
| | | 1 | 0 | φ/48 | 104 kHz | 167 kHz | 208 kHz | 333 kHz | 417 kHz |
| | | | 1 | φ/64 | 78.1 kHz | 125 kHz | 156 kHz | 250 kHz | 313 kHz |
| | 1 | 0 | 0 | φ/80 | 62.5 kHz | 100 kHz | 125 kHz | 200 kHz | 250 kHz |
| | | | 1 | φ/100 | 50.0 kHz | 80.0 kHz | 100 kHz | 160 kHz | 200 kHz |
| | | 1 | 0 | φ/112 | 44.6 kHz | 71.4 kHz | 89.3 kHz | 143 kHz | 179 kHz |
| | | | 1 | φ/128 | 39.1 kHz | 62.5 kHz | 78.1 kHz | 125 kHz | 156 kHz |
| 1 | 0 | 0 | 0 | φ/56 | 89.3 kHz | 143 kHz | 179 kHz | 286 kHz | 357 kHz |
| | | | 1 | φ/80 | 62.5 kHz | 100 kHz | 125 kHz | 200 kHz | 250 kHz |
| | | 1 | 0 | φ/96 | 52.1 kHz | 83.3 kHz | 104 kHz | 167 kHz | 208 kHz |
| | | | 1 | φ/128 | 39.1 kHz | 62.5 kHz | 78.1 kHz | 125 kHz | 156 kHz |
| | 1 | 0 | 0 | φ/160 | 31.3 kHz | 50.0 kHz | 62.5 kHz | 100 kHz | 125 kHz |
| | | | 1 | φ/200 | 25.0 kHz | 40.0 kHz | 50.0 kHz | 80.0 kHz | 100 kHz |
| | | 1 | 0 | φ/224 | 22.3 kHz | 35.7 kHz | 44.6 kHz | 71.4 kHz | 89.3 kHz |
| | | | 1 | φ/256 | 19.5 kHz | 31.3 kHz | 39.1 kHz | 62.5 kHz | 78.1 kHz |

### 18.3.2 I²C Bus Control Register 2 (ICCR2)

ICCR2 issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in the control part of the I²C bus interface 2.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | BBSY | 0 | R/W | Bus Busy<br>This bit enables to confirm whether the I²C bus is occupied or released and to issue start/stop conditions in master mode. With the clocked synchronous serial format, this bit has no meaning. With the I²C bus format, this bit is set to 1 when the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Also follow this procedure when a repeated start condition is issued. Write 0 in BBSY and 0 in SCP to issue a stop condition. To issue start/stop conditions, use the MOV instruction. |
| 6 | SCP | 1 | W | Start/Stop Issue Condition Disable<br>The SCP bit controls the issue of start/stop conditions in master mode.<br>To issue a start condition, write 1 in BBSY and 0 in SCP. A repeated start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. If 1 is written, the data is not stored. |
| 5 | SDAO | 1 | R/W | SDA Output Value Control<br>This bit is used with SDAOP when modifying output level of SDA. This bit should not be manipulated during transfer.<br>0: When reading, SDA pin outputs low.<br>When writing, SDA pin is changed to output low.<br>1: When reading, SDA pin outputs high.<br>When writing, SDA pin is changed to output Hi-Z (outputs high by external pull-up resistance). |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | SDAOP | 1 | R/W | SDAO Write Protect<br>This bit controls change of output level of the SDA pin by modifying the SDAO bit. To change the output level, clear SDAO and SDAOP to 0 or set SDAO to 1 and clear SDAOP to 0 by the MOV instruction. This bit is always read as 1. |
| 3 | SCLO | 1 | R | This bit monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low. |
| 2 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 1 | IICRST | 0 | R/W | IIC Control Part Reset<br>This bit resets the control part except for I²C registers. If this bit is set to 1 when hang-up occurs because of communication failure during I²C operation, I²C control part can be reset without setting ports and initializing registers. |
| 0 | — | 1 | — | Reserved<br>This bit is always read as 1. |

### 18.3.3 I²C Bus Mode Register (ICMR)

ICMR selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the transfer bit count.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | MLS | 0 | R/W | MSB-First/LSB-First Select<br>0: MSB-first<br>1: LSB-first<br>Set this bit to 0 when the I²C bus format is used. |
| 6 | WAIT | 0 | R/W | Wait Insertion Bit<br>In master mode with the I²C bus format, this bit selects whether to insert a wait after data transfer except the acknowledge bit. When WAIT is set to 1, after the fall of the clock for the final data bit, low period is extended for two transfer clocks. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted.<br>The setting of this bit is invalid in slave mode with the I²C bus format or with the clocked synchronous serial format. |
| 5, 4 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 3 | BCWP | 1 | R/W | BC Write Protect<br>This bit controls the BC2 to BC0 modifications. When modifying BC2 to BC0, this bit should be cleared to 0 and use the MOV instruction. In clock synchronous serial mode, BC should not be modified.<br>0: When writing, values of BC2 to BC0 are set.<br>1: When reading, 1 is always read.<br>When writing, settings of BC2 to BC0 are invalid. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | BC2 | 0 | R/W | Bit Counter 2 to 0 |
| 1 | BC1 | 0 | R/W | These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. With the I²C bus format, the data is transferred with one addition acknowledge bit. Bit BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL pin is low. The value returns to 000 at the end of a data transfer, including the acknowledge bit. With the clock synchronous serial format, these bits should not be modified. |
| 0 | BC0 | 0 | R/W | |

| I²C Bus Format | Clock Synchronous Serial Format |
|---|---|
| 000: 9 bits | 000: 8 bits |
| 001: 2 bits | 001: 1 bit |
| 010: 3 bits | 010: 2 bits |
| 011: 4 bits | 011: 3 bits |
| 100: 5 bits | 100: 4 bits |
| 101: 6 bits | 101: 5 bits |
| 110: 7 bits | 110: 6 bits |
| 111: 8 bits | 111: 7 bits |

### 18.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits to be received.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TIE | 0 | R/W | Transmit Interrupt Enable<br>When the TDRE bit in ICSR is set to 1, this bit enables or disables the transmit data empty interrupt (TXI).<br>0: Transmit data empty interrupt request (TXI) is disabled.<br>1: Transmit data empty interrupt request (TXI) is enabled. |
| 6 | TEIE | 0 | R/W | Transmit End Interrupt Enable<br>This bit enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by clearing the TEND bit or the TEIE bit to 0.<br>0: Transmit end interrupt request (TEI) is disabled.<br>1: Transmit end interrupt request (TEI) is enabled. |
| 5 | RIE | 0 | R/W | Receive Interrupt Enable<br>This bit enables or disables the receive data full interrupt request (RXI) and the overrun error interrupt request (ERI) with the clocked synchronous format, when a receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0.<br>0: Receive data full interrupt request (RXI) and overrun error interrupt request (ERI) with the clocked synchronous format are disabled.<br>1: Receive data full interrupt request (RXI) and overrun error interrupt request (ERI) with the clocked synchronous format are enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | NAKIE | 0 | R/W | NACK Receive Interrupt Enable<br>This bit enables or disables the NACK receive interrupt request (NAKI) and the overrun error (setting of the OVE bit in ICSR) interrupt request (ERI) with the clocked synchronous format, when the NACKF and AL bits in ICSR are set to 1. NAKI can be canceled by clearing the NACKF, OVE, or NAKIE bit to 0.<br>0: NACK receive interrupt request (NAKI) is disabled.<br>1: NACK receive interrupt request (NAKI) is enabled. |
| 3 | STIE | 0 | R/W | Stop Condition Detection Interrupt Enable<br>0: Stop condition detection interrupt request (STPI) is disabled.<br>1: Stop condition detection interrupt request (STPI) is enabled. |
| 2 | ACKE | 0 | R/W | Acknowledge Bit Judgment Select<br>0: The value of the receive acknowledge bit is ignored, and continuous transfer is performed.<br>1: If the receive acknowledge bit is 1, continuous transfer is halted. |
| 1 | ACKBR | 0 | R | Receive Acknowledge<br>In transmit mode, this bit stores the acknowledge data that are returned by the receive device. This bit cannot be modified.<br>0: Receive acknowledge = 0<br>1: Receive acknowledge = 1 |
| 0 | ACKBT | 0 | R/W | Transmit Acknowledge<br>In receive mode, this bit specifies the bit to be sent at the acknowledge timing.<br>0: 0 is sent at the acknowledge timing.<br>1: 1 is sent at the acknowledge timing. |

### 18.3.5 I²C Bus Status Register (ICSR)

ICSR performs confirmation of interrupt request flags and status.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TDRE | 0 | R/W | Transmit Data Register Empty<br>[Setting conditions]<br>• When data is transferred from ICDRT to ICDRS and ICDRT becomes empty<br>• When TRS is set<br>• When a start condition (including re-transfer) has been issued<br>• When transmit mode is entered from receive mode in slave mode<br>[Clearing conditions]<br>• When 0 is written in TDRE after reading TDRE = 1<br>• When data is written to ICDRT with an instruction |
| 6 | TEND | 0 | R/W | Transmit End<br>[Setting conditions]<br>• When the ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1<br>• When the final bit of transmit frame is sent with the clock synchronous serial format<br>[Clearing conditions]<br>• When 0 is written in TEND after reading TEND = 1<br>• When data is written to ICDRT with an instruction |
| 5 | RDRF | 0 | R/W | Receive Data Register Full<br>[Setting condition]<br>• When a receive data is transferred from ICDRS to ICDRR<br>[Clearing conditions]<br>• When 0 is written in RDRF after reading RDRF = 1<br>• When ICDRR is read with an instruction |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | NACKF | 0 | R/W | No Acknowledge Detection Flag<br>[Setting condition]<br>• When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1<br>[Clearing condition]<br>• When 0 is written in NACKF after reading NACKF = 1 |
| 3 | STOP | 0 | R/W | Stop Condition Detection Flag<br>[Setting condition]<br>• When a stop condition is detected after frame transfer<br>[Clearing condition]<br>• When 0 is written in STOP after reading STOP = 1 |
| 2 | AL/OVE | 0 | R/W | Arbitration Lost Flag/Overrun Error Flag<br>This flag indicates that arbitration was lost in master mode with the I²C bus format and that the final bit has been received while RDRF = 1 with the clocked synchronous format.<br>When two or more master devices attempt to seize the bus at nearly the same time, if the I²C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master.<br>[Setting conditions]<br>• If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode<br>• When the SDA pin outputs high in master mode while a start condition is detected<br>• When the final bit is received with the clocked synchronous format while RDRF = 1<br>[Clearing condition]<br>• When 0 is written in AL/OVE after reading AL/OVE=1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | AAS | 0 | R/W | Slave Address Recognition Flag<br>In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR.<br>[Setting conditions]<br>• When the slave address is detected in slave receive mode<br>• When the general call address is detected in slave receive mode.<br>[Clearing condition]<br>• When 0 is written in AAS after reading AAS=1 |
| 0 | ADZ | 0 | R/W | General Call Address Recognition Flag<br>This bit is valid in I²C bus format slave receive mode.<br>[Setting condition]<br>• When the general call address is detected in slave receive mode<br>[Clearing condition]<br>• When 0 is written in ADZ after reading ADZ=1 |

### 18.3.6 Slave Address Register (SAR)

SAR selects the communication format and sets the slave address. When the chip is in slave mode with the I²C bus format, if the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | SVA6 to SVA0 | All 0 | R/W | Slave Address 6 to 0<br>These bits set a unique address in bits SVA6 to SVA0, differing form the addresses of other slave devices connected to the I²C bus. |
| 0 | FS | 0 | R/W | Format Select<br>0: I²C bus format is selected.<br>1: Clocked synchronous serial format is selected. |

### 18.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. If the MLS bit of ICMR is set to 1 and when the data is written to ICDRT, the MSB/LSB inverted data is read. The initial value of ICDRT is H'FF.

### 18.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot write to this register. The initial value of ICDRR is H'FF.

### 18.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after data of one byte is received. This register cannot be read directly from the CPU.

## 18.4 Operation

The I²C bus interface can communicate either in I²C bus mode or clocked synchronous serial mode by setting FS in SAR.

### 18.4.1 I²C Bus Format

Figure 18.3 shows the I²C bus formats. Figure 18.4 shows the I²C bus timing. The first frame following a start condition always consists of 8 bits.

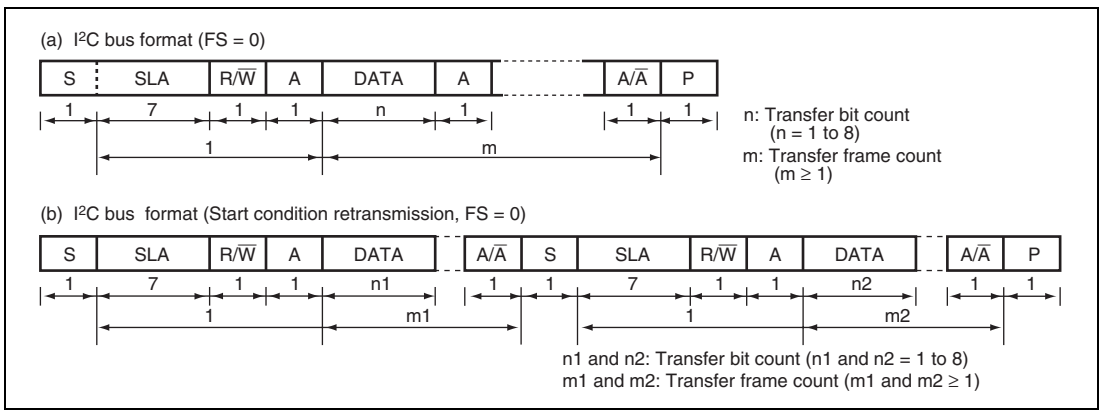



**Figure 18.3 I²C Bus Formats**

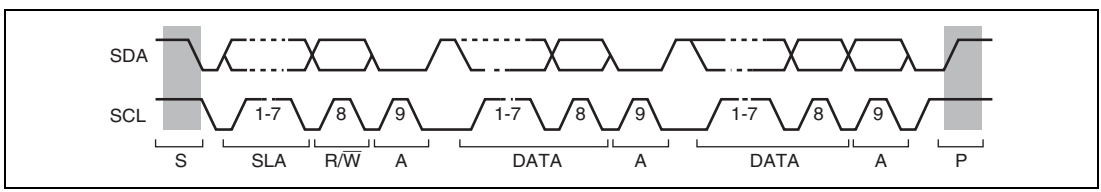



**Figure 18.4 I²C Bus Timing**

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/W: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives SDA to low.

DATA: Transfer data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

### 18.4.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For master transmit mode operation timing, see figures 18.5 and 18.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is free. Set the MST and TRS bits in ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using MOV instruction. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP using MOV instruction. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.

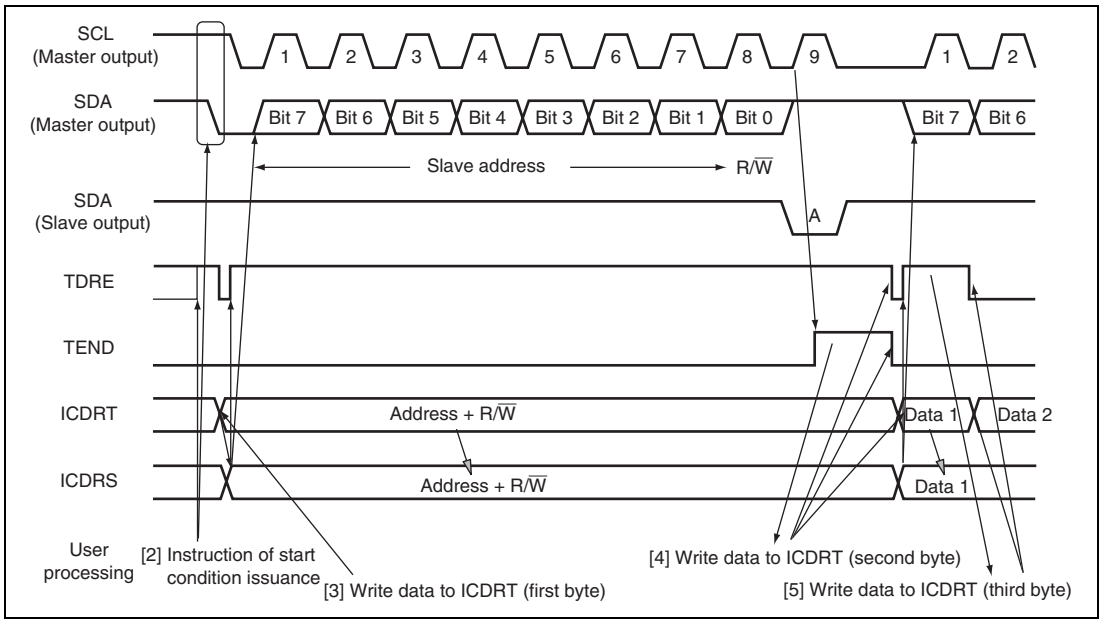



**Figure 18.5 Master Transmit Mode Operation Timing (1)**

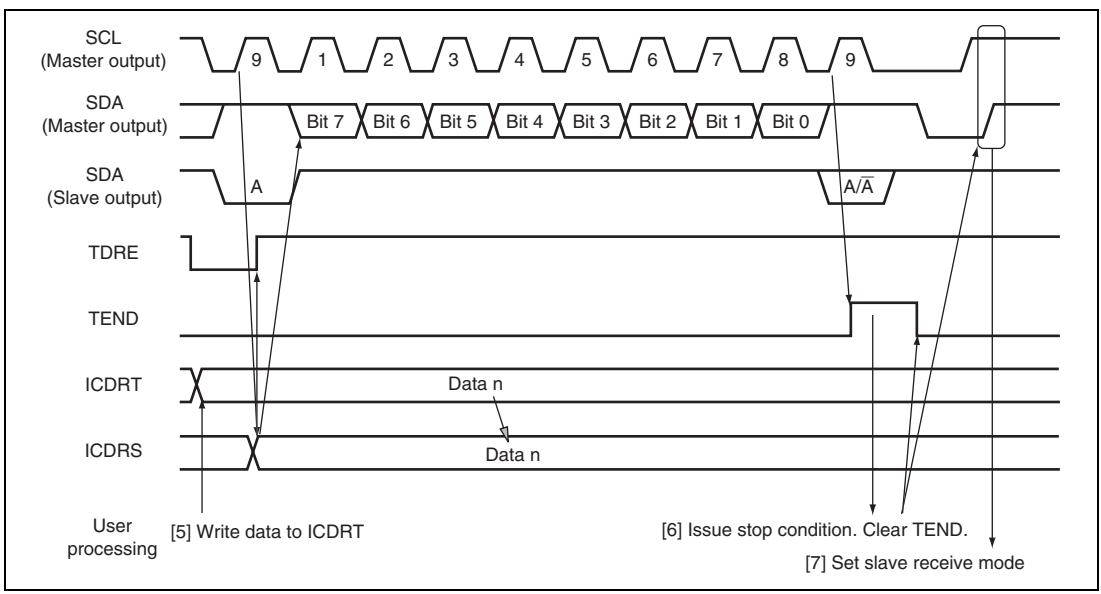



**Figure 18.6 Master Transmit Mode Operation Timing (2)**

### 18.4.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. For master receive mode operation timing, see figures 18.7 and 18.8. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCR1 to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0.
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICST is set to 1 at the rise of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, and RDRF is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. If 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stage condition.
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to the slave receive mode.

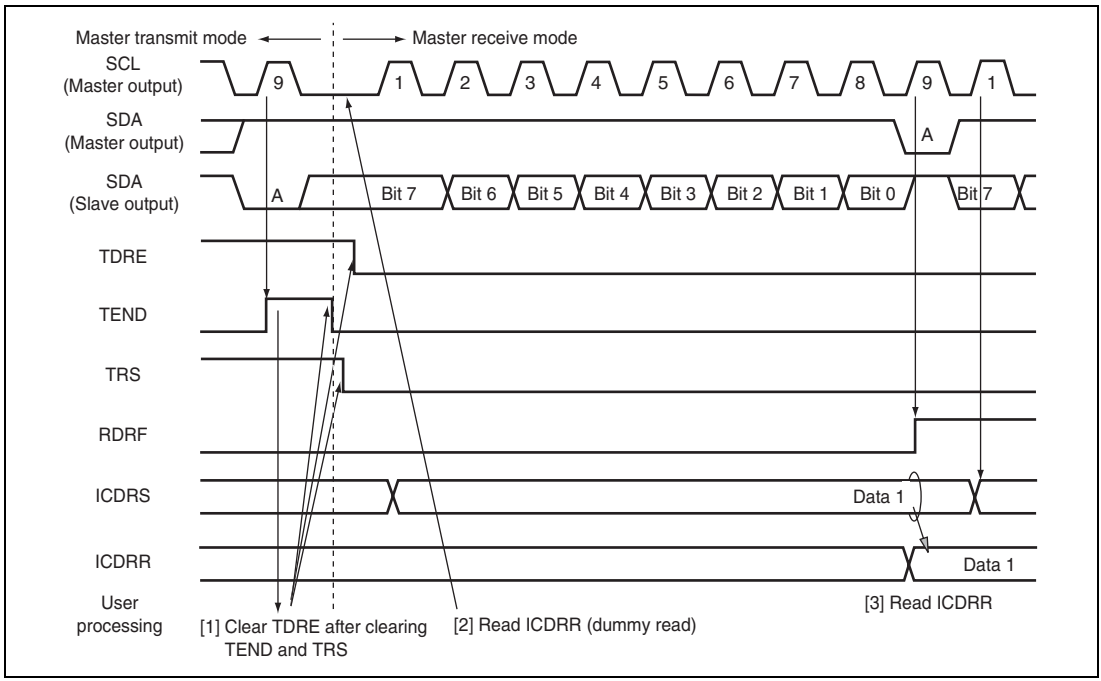



**Figure 18.7 Master Receive Mode Operation Timing (1)**

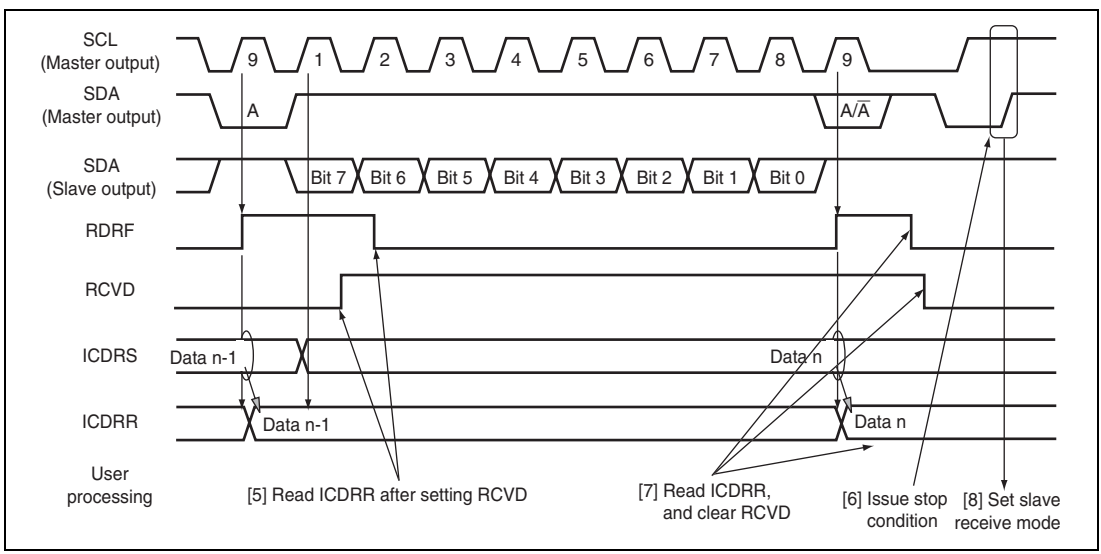



**Figure 18.8 Master Receive Mode Operation Timing (2)**

### 18.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. For slave transmit mode operation timing, see figures 18.9 and 18.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS and ICSR bits in ICCR1 are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by writing transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is set to 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). SCL is free.
5. Clear TDRE.

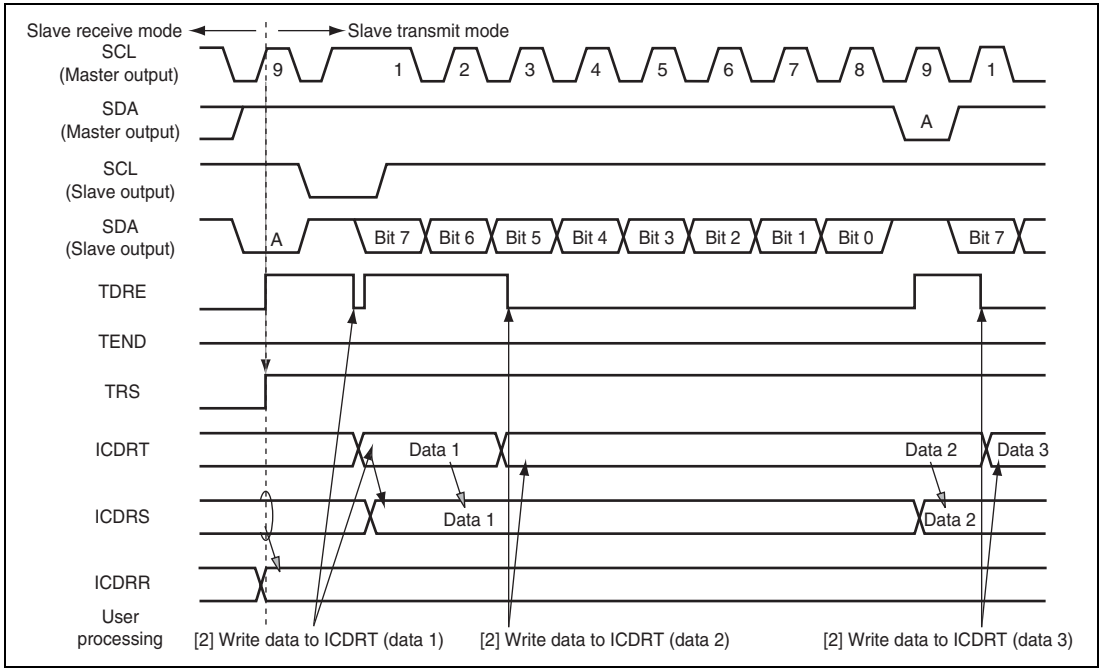



**Figure 18.9 Slave Transmit Mode Operation Timing (1)**

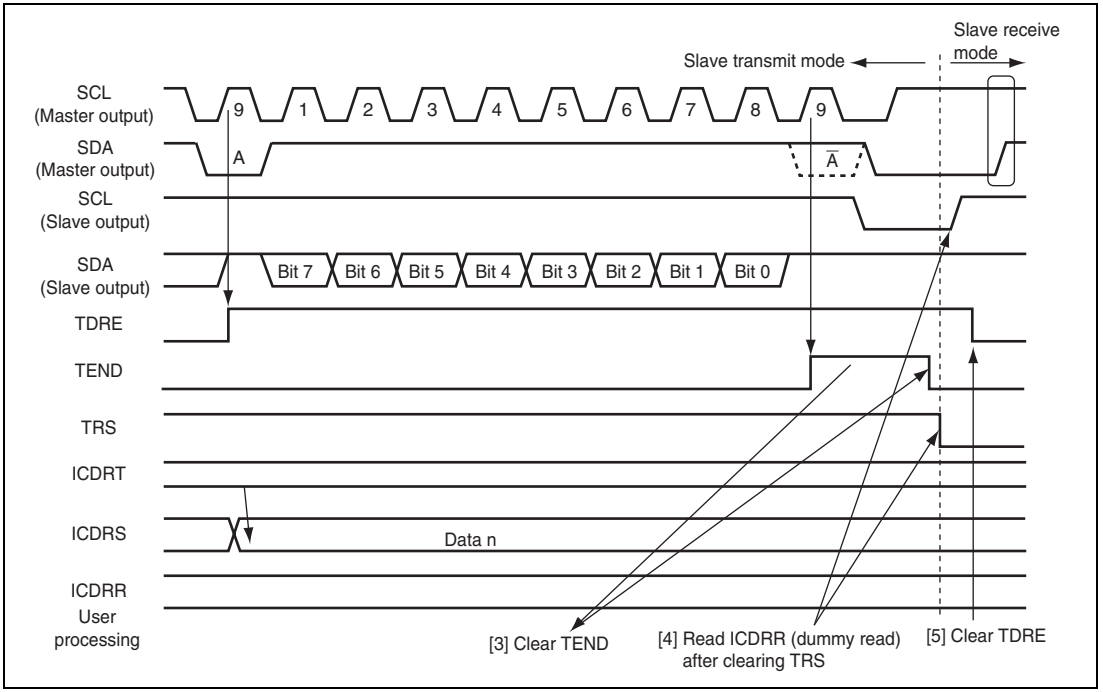



**Figure 18.10 Slave Transmit Mode Operation Timing (2)**

### 18.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For slave receive mode operation timing, see figures 18.11 and 18.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (Since the read data show the slave address and R/$\overline{W}$, it is not used.)
3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is 1, SCL is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR, to be returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

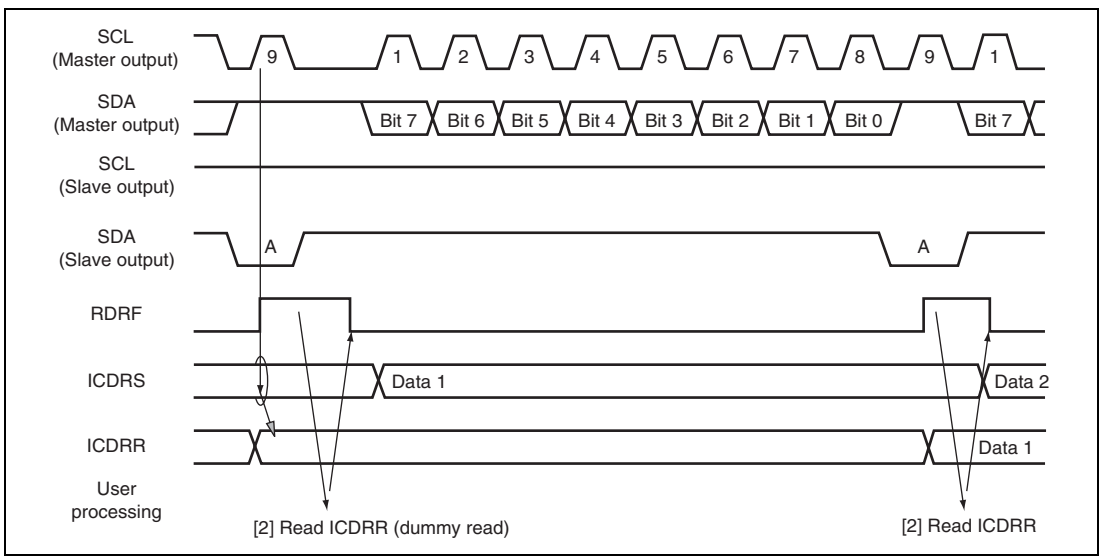



**Figure 18.11 Slave Receive Mode Operation Timing (1)**

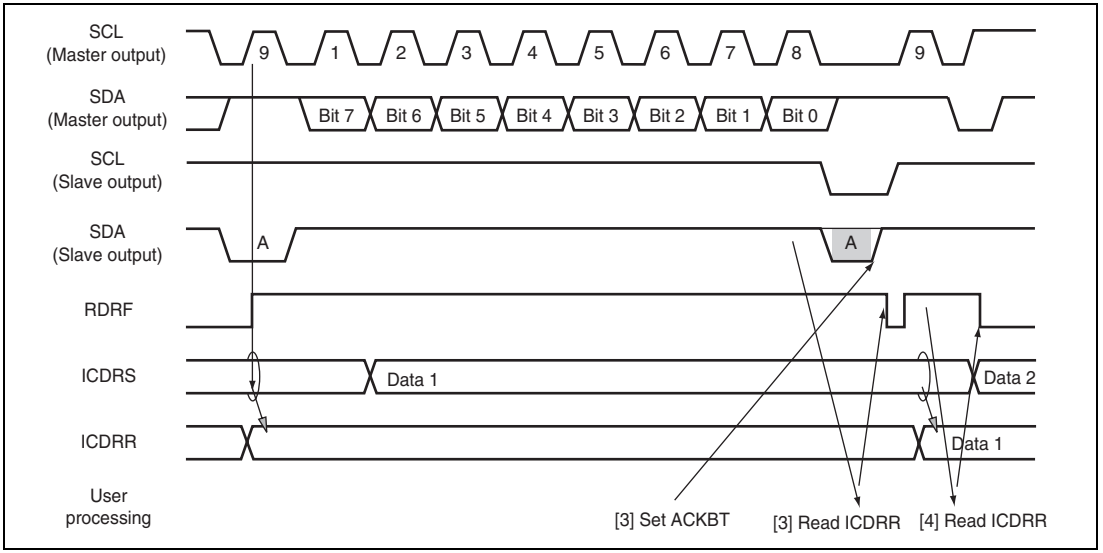



**Figure 18.12 Slave Receive Mode Operation Timing (2)**

### 18.4.6 Clocked Synchronous Serial Format

This module can be operated with the clocked synchronous serial format, by setting the FS bit in SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selected. When MST is 0, the external clock input is selected.

#### (1) Data Transfer Format

Figure 18.13 shows the clocked synchronous serial transfer format.

The transfer data is output from the rise to the fall of the SCL clock, and the data at the rising edge of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in either the MSB first or LSB first. The output level of SDA can be changed during the transfer wait, by the SDAO bit in ICCR2.

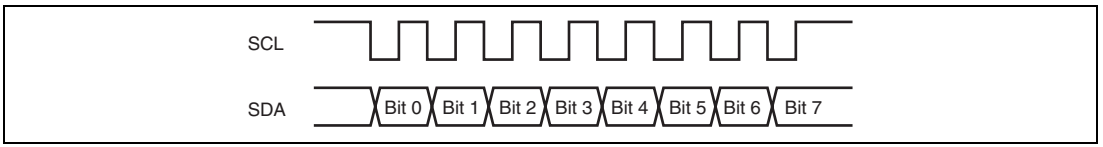



**Figure 18.13 Clocked Synchronous Serial Transfer Format**

### (2) Transmit Operation

In transmit mode, transmit data is output from SDA, in synchronization with the fall of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For transmit mode operation timing, see figure 18.14. The transmission procedure and operations in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. Set the TRS bit in ICCR1 to select the transmit mode. Then, TDRE in ICSR is set.
3. Confirm that TDRE has been set. Then, write the transmit data to ICDRT. The data is transferred from ICDRT to ICDRS, and TDRE is set automatically. The continuous transmission is performed by writing data to ICDRT every time TDRE is set. When changing from transmit mode to receive mode, clear TRS while TDRE is 1.

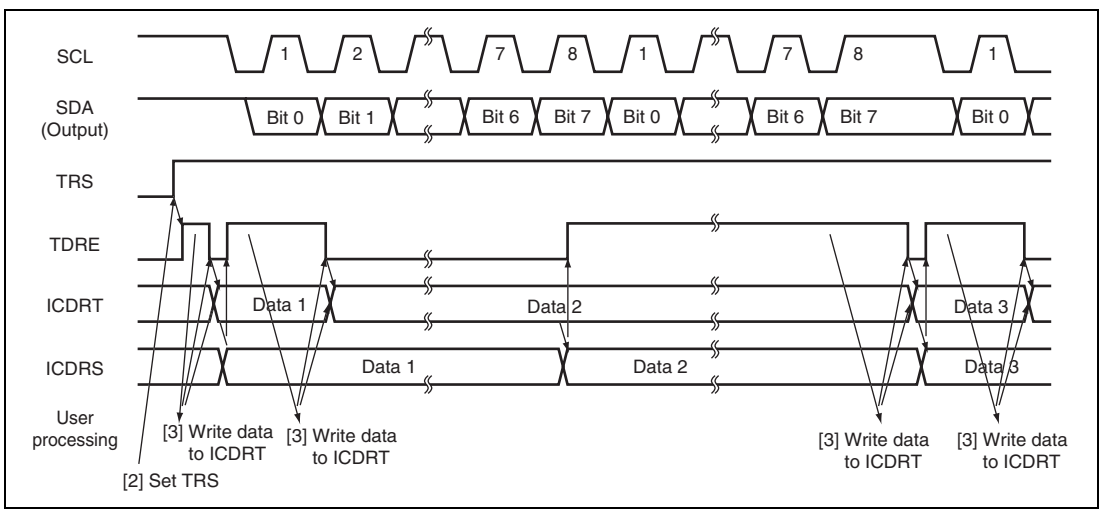



**Figure 18.14 Transmit Mode Operation Timing**

### (3) Receive Operation

In receive mode, data is latched at the rise of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For receive mode operation timing, see figure 18.15. The reception procedure and operations in receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR and RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every time RDRF is set. When the 8th clock is risen while RDRF is 1, the overrun is detected and AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDRR.
4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then, SCL is fixed high after receiving the next byte data.

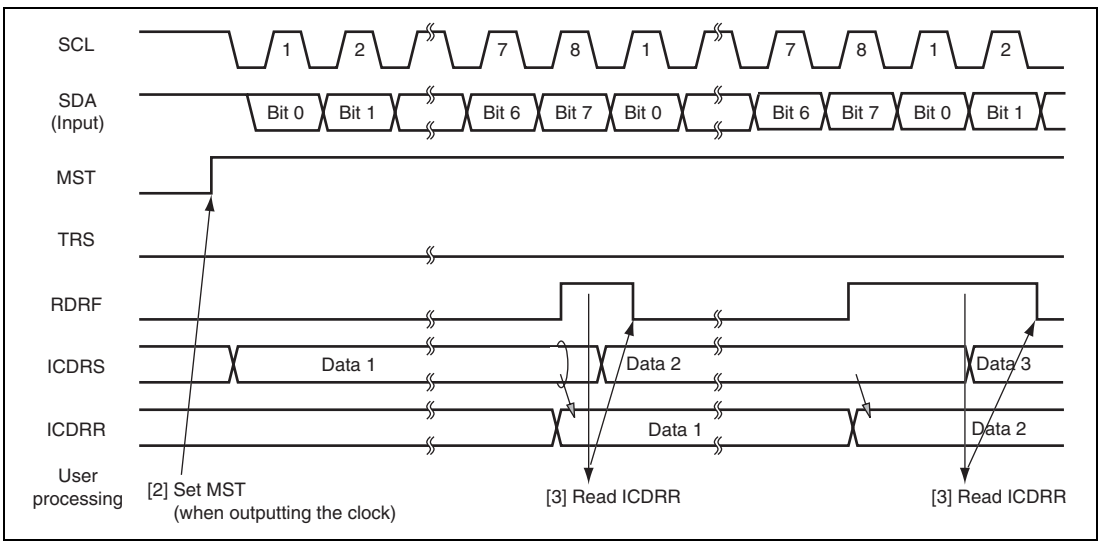



**Figure 18.15 Receive Mode Operation Timing**

### 18.4.7 Noise Canceller

The logic levels at the SCL and SDA pins are routed through noise cancellers before being latched internally. Figure 18.16 shows a block diagram of the noise canceller circuit.

The noise canceller consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.

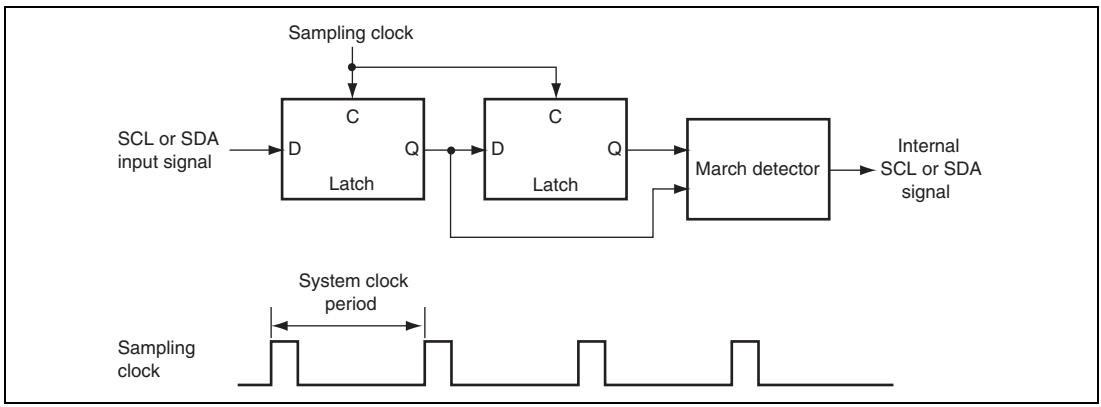



**Figure 18.16 Block Diagram of Noise Canceller**

### 18.4.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface are shown in figures 18.17 to 18.20.

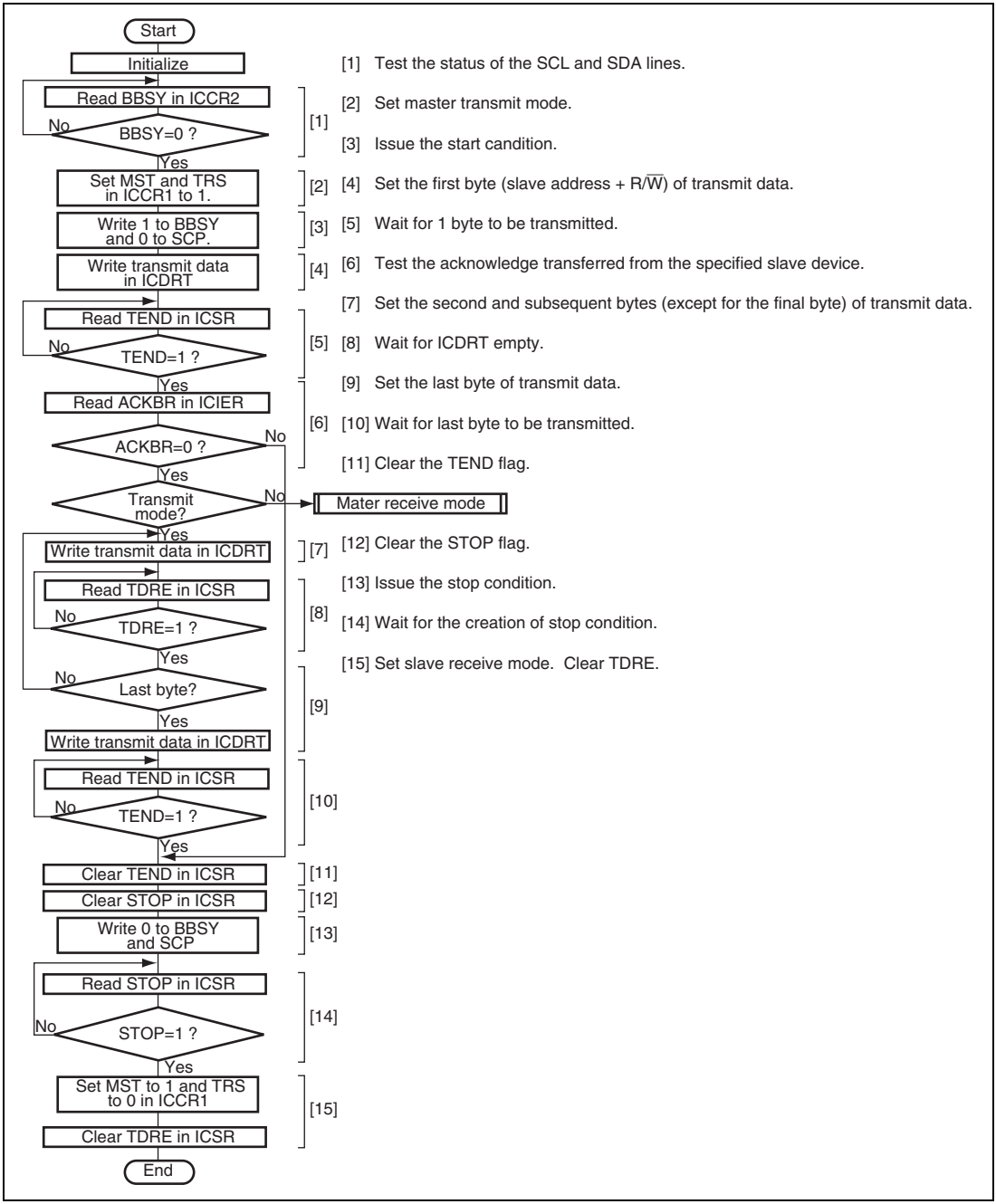
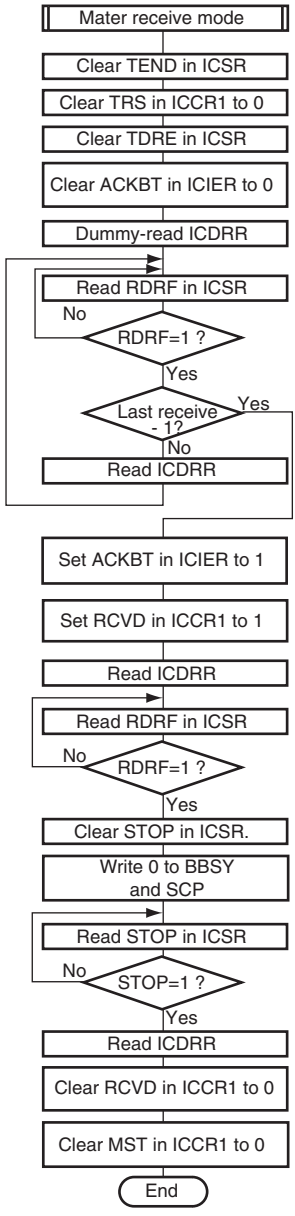



**Figure 18.17 Sample Flowchart for Master Transmit Mode**




[1] [1] Clear TEND, select master receive mode, and then clear TDRE.*
[2] Set acknowledge to the transmit device.*
[3] Dummy-read ICDDR.*

[2] [4] Wait for 1 byte to be received

[3] [5] Check whether it is the (last receive - 1).

[4] [6] Read the receive data last.
[7] Set acknowledge of the final byte. Disable continuous reception (RCVD = 1).
[8] Read the (final byte - 1) of receive data.

[5] [9] Wait for the last byte to be receive.

[6] [10] Clear the STOP flag.
[11] Issue the stop condition.

[7] [12] Wait for the creation of stop condition.
[13] Read the last byte of receive data.
[14] Clear RCVD.

[8] [15] Set slave receive mode.

[9]

[10]

[11]

[12]

[13]

[14]

[15]

Note: Do not activate an interrupt during the execution of steps [1] to [3].

Supplementary explanation: When one byte is received, steps [2] to [6] are skipped after step [1], before jumping to step [7]. The step [8] is dummy-read in ICDRR.

**Figure 18.18 Sample Flowchart for Master Receive Mode**

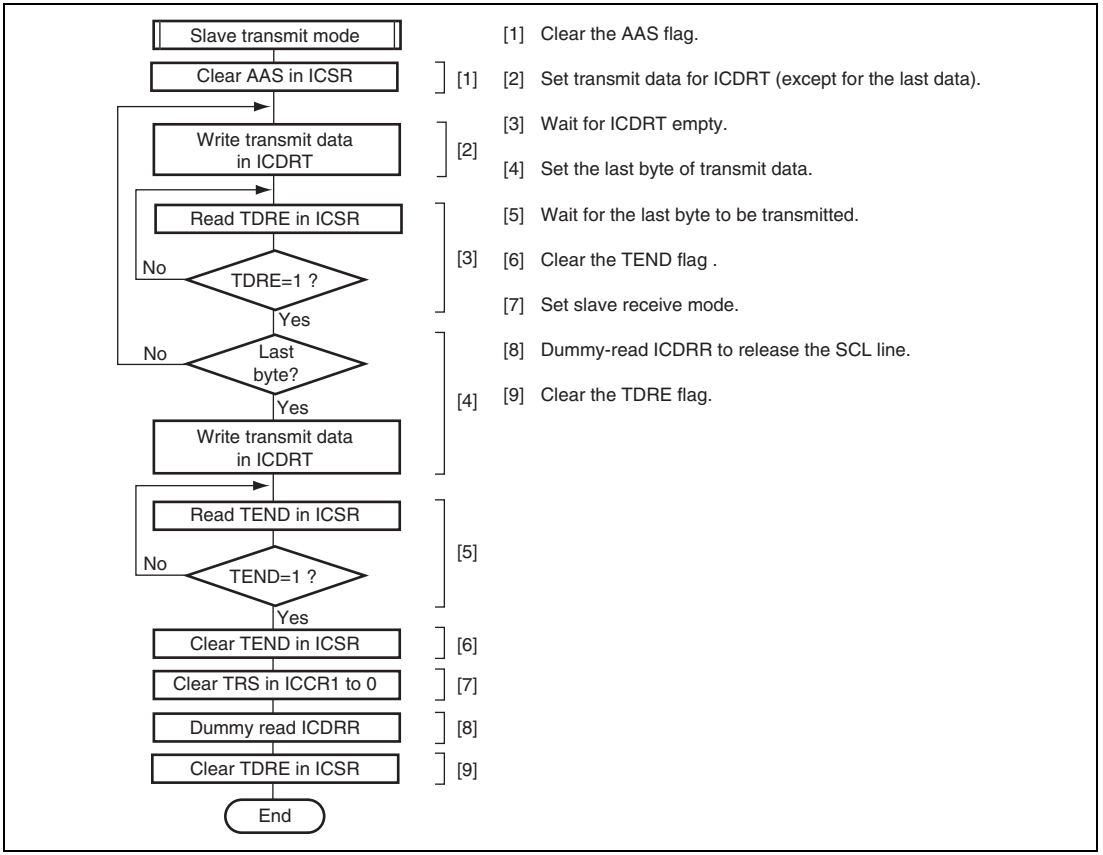
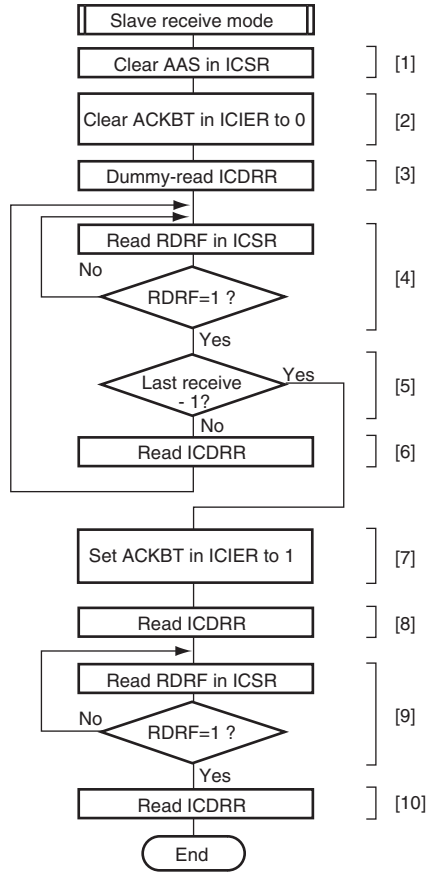



**Figure 18.19 Sample Flowchart for Slave Transmit Mode**




[1] Clear the AAS flag.
[2] Set acknowledge to the transmit device.
[3] Dummy-read ICDRR.
[4] Wait for 1 byte to be received.
[5] Check whether it is the (last receive - 1).
[6] Read the receive data.
[7] Set acknowledge of the last byte.
[8] Read the (last byte - 1) of receive data.
[9] Wait the last byte to be received.
[10] Read for the last byte of receive data.

Supplementary explanation: When one byte is received, steps [2] to [6] are skipped after step [1], before jumping to step [7]. The step [8] is dummy-read in ICDRR.

**Figure 18.20 Sample Flowchart for Slave Receive Mode**

## 18.5 Interrupts

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK receive, STOP recognition, and arbitration lost/overrun error. Table 18.3 shows the contents of each interrupt request.

**Table 18.3 Interrupt Requests**

| Interrupt Request | Abbreviation | Interrupt Condition | I²C Mode | Clocked Synchronous Mode |
|---|---|---|---|---|
| Transmit Data Empty | TXI | (TDRE=1) • (TIE=1) | ○ | ○ |
| Transmit End | TEI | (TEND=1) • (TEIE=1) | ○ | ○ |
| Receive Data Full | RXI | (RDRF=1) • (RIE=1) | ○ | ○ |
| STOP Recognition | STPI | (STOP=1) • (STIE=1) | ○ | × |
| NACK Receive | NAKI | {(NACKF=1)+(AL=1)} • (NAKIE=1) | ○ | × |
| Arbitration Lost/Overrun Error | | | ○ | ○ |

When interrupt conditions described in table 18.3 are 1 and the I bit in CCR is 0, the CPU executes an interrupt exception processing. Interrupt sources should be cleared in the exception processing. TDRE and TEND are automatically cleared to 0 by writing the transmit data to ICDRT. RDRF are automatically cleared to 0 by reading ICDRR. TDRE is set to 1 again at the same time when transmit data is written to ICDRT. When TDRE is cleared to 0, then an excessive data of one byte may be transmitted.

## 18.6 Bit Synchronous Circuit

In master mode, this module has a possibility that high level period may be short in the two states described below.

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lowered by the load of the SCL line (load capacitance or pull-up resistance)

Therefore, it monitors SCL and communicates by bit with synchronization.

Figure 18.21 shows the timing of the bit synchronous circuit and table 18.4 shows the time when SCL output changes from low to Hi-Z then SCL is monitored.

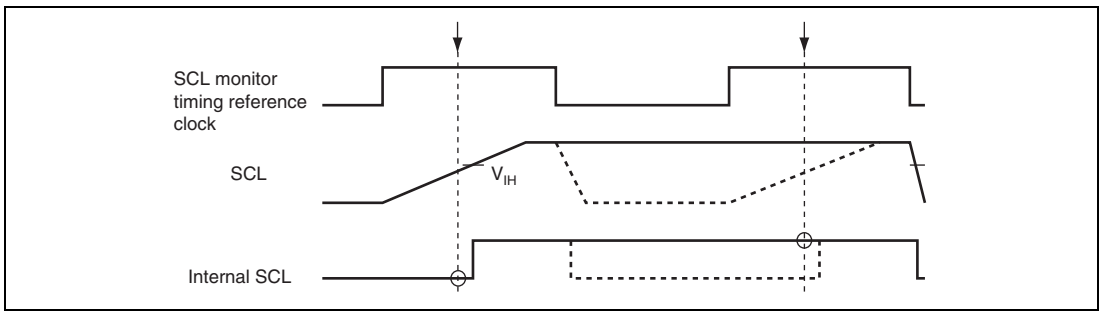



**Figure 18.21 Timing of Bit Synchronous Circuit**

**Table 18.4 Time for Monitoring SCL**

| CKS3 | CKS2 | Time for Monitoring SCL |
|---|---|---|
| 0 | 0 | 7.5 tcyc |
| | 1 | 19.5 tcyc |
| 1 | 0 | 17.5 tcyc |
| | 1 | 41.5 tcyc |

## 18.7 Usage Notes

### 18.7.1 Restriction on Setting of Transfer Rate in Use of Multi-Master

In multi-master usage when the IIC transfer rate setting of this LSI is lower than those of the other masters, unexpected length of SCL may occasionally be output. To avoid this, the specified value must be greater than or equal to the value produced by multiplying the fastest transfer rate among the other masters by 1/1.8. For example, when the transfer rate of the fastest bus master among the other bus masters is 400 kbps, the transfer rate of the IIC of this LSI must be set to at least 223 kbps (= 400/1.8).

### 18.7.2 Restriction on Use of Bit Manipulation Instructions to Set MST and TRS when Multi-Master is Used

When master transmission is selected by consecutively manipulating the MST and TRS bits in multi-master usage, an arbitration loss during execution of the bit-manipulation instruction for TRS leads to the contradictory situation where AL in ICSR is 1 in master transmit mode (MST = 1, TRS = 1).

Ways to avoid this effect are listed below.

- Use the MOV instruction to set MST and TRS in multi-master usage.
- When arbitration is lost, confirm that MST = 0 and TRS = 0. If the setting of MST = 0 and TRS = 0 is not confirmed, set MST = 0 and TRS = 0 again.

### 18.7.3 Usage Note on Master Receive Mode

In master receive mode, when SCL is fixed low on the falling edge of the 8th clock while the RDRF bit is set to 1 and ICDRR is read around the falling edge of the 8th clock, the clock is only fixed low in the 8th clock of the next round of data reception.

The SCL is then released from its fixed state without reading ICDRR and the 9th clock is output. As a result, some receive data is lost.

Ways to avoid this phenomenon are listed below.

- Read ICDRR in master receive mode before the rising edge of the 8th clock.
- Set RCVD to 1 in master receive mode and perform communication in units of one byte.

### 18.7.4 Note Regarding Master Receive Mode of I²C-Bus Interface Mode

When stop condition issuance or start condition reissuance overlaps with the falling edge of the 9th clock cycle of the SCL signal, an additional cycle is output after the 9th clock cycle.

As a workaround, After a master receive operation is completed, confirm the falling edge of the 9th clock cycle of the SCL signal and issue a stop condition or reissue a start condition.

Confirm the falling edge of the 9th clock cycle of the SCL signal as follows: Confirm the SCLO bit in the ICCR2 register (SCL monitor flag) becomes 0 (SCL pin is low) after confirming the RDRF bit in the ICSR register (receive data register full flag) becomes 1.

### 18.7.5 Accessing ICE and IICRST During I²C Bus Operation

< Problem >

When 0 is written to ICE in ICCR1 or 1 is written to IICRST in ICCR2 during I²C bus operation, the values of BBSY in ICCR2 and STOP in ICSR become undefined.

< Conditions >

The above problem occurs when 0 is written to ICE in ICCR1 or 1 is written to IICRST in ICCR2 when any of conditions (1) to (4) below is satisfied.

(1) In master transmit mode (MST = 1 and TRS = 1 in ICCR1), when the IIC2 module holds access rights to the I²C bus.

(2) In master receive mode (MST = 1 and TRS = 0 in ICCR1), when the IIC2 module holds access rights to the I²C bus.

(3) In slave transmit mode (MST = 0 and TRS = 1 in ICCR1), when the IIC2 module is performing a data transmit operation.

(4) In slave receive mode (MST = 0 and TRS = 0 in ICCR1), when the IIC2 module is transmitting an acknowledge signal.

< Workarounds >

The undefined state of BBSY in ICCR2 can be canceled by using one of the following workarounds.

- BBSY is set to 1 when a start condition (SCL is high-level and SDA falling edge) is input.

- BBSY is cleared to 0 when an stop condition (SCL is high-level and SDA rising edge) is input.
- In master transmit mode, write 1 to BBSY and 0 to SCP in ICCR2 while SCL and SDA are both high-level to issue a start condition. BBSY is set to 1 when a start condition (SCL is high-level and SDA falling edge) is output.
- In master transmit mode or master receive mode, when SDA is low-level and there is no device other than the IIC2 or IIC3 module to drive SCL low, write 0 to BBSY and SCP in ICCR2 to issue an stop condition. BBSY is cleared to 0 when an stop condition (SCL is high-level and SDA rising edge) is output.
- Writing 1 to FS in SAR causes BBSY to be cleared to 0.

Supplemental Description of Register Functions

Using IICRST for register initialization

- Writing 1 to IICRST in ICCR2 causes SDAO and SCLO in ICCR2 to be set to 1.
- In master transmit mode or slave transmit mode, writing 1 to IICRST causes TDRE in ICSR to be set to 1.
- During a reset interval triggered by setting IICRST to 1, write access to BBSY, SCP, and SDAO in ICCR2 is disabled.
- Writing 1 to IICRST does not clear the BBSY bit in ICCR2 to 0. However, depending on the states of pins SCL and SDA, an stop condition (SCL is high-level and SDA rising edge) may occur, resulting in BBSY being cleared to 0. In addition, there may be similar effects on other bits as well.
- During a reset interval triggered by setting IICRST to 1, data reception and transmission halts. However, the functions that detect start conditions, stop conditions, and bus arbitration lost conditions are active. This means that the states of ICCR1, ICCR2, and ICSR may be updated, depending on the signals input to SCL and SDA.

### 18.7.6 Note on Issuance of Stop Condition in Master Transmit Mode

When a stop condition is issued in master transmit mode while the ACKE bit in the I2C bus interrupt enable register (ICIER) is 1, the stop condition may not be normally output depending on the issued timing.

In master transmit mode while the ACKE bit in the I2C bus interrupt enable register (ICIER) is 1, recognize the falling edge of the 9th clock before issuance of the stop condition. The falling edge of the 9th clock can be recognized by checking the SCLO bit in the I2C control register 2 (ICCR2).

# Section 19 A/D Converter

This LSI includes a 10-bit successive approximation A/D converter that allows up to 16 analog input channels to be selected. The block diagram of the A/D converter is shown in figure 19.1.

## 19.1 Features

- 10-bit resolution
- 16 input channels
- Conversion time: 3.5 μs per channel at 20-MHz operation (minimum)
- Two operating modes
  Single mode: Single-channel A/D conversion
  Scan mode: Continuous A/D conversion on one to four channels
- Four data registers
  Conversion results are held in a data register for each channel
- Sample-and-hold function
- Two conversion start methods
  Software
  External trigger signal
- Interrupt source
  An A/D conversion end interrupt (ADI) request can be generated

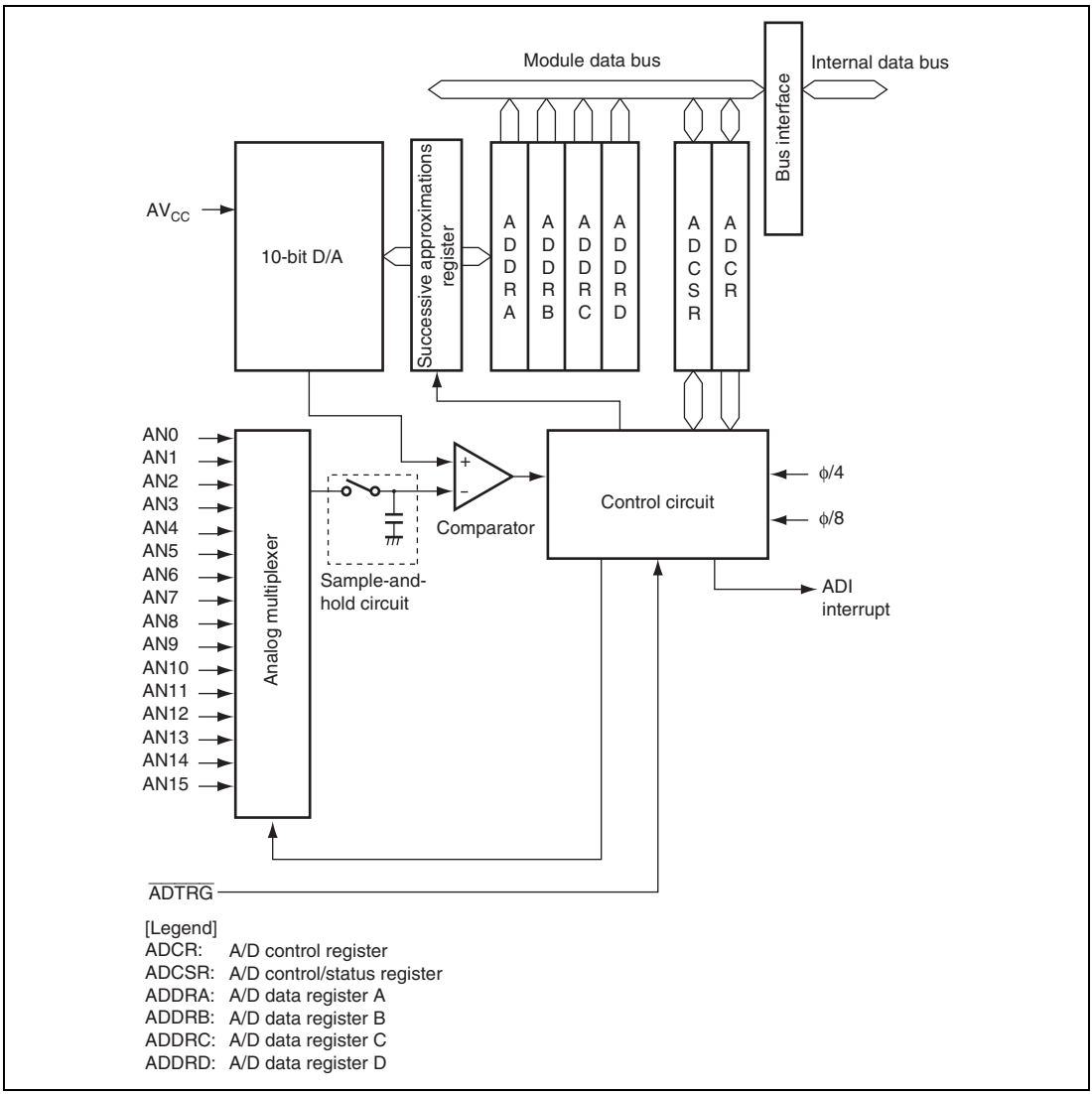



**Figure 19.1 Block Diagram of A/D Converter**

## 19.2 Input/Output Pins

Table 19.1 summarizes the input pins used by the A/D converter. The 16 analog input pins are divided into four groups, each of which has four channels. Group 0 comprises analog input pins 0 to 3 (AN0 to AN3), group 1 comprises analog input pins 4 to 7 (AN4 to AN7), group 2 comprises analog input pins 8 to 11 (AN8 to AN11), and group 3 comprises analog input pins 12 to 15 (AN12 to AN15). The AVcc pin is the power supply pin for the analog block in the A/D converter.

**Table 19.1 Pin Configuration**

| Pin Name | Abbreviation | I/O | Function |
|---|---|---|---|
| Analog power supply pin | $AV_{CC}$ | Input | Analog block power supply |
| Analog input pin 0 | AN0 | Input | Group 0 analog input |
| Analog input pin 1 | AN1 | Input | |
| Analog input pin 2 | AN2 | Input | |
| Analog input pin 3 | AN3 | Input | |
| Analog input pin 4 | AN4 | Input | Group 1 analog input |
| Analog input pin 5 | AN5 | Input | |
| Analog input pin 6 | AN6 | Input | |
| Analog input pin 7 | AN7 | Input | |
| Analog input pin 8 | AN8 | Input | Group 2 analog input |
| Analog input pin 9 | AN9 | Input | |
| Analog input pin 10 | AN10 | Input | |
| Analog input pin 11 | AN11 | Input | |
| Analog input pin 12 | AN12 | Input | Group 3 analog input |
| Analog input pin 13 | AN13 | Input | |
| Analog input pin 14 | AN14 | Input | |
| Analog input pin 15 | AN15 | Input | |
| A/D external trigger input pin | $\overline{\text{ADTRG}}$ | Input | External trigger input for starting A/D conversion |

## 19.3 Register Descriptions

The A/D converter has the following registers.

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

### 19.3.1 A/D Data Registers A to D (ADDRA to ADDRD)

There are four 16-bit read-only ADDR registers; ADDRA to ADDRD, used to store the results of A/D conversion. The ADDR registers, which store a conversion result for each analog input channel, are shown in table 19.2.

The converted 10-bit data is stored in bits 15 to 6. The lower 6 bits are always read as 0.

The data bus width between the CPU and the A/D converter is 8 bits. The upper byte can be read directly from the CPU, however the lower byte should be read via a temporary register. The temporary register contents are transferred from the ADDR when the upper byte data is read. Therefore byte access to ADDR should be done by reading the upper byte first then the lower one. Word access is also possible. ADDR is initialized to H'0000.

**Table 19.2 Analog Input Channels and Corresponding ADDR Registers**

| Analog Input Channel | | | | |
|---|---|---|---|---|
| CH3 = 0 | | CH3 = 1 | | |
| Group 0 (CH2 = 0) | Group 1 (CH2 = 1) | Group 2 (CH2 = 0) | Group 3 (CH2 = 1) | A/D Data Register to Store Results of A/D Conversion |
| AN0 | AN4 | AN8 | AN12 | ADDRA |
| AN1 | AN5 | AN9 | AN13 | ADDRB |
| AN2 | AN6 | AN10 | AN14 | ADDRC |
| AN3 | AN7 | AN11 | AN15 | ADDRD |

### 19.3.2 A/D Control/Status Register (ADCSR)

ADCSR consists of the control bits and conversion end status bits of the A/D converter.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | ADF | 0 | R/W | A/D End Flag<br>[Setting conditions]<br>• When A/D conversion ends in single mode<br>• When A/D conversion ends once on all the channels selected in scan mode<br>[Clearing condition]<br>When 0 is written after reading ADF = 1 |
| 6 | ADIE | 0 | R/W | A/D Interrupt Enable<br>A/D conversion end interrupt request (ADI) is enabled by ADF when this bit is set to 1 |
| 5 | ADST | 0 | R/W | A/D Start<br>Setting this bit to 1 starts A/D conversion. In single mode, this bit is cleared to 0 automatically when conversion on the specified channel is complete. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by software, a reset, or a transition to standby mode. |
| 4 | SCAN | 0 | R/W | Scan Mode<br>Selects single mode or scan mode as the A/D conversion operating mode.<br>0: Single mode<br>1: Scan mode |
| 3 | CKS | 0 | R/W | Clock Select<br>Selects the A/D conversions time.<br>0: Conversion time = 134 states (max.)<br>1: Conversion time = 70 states (max.)<br>Clear the ADST bit to 0 before switching the conversion time. |

| Bit | Bit Name | Initial Value | R/W | Description | |
|---|---|---|---|---|---|
| 2 | CH2 | 0 | R/W | Channel Select 2 to 0 | |
| 1 | CH1 | 0 | R/W | Select analog input channels according to a combination of the CH3 bit in ADCR. | |
| 0 | CH0 | 0 | R/W | When SCAN = 0 | When SCAN = 1 |
| | | | | 0000: AN0 | 0000: AN0 |
| | | | | 0001: AN1 | 0001: AN0 and AN1 |
| | | | | 0010: AN2 | 0010: AN0 to AN2 |
| | | | | 0011: AN3 | 0011: AN0 to AN3 |
| | | | | 0100: AN4 | 0100: AN4 |
| | | | | 0101: AN5 | 0101: AN4 and AN5 |
| | | | | 0110: AN6 | 0110: AN4 to AN6 |
| | | | | 0111: AN7 | 0111: AN4 to AN7 |
| | | | | 1000: AN8 | 1000: AN8 |
| | | | | 1001: AN9 | 1001: AN8 and AN9 |
| | | | | 1010: AN10 | 1010: AN8 to AN10 |
| | | | | 1011: AN11 | 1011: AN8 to AN11 |
| | | | | 1100: AN12 | 1100: AN12 |
| | | | | 1101: AN13 | 1101: AN12 and AN13 |
| | | | | 1110: AN14 | 1110: AN12 to AN14 |
| | | | | 1111: AN15 | 1111: AN12 to AN15 |

### 19.3.3 A/D Control Register (ADCR)

ADCR enables A/D conversion started by an external trigger signal.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TRGE | 0 | R/W | Trigger Enable<br>A/D conversion is started by an assertion of the external trigger signal from timer RD or the falling or rising edge of the external $\overline{\text{ADTRG}}$ signal when this bit is set to 1. The trigger source is selected by bits PMRG3 and PMRG2 in port mode register G (PMRG).<br>The falling or rising edge of the external $\overline{\text{ADTRG}}$ signal is selected by bits PMRG2 and PMRG1. |
| 6 to 4 | — | All 1 | — | Reserved<br>These bits are always read as 1. |
| 3, 2 | — | All 0 | R/W | Reserved<br>The write value should always be 0. |
| 1 | — | 1 | — | Reserved<br>This bit is always read as 1. |
| 0 | CH3 | 0 | R/W | Reserved<br>Selects the analog input channel according to bits CH2 to CH0 in ADCSR. |

## 19.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes; single mode and scan mode. When changing the operating mode or analog input channel, in order to prevent incorrect operation, first clear the bit ADST in ADCSR to 0. The ADST bit can be set at the same time as the operating mode or analog input channel is changed.

### 19.4.1 Single Mode

In single mode, A/D conversion is performed once for the analog input of the specified single channel as follows:

1. A/D conversion is started when the ADST bit in ADCSR is set to 1, according to software or external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/D data register of the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters the wait state.

### 19.4.2 Scan Mode

In scan mode, A/D conversion is performed sequentially for the analog input of the specified channels (four channels maximum) as follows:

1. When the ADST bit in ADCSR is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH3 and CH2 = B'00, AN4 when CH3 and CH2 = B'01, AN8 when CH3 and CH2 = B'10, AN12 when CH3 and CH2 = B'11).
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversion of all the selected channels is completed, the ADF flag in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt requested is generated. A/D conversion starts again on the first channel in the group.
4. The ADST bit is not automatically cleared to 0. Steps [2] and [3] are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops.

### 19.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input when the A/D conversion start delay time ($t_D$) has passed after the ADST bit is set to 1, then starts conversion. Figure 19.2 shows the A/D conversion timing. Table 19.3 shows the A/D conversion time.

As indicated in figure 19.2, the A/D conversion time includes $t_D$ and the input sampling time. The length of $t_D$ varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 19.3.

In scan mode, the values given in table 19.3 apply to the first conversion time. In the second and subsequent conversions, the conversion time is 128 states (fixed) when CKS = 0 and 66 states (fixed) when CKS = 1.

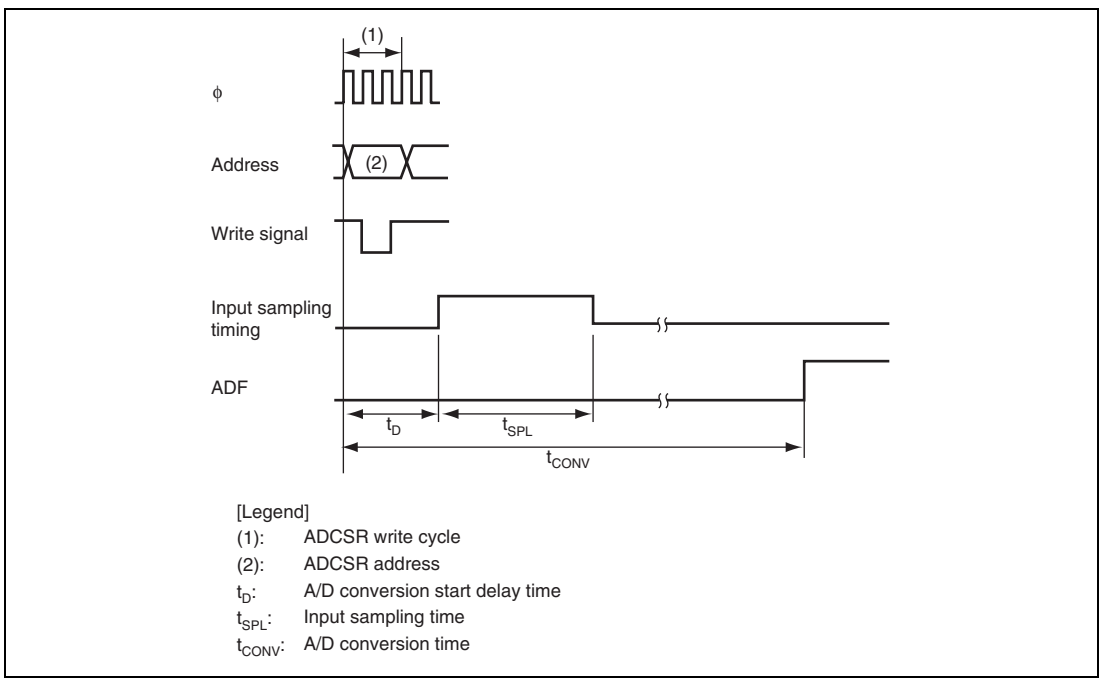



**Figure 19.2 A/D Conversion Timing**

**Table 19.3 A/D Conversion Time (Single Mode)**

| | | CKS = 0 | | | CKS = 1 | | |
|---|---|---|---|---|---|---|---|
| **Item** | **Symbol** | **Min.** | **Typ.** | **Max.** | **Min.** | **Typ.** | **Max.** |
| A/D conversion start delay time | $t_D$ | 6 | — | 9 | 4 | — | 5 |
| Input sampling time | $t_{SPL}$ | — | 31 | — | — | 15 | — |
| A/D conversion time | $t_{CONV}$ | 131 | — | 134 | 69 | — | 70 |

Note: All values represent the number of states.

### 19.4.4 External Trigger Input Timing

A/D conversion can also be started by an external trigger input. When the TRGE bit in ADCR is set to 1, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTRG}}$ input pin sets the ADST bit in ADCSR to 1, starting A/D conversion. Other operations, in both single and scan modes, are the same as when the bit ADST has been set to 1 by software. Figure 19.3 shows the timing.

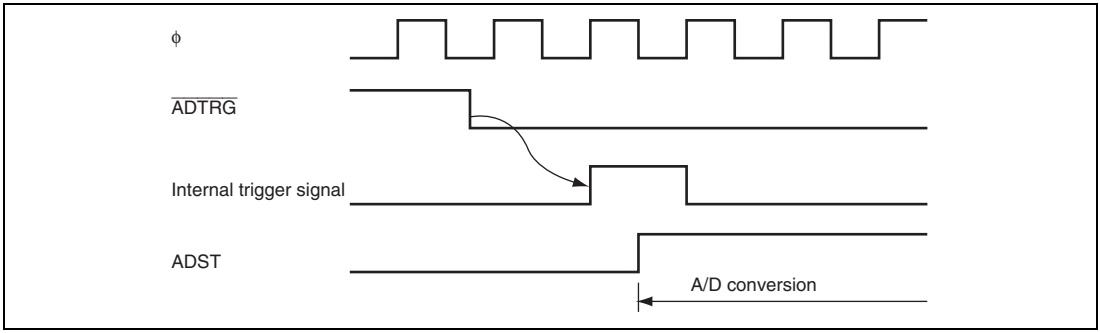



**Figure 19.3 External Trigger Input Timing**

## 19.5 A/D Conversion Accuracy Definitions

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital output codes
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 19.4).
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value 0000000000 to 0000000001 (see figure 19.5).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 1111111110 to 1111111111 (see figure 19.5).
- Nonlinearity error

  The deviation from the ideal A/D conversion characteristic as the voltage changes from zero to full scale. This does not include the offset error, full-scale error, or quantization error.
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes offset error, full-scale error, quantization error, and nonlinearity error.

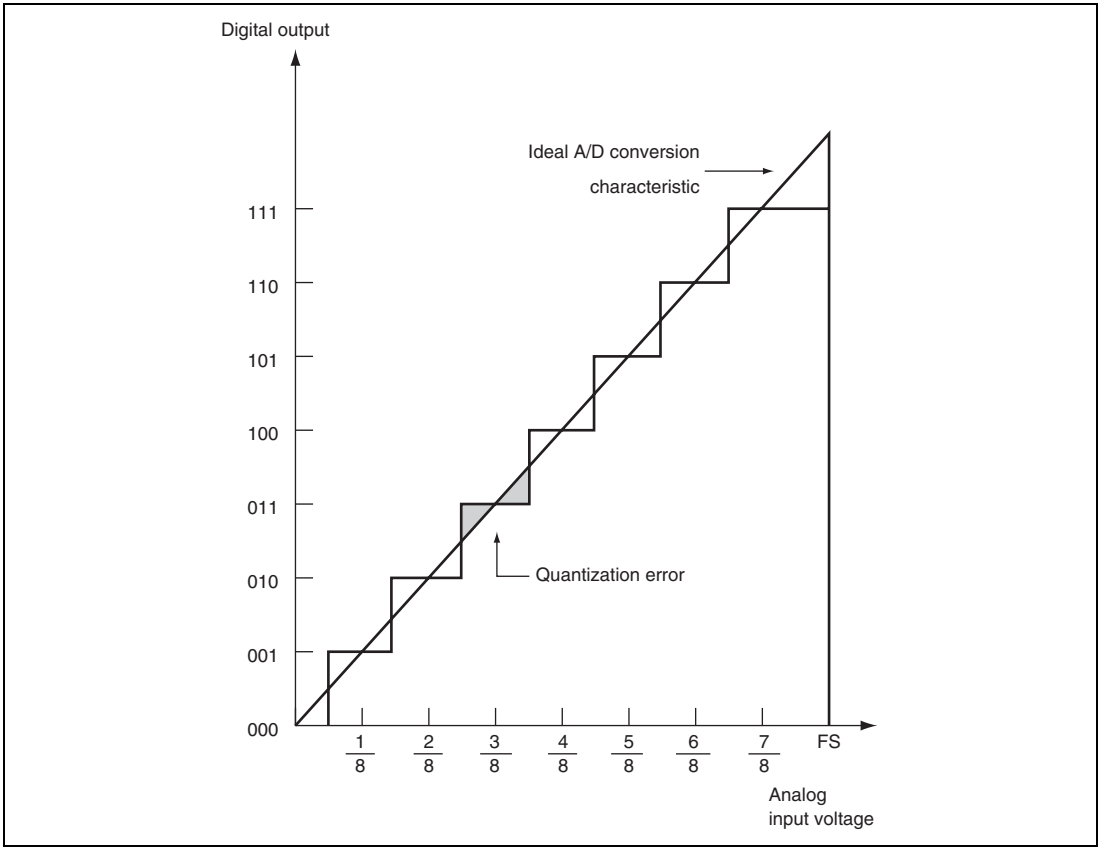



**Figure 19.4 A/D Conversion Accuracy Definitions (1)**

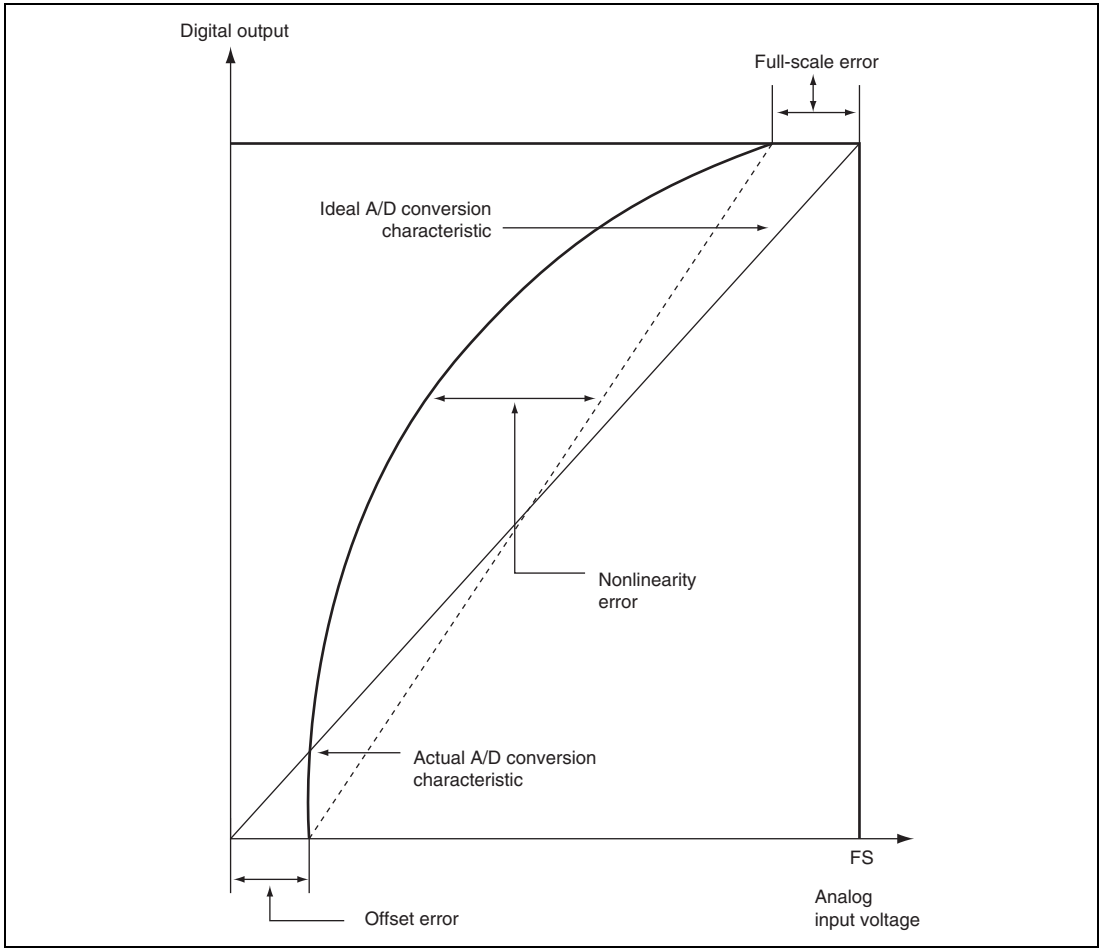



**Figure 19.4 A/D Conversion Accuracy Definitions (2)**

## 19.6 Usage Notes

### 19.6.1 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion accuracy is guaranteed for an input signal for which the signal source impedance is 5 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion accuracy. However, for A/D conversion in single mode with a large capacitance provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/μs or greater) (see figure 19.5). When converting a high-speed analog signal or converting in scan mode, a low-impedance buffer should be inserted.

### 19.6.2 Influences on Absolute Accuracy

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute accuracy. Be sure to make the connection to an electrically stable GND.

Care is also required to ensure that filter circuits do not interfere with digital signals or act as antennas on the board.

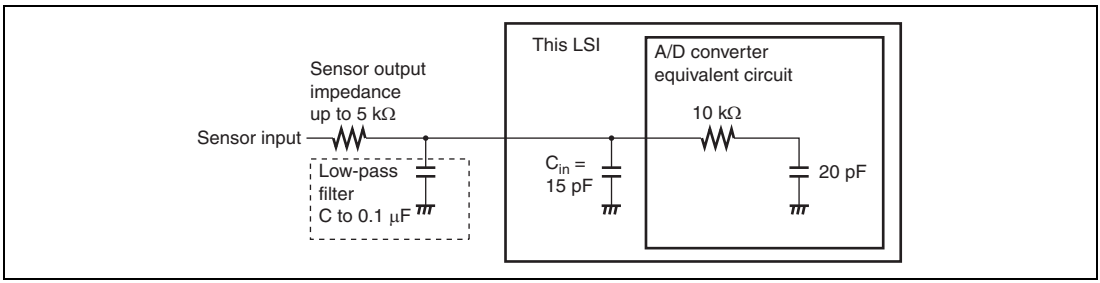



**Figure 19.5 Analog Input Circuit Example**

### 19.6.3 Notes on Analog Pins

The AN8 to AN15 pins also function as port G pins. Therefore, switching input/output of port G or changing the output value during A/D conversion may affect the conversion accuracy. Evaluate the accuracy of A/D conversion sufficiently, when port G is used as a general I/O port.

# Section 20 Band-Gap Regulator, Power-On Reset (Optional), and Low-Voltage Detection Circuits (Optional)

This LSI includes a band-gap regulator (BGR), and can include a power-on reset circuit and low-voltage detection circuit as optional circuits.

The BGR supplies a reference voltage to the on-chip oscillator and low-voltage detection circuit. Figure 20.1 is a block diagram showing the position of the BGR.

The low-voltage detection circuit consists of two circuits: LVDI (interrupt by low voltage detect) and LVDR (reset by low voltage detect) circuits.

This circuit is used to prevent abnormal operation (runaway execution) from occurring due to the power supply voltage fall and to recreate the state before the power supply voltage fall when the power supply voltage rises again.

Even if the power supply voltage falls, the unstable state when the power supply voltage falls below the guaranteed operating voltage can be removed by entering standby mode when exceeding the guaranteed operating voltage and during normal operation. Thus, system stability can be improved. If the power supply voltage falls more, the reset state is automatically entered. If the power supply voltage rises again, the reset state is held for a specified period, then active mode is automatically entered.

Figure 20.2 is a block diagram of the power-on reset circuit and the low-voltage detection circuit.

## 20.1 Features

- BGR circuit

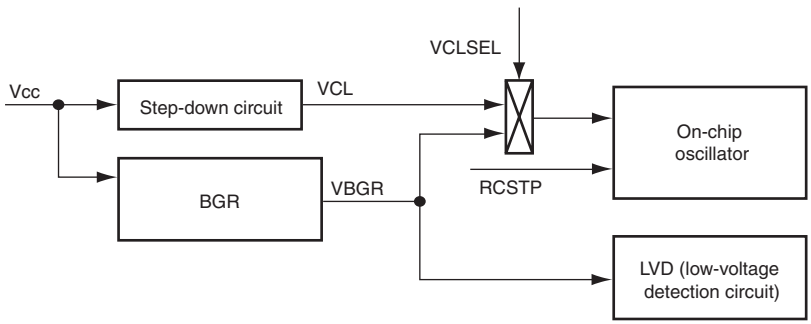
  Supplies stable reference voltage covering the entire operating voltage range and the operating temperature range.
- Power-on reset circuit

  Uses an external capacitor to generate an internal reset signal when power is first supplied.
- Low-voltage detection circuit

  LVDR: Monitors the power-supply voltage, and generates an internal reset signal when the voltage falls below a specified value.

  LVDI: Monitors the power-supply voltage, and generates an interrupt when the voltage falls below or rises above respective specified values.

  Two pairs of detection levels for reset generation voltage are available: when only the LVDR circuit is used, or when the LVDI and LVDR circuits are both used.




[Legend]
Vcc: Power supply
VCL: Internal power supply generated from Vcc by the step-down circuit
VBGR: Reference voltage from BGR
VCLSEL: Select signal for the source of the on-chip oscillator power supply
RCSTP: On-chip oscillator stop signal

**Figure 20.1 Block Diagram around BGR**

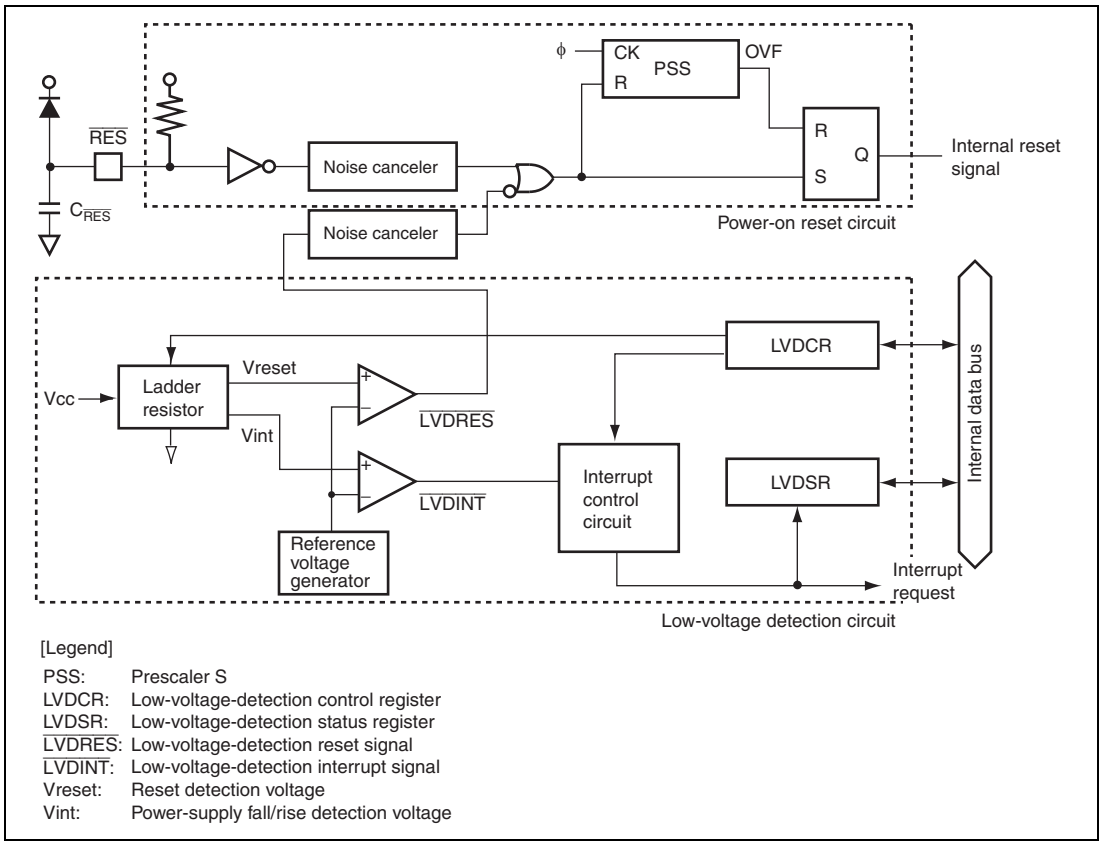



**Figure 20.2 Block Diagram of Power-On Reset Circuit and Low-Voltage Detection Circuit**

## 20.2 Register Descriptions

The low-voltage detection circuit has the following registers.

- Low-voltage-detection control register (LVDCR)
- Low-voltage-detection status register (LVDSR)

### 20.2.1 Low-Voltage-Detection Control Register (LVDCR)

LVDCR is used to enable or disable the low-voltage detection circuit, set the detection levels for the LVDR function, enable or disable the LVDR function, and enable or disable generation of an interrupt when the power-supply voltage rises above or falls below the respective levels.

Table 20.1 shows the relationship between the LVDCR settings and select functions. LVDCR should be set according to table 20.1.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 4 | — | All 1 | — | Reserved<br>These bits are always read as 1. Data write is inhibited. |
| 3 | LVDSEL | 1 | R/W | LVDR Detection Level Select<br>0: Reset detection voltage is 2.3 V (typ.)<br>1: Reset detection voltage is 3.6 V (typ.)<br>When the falling or rising voltage detection interrupt is used, reset detection voltage of 2.3 V (typ.) should be used. When only a reset detection interrupt is used, reset detection voltage of 3.6 V (typ.) should be used. This bit is initialized by a LVDR reset. |
| 2 | — | 1 | — | Reserved<br>This bit is always read as 1. Data write is inhibited. |
| 1 | LVDDE | 0 | R/W | Voltage-Fall-Interrupt Enable<br>0: Interrupt on the power-supply voltage falling below the selected detection level disabled<br>1: Interrupt on the power-supply voltage falling below the selected detection level enabled |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | LVDUE | 0 | R/W | Voltage-Rise-Interrupt Enable<br>0: Interrupt on the power-supply voltage rising above the selected detection level disabled<br>1: Interrupt on the power-supply voltage rising above the selected detection level enabled |

**Table 20.1 LVDCR Settings and Select Functions**

| LVDCR Settings | | | Select Functions | | | |
|---|---|---|---|---|---|---|
| LVDSEL | LVDDE | LVDUE | Power-On Reset | LVDR Reset | Low-Voltage-Detection Falling Interrupt | Low-Voltage-Detection Rising Interrupt |
| 1 | 0 | 0 | √ | √ | — | — |
| 0 | 1 | 0 | √ | √ | √ | — |
| 0 | 1 | 1 | √ | √ | √ | √ |

### 20.2.2 Low-Voltage-Detection Status Register (LVDSR)

LVDSR indicates whether the power-supply voltage falls below or rises above the respective specified values.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 2 | — | All 1 | — | Reserved<br>These bits are always read as 1, and cannot be modified. |
| 1 | LVDDF | 0* | R/W | LVD Power-Supply Voltage Fall Flag<br>[Setting condition]<br>When the power-supply voltage falls below Vint (D) (typ. = 3.7 V)<br>[Clearing condition]<br>Writing 0 to this bit after reading it as 1 |
| 0 | LVDUF | 0* | R/W | LVD Power-Supply Voltage Rise Flag<br>[Setting condition]<br>When the power supply voltage falls below Vint (D) while the LVDUE bit in LVDCR is set to 1, then rises above Vint (U) (typ. = 4.0 V) before falling below Vreset1 (typ. = 2.3 V)<br>[Clearing condition]<br>Writing 0 to this bit after reading it as 1 |

Note: * Initialized by LVDR.

# 20.3 Operation

## 20.3.1 Power-On Reset Circuit

Figure 20.3 shows the timing of the operation of the power-on reset circuit. As the power-supply voltage rises, the capacitor which is externally connected to the $\overline{RES}$ pin is gradually charged via the on-chip pull-up resistor (typ. 150 kΩ). Since the level of the $\overline{RES}$ signal is transmitted within this LSI, prescaler S and the entire LSI are in their reset states. When the level of the $\overline{RES}$ signal reaches the threshold level, the prescaler S is released from its reset state and it starts counting. The OVF signal is generated to negate the internal reset signal after the prescaler S has counted 131,072 clock (ϕ) cycles. The noise cancellation circuit of approximately 100 ns is incorporated to prevent the incorrect operation of this LSI by noise on the $\overline{RES}$ signal.

To achieve stable operation of this LSI, the power supply needs to rise to its full level and settles within the specified time. The maximum time required for the power supply to rise and settle after power has been supplied ($t_{PWON}$) is determined by the oscillation frequency ($f_{OSC}$) and capacitance which is connected to $\overline{RES}$ pin ($C_{\overline{RES}}$). If $t_{PWON}$ means the time required to reach 90 % of power supply voltage, the power supply circuit should be designed to satisfy the following formula.

$$t_{PWON}\ (\text{ms}) \leq 90 \times C_{\overline{RES}}\ (\mu\text{F}) + 162/f_{OSC}\ (\text{MHz})$$

($t_{PWON} \leq 3000$ ms, $C_{\overline{RES}} \geq 0.22$ μF, and $f_{OSC} = 10$ in 4-MHz to 10-MHz operation)

Note that the power supply voltage (Vcc) must fall below Vpor = 100 mV and rise after charge on the $\overline{RES}$ pin is removed. To remove charge on the $\overline{RES}$ pin, it is recommended that a diode should be placed near Vcc. If the power supply voltage (Vcc) rises from the point above Vpor, a power-on reset may not occur.

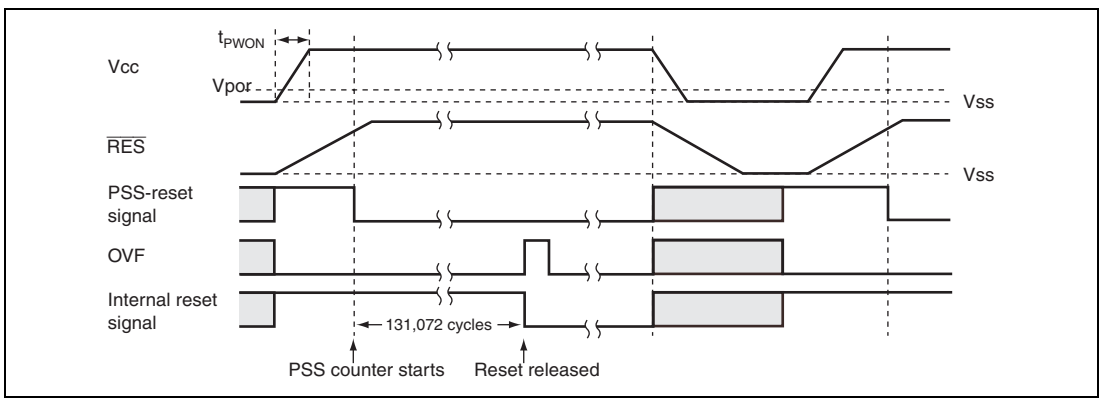



**Figure 20.3 Operational Timing of Power-On Reset Circuit**

### 20.3.2 Low-Voltage Detection Circuit

#### (1) LVDR (Reset by Low Voltage Detect) Circuit

Figure 20.4 shows the timing of the LVDR function. The LVDR is enabled after a power-on reset signal is negated.

When the power-supply voltage falls below the Vreset voltage (typ. = 2.3 V or 3.6 V), the LVDR clears the $\overline{\text{LVDRES}}$ signal to 0, and resets prescaler S. The low-voltage detection reset state remains in place until a power-on reset is generated. When the power-supply voltage rises above the Vreset voltage (typ. = 3.6 V) again, prescaler S starts counting. It counts 131,072 clock (ϕ) cycles, and then releases the internal reset signal. Since the LVDSEL bit in the LVDCR is initialized to 1 at this point, Vreset during Vcc rising remains 3.6 V, even if the LVDSEL bit had been set to 0.

Note that if the power supply voltage (Vcc) falls below $V_{LVDRmin}$ = 1.0 V and then rises from that point, the low-voltage detection reset may not occur.

If the power supply voltage (Vcc) falls below Vpor = 100 mV, a power-on reset occurs.

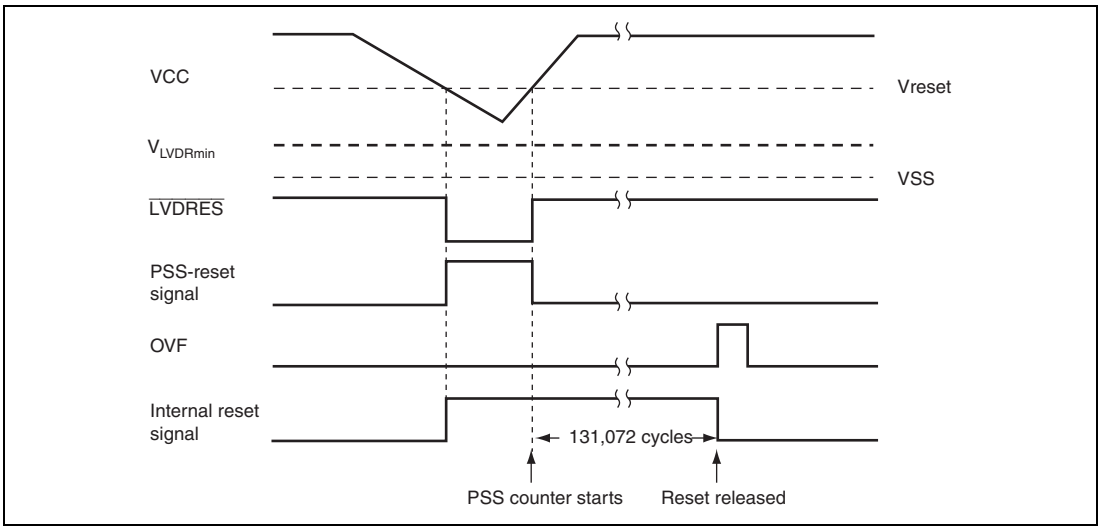



**Figure 20.4 Operational Timing of LVDR Circuit**

### (2) LVDI (Interrupt by Low Voltage Detection) Circuit

Figure 20.5 shows the timing of LVDI functions. To start the LVDI, set the LVDDE and LVDUE bits in LVDCR to 1.

When the power-supply voltage falls below Vint (D) (typ. = 3.7 V) voltage, the LVDI clears the $\overline{\text{LVDINT}}$ signal to 0 and the LVDDF bit in LVDSR is set to 1. If the LVDDE bit is 1 at this time, an IRQ0 interrupt request is simultaneously generated. In this case, the necessary data must be saved in the external EEPROM, etc, and a transition must be made to the standby or subsleep mode. Until this processing is completed, the power supply voltage must be higher than the lower limit of the guaranteed operating voltage.

When the power-supply voltage does not fall below Vreset1 (typ. = 2.3 V) voltage but rises above Vint (U) (typ. = 4.0 V) voltage, the LVDI sets the $\overline{\text{LVDINT}}$ signal to 1. If the LVDUE bit is 1 at this time, the LVDUF bit in LVDSR is set to 1 and an IRQ0 interrupt request is simultaneously generated.

If the power supply voltage (Vcc) falls below Vreset1 (typ. = 2.3 V) voltage, the LVDR function is performed.

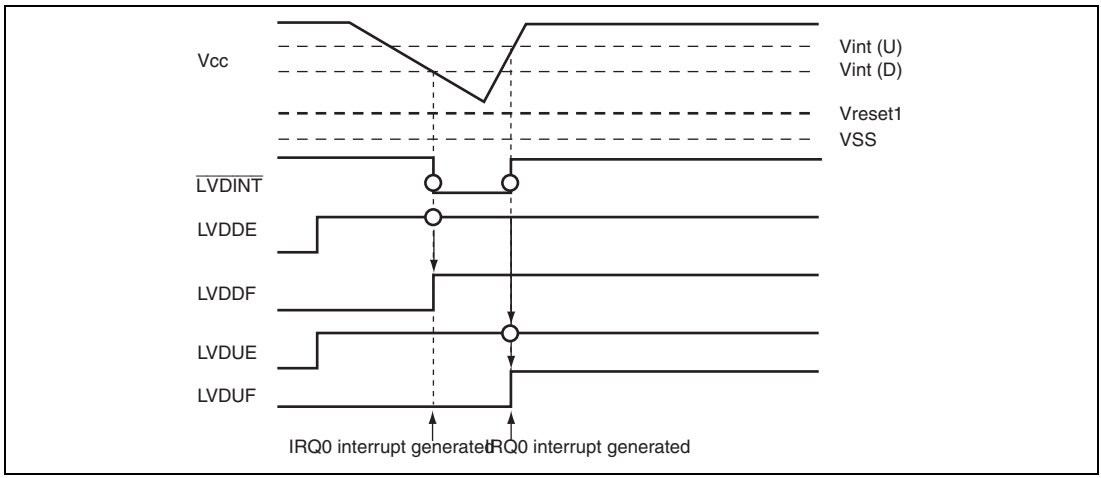



**Figure 20.5 Operational Timing of LVDI Circuit**

# Section 21 Power Supply Circuit

This LSI incorporates an internal power supply step-down circuit. Use of this circuit enables the internal power supply to be fixed at a constant level of approximately 3.0 V, independently of the voltage of the power supply connected to the external $V_{CC}$ pin. As a result, the current consumed when an external power supply is used at 3.0 V or above can be held down to virtually the same low level as when used at approximately 3.0 V. If the external power supply is 3.0 V or below, the internal voltage will be practically the same as the external voltage. It is, of course, also possible to use the same level of external power supply voltage and internal power supply voltage without using the internal power supply step-down circuit.

## 21.1 When Using Internal Power Supply Step-Down Circuit

Connect the external power supply to the $V_{CC}$ pin, and connect a capacitance of approximately 0.1 µF between $V_{CL}$ and $V_{SS}$, as shown in figure 21.1. The internal step-down circuit is made effective simply by adding this external circuit. In the external circuit interface, the external power supply voltage connected to $V_{CC}$ and the GND potential connected to $V_{SS}$ are the reference levels. For example, for port input/output levels, the $V_{CC}$ level is the reference for the high level, and the $V_{SS}$ level is that for the low level. The A/D converter analog power supply is not affected by the internal step-down circuit.

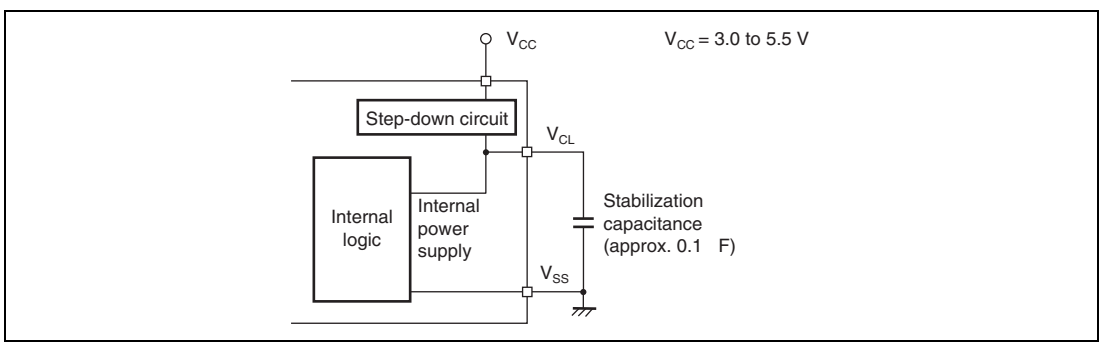



**Figure 21.1 Power Supply Connection when Internal Step-Down Circuit is Used**

## 21.2 When Not Using Internal Power Supply Step-Down Circuit

When the internal power supply step-down circuit is not used, connect the external power supply to the $V_{CL}$ pin and $V_{CC}$ pin, as shown in figure 21.2. The external power supply is then input directly to the internal power supply. The permissible range for the power supply voltage is 3.0 V to 3.6 V. Operation cannot be guaranteed if a voltage outside this range (less than 3.0 V or more than 3.6 V) is input.

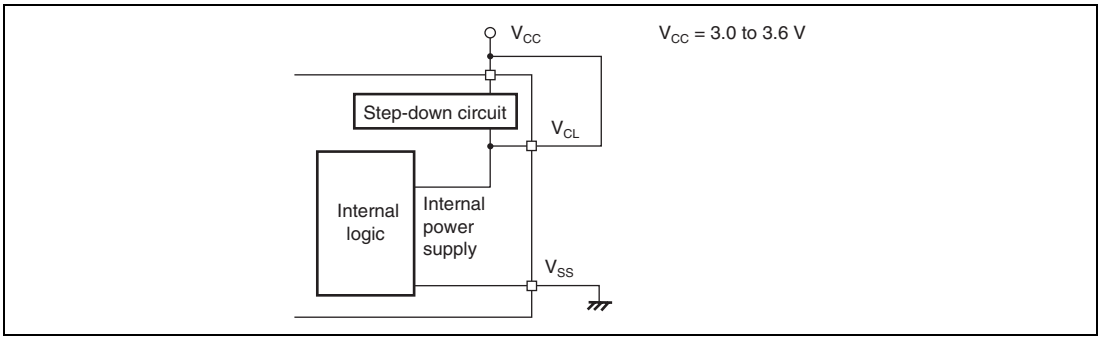



**Figure 21.2 Power Supply Connection when Internal Step-Down Circuit is Not Used**

# Section 22 List of Registers

The register list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register addresses (address order)
- Registers are listed from the lower allocation addresses.
- Registers are classified by functional modules.
- The data bus width is indicated.
- The number of access states is indicated.

2. Register bits
- Bit configurations of the registers are described in the same order as the register addresses.
- Reserved bits are indicated by — in the bit name column.
- When registers consist of 16 bits, bits are described from the MSB side.

3. Register states in each operating mode
- Register states are described in the same order as the register addresses.
- The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

## 22.1 Register Addresses (Address Order)

The data-bus width column indicates the number of bits. The access-state column shows the number of states of the specified basic clock that is required for access to the register.

Note: Access to undefined or reserved addresses is prohibited. Correct operation of the access itself or later operations is not guaranteed when such a register is accessed.

| Register Name | Abbreviation | Bit No. | Address | Module Name | Data Bus Width | Access State |
|---|---|---|---|---|---|---|
| Timer RD counter_0 | TRDCNT_0 | 16 | H'FFF100 | Timer RD (Channel 0) | 16*[1] | 4 |
| General register A_0 | GRA_0 | 16 | H'FFF102 | Timer RD (Channel 0) | 16*[1] | 4 |
| General register B_0 | GRB_0 | 16 | H'FFF104 | Timer RD (Channel 0) | 16*[1] | 4 |
| General register C_0 | GRC_0 | 16 | H'FFF106 | Timer RD (Channel 0) | 16*[1] | 4 |
| General register D_0 | GRD_0 | 16 | H'FFF108 | Timer RD (Channel 0) | 16*[1] | 4 |
| Timer RD counter_1 | TRDCNT_1 | 16 | H'FFF10A | Timer RD (Channel 1) | 16*[1] | 4 |
| General register A_1 | GRA_1 | 16 | H'FFF10C | Timer RD (Channel 1) | 16*[1] | 4 |
| General register B_1 | GRB_1 | 16 | H'FFF10E | Timer RD (Channel 1) | 16*[1] | 4 |
| General register C_1 | GRC_1 | 16 | H'FFF110 | Timer RD (Channel 1) | 16*[1] | 4 |
| General register D_1 | GRD_1 | 16 | H'FFF112 | Timer RD (Channel 1) | 16*[1] | 4 |
| Timer RD counter_2 | TRDCNT_2 | 16 | H'FFF140 | Timer RD (Channel 2) | 16*[1] | 4 |
| General register A_2 | GRA_2 | 16 | H'FFF142 | Timer RD (Channel 2) | 16*[1] | 4 |
| General register B_2 | GRB_2 | 16 | H'FFF144 | Timer RD (Channel 2) | 16*[1] | 4 |
| General register C_2 | GRC_2 | 16 | H'FFF146 | Timer RD (Channel 2) | 16*[1] | 4 |

| Register Name | Abbreviation | Bit No. | Address | Module Name | Data Bus Width | Access State |
|---|---|---|---|---|---|---|
| General register D_2 | GRD_2 | 16 | H'FFF148 | Timer RD (Channel 2) | 16*[1] | 4 |
| Timer RD counter_3 | TRDCNT_3 | 16 | H'FFF14A | Timer RD (Channel 3) | 16*[1] | 4 |
| General register A_3 | GRA_3 | 16 | H'FFF14C | Timer RD (Channel 3) | 16*[1] | 4 |
| General register B_3 | GRB_3 | 16 | H'FFF14E | Timer RD (Channel 3) | 16*[1] | 4 |
| General register C_3 | GRC_3 | 16 | H'FFF150 | Timer RD (Channel 3) | 16*[1] | 4 |
| General register D_3 | GRD_3 | 16 | H'FFF152 | Timer RD (Channel 3) | 16*[1] | 4 |
| Timer RC counter | TRCCNT | 16 | H'FFF180 | Timer RC | 16*[1] | 4 |
| General register A | GRA | 16 | H'FFF182 | Timer RC | 16*[1] | 4 |
| General register B | GRB | 16 | H'FFF184 | Timer RC | 16*[1] | 4 |
| General register C | GRC | 16 | H'FFF186 | Timer RC | 16*[1] | 4 |
| General register D | GRD | 16 | H'FFF188 | Timer RC | 16*[1] | 4 |
| Serial mode register_3 | SMR_3 | 8 | H'FFF600 | SCI3_3 | 8 | 4 |
| Bit rate register_3 | BRR_3 | 8 | H'FFF601 | SCI3_3 | 8 | 4 |
| Serial control register 3_3 | SCR3_3 | 8 | H'FFF602 | SCI3_3 | 8 | 4 |
| Transmit data register_3 | TDR_3 | 8 | H'FFF603 | SCI3_3 | 8 | 4 |
| Serial status register_3 | SSR_3 | 8 | H'FFF604 | SCI3_3 | 8 | 4 |
| Receive data register_3 | RDR_3 | 8 | H'FFF605 | SCI3_3 | 8 | 4 |
| Serial mode control register_3 | SMCR_3 | 8 | H'FFF608 | SCI3_3 | 8 | 4 |
| A/D data register A | ADDRA | 16 | H'FFF610 | A/D converter | 8 | 4 |
| A/D data register B | ADDRB | 16 | H'FFF612 | A/D converter | 8 | 4 |
| A/D data register C | ADDRC | 16 | H'FFF614 | A/D converter | 8 | 4 |
| A/D data register D | ADDRD | 16 | H'FFF616 | A/D converter | 8 | 4 |
| A/D control/status register | ADCSR | 8 | H'FFF618 | A/D converter | 8 | 4 |
| A/D control register | ADCR | 8 | H'FFF619 | A/D converter | 8 | 4 |
| Port data register D | PDRD | 8 | H'FFF624 | I/O port | 8 | 4 |
| Port data register E | PDRE | 8 | H'FFF625 | I/O port | 8 | 4 |
| Port data register F | PDRF | 8 | H'FFF626 | I/O port | 8 | 4 |

| Register Name | Abbreviation | Bit No. | Address | Module Name | Data Bus Width | Access State |
|---|---|---|---|---|---|---|
| Port data register G | PDRG | 8 | H'FFF627 | I/O port | 8 | 4 |
| Port data register H | PDRH | 8 | H'FFF628 | I/O port | 8 | 4 |
| Port data register J | PDRJ | 8 | H'FFF629 | I/O port | 8 | 4 |
| Port mode register F | PMRF | 8 | H'FFF630 | I/O port | 8 | 4 |
| Port mode register G | PMRG | 8 | H'FFF631 | I/O port | 8 | 4 |
| Port control register D | PCRD | 8 | H'FFF634 | I/O port | 8 | 4 |
| Port control register E | PCRE | 8 | H'FFF635 | I/O port | 8 | 4 |
| Port control register G | PCRG | 8 | H'FFF637 | I/O port | 8 | 4 |
| Port control register H | PCRH | 8 | H'FFF638 | I/O port | 8 | 4 |
| Port control register J | PCRJ | 8 | H'FFF639 | I/O port | 8 | 4 |
| Module standby control register 4 | MSTCR4 | 8 | H'FFF64F | Power-down modes | 8 | 4 |
| Timer RD control register_0 | TRDCR_0 | 8 | H'FFF654 | Timer RD (Channel 0) | 8 | 4 |
| Timer RD I/O control register A_0 | TRDIORA_0 | 8 | H'FFF655 | Timer RD (Channel 0) | 8 | 4 |
| Timer RD I/O control register C_0 | TRDIORC_0 | 8 | H'FFF656 | Timer RD (Channel 0) | 8 | 4 |
| Timer RD status register_0 | TRDSR_0 | 8 | H'FFF657 | Timer RD (Channel 0) | 8 | 4 |
| Timer RD interrupt enable register_0 | TRDIER_0 | 8 | H'FFF658 | Timer RD (Channel 0) | 8 | 4 |
| PWM mode output level control register_0 | POCR_0 | 8 | H'FFF659 | Timer RD (Channel 0) | 8 | 4 |
| Timer RD digital filtering function select register_0 | TRDDF_0 | 8 | H'FFF65A | Timer RD (Channel 0) | 8 | 4 |
| Timer RD control register_1 | TRDCR_1 | 8 | H'FFF65B | Timer RD (Channel 1) | 8 | 4 |
| Timer RD I/O control register A_1 | TRDIORA_1 | 8 | H'FFF65C | Timer RD (Channel 1) | 8 | 4 |
| Timer RD I/O control register C_1 | TDRIORC_1 | 8 | H'FFF65D | Timer RD (Channel 1) | 8 | 4 |
| Timer RD status register_1 | TRDSR_1 | 8 | H'FFF65E | Timer RD (Channel 1) | 8 | 4 |

| Register Name | Abbreviation | Bit No. | Address | Module Name | Data Bus Width | Access State |
|---|---|---|---|---|---|---|
| Timer RD interrupt enable register_1 | TRDIER_1 | 8 | H'FFF65F | Timer RD (Channel 1) | 8 | 4 |
| PWM mode output level control register_1 | POCR_1 | 8 | H'FFF660 | Timer RD (Channel 1) | 8 | 4 |
| Timer RD digital filtering function select register_1 | TRDDF_1 | 8 | H'FFF661 | Timer RD (Channel 1) | 8 | 4 |
| Timer RD start register_01 | TRDSTR_01 | 8 | H'FFF662 | Timer RD (Channel 0 and 1 common) | 8 | 4 |
| Timer RD mode register_01 | TRDMDR_01 | 8 | H'FFF663 | Timer RD (Channel 0 and 1 common) | 8 | 4 |
| Timer RD PWM mode register_01 | TRDPMR_01 | 8 | H'FFF664 | Timer RD (Channel 0 and 1 common) | 8 | 4 |
| Timer RD function control register_01 | TRDFCR_01 | 8 | H'FFF665 | Timer RD (Channel 0 and 1 common) | 8 | 4 |
| Timer output master enable register 1_01 | TRDOER1_01 | 8 | H'FFF666 | Timer RD (Channel 0 and 1 common) | 8 | 4 |
| Timer output master enable register 2_01 | TRDOER2_01 | 8 | H'FFF667 | Timer RD (Channel 0 and 1 common) | 8 | 4 |
| Timer RD output control register_01 | TRDOCR_01 | 8 | H'FFF668 | Timer RD (Channel 0 and 1 common) | 8 | 4 |
| Timer RD control register_2 | TRDCR_2 | 8 | H'FFF694 | Timer RD (Channel 2) | 8 | 4 |
| Timer RD I/O control register A_2 | TRDIORA_2 | 8 | H'FFF695 | Timer RD (Channel 2) | 8 | 4 |
| Timer RD I/O control register C_2 | TDRIORC_2 | 8 | H'FFF696 | Timer RD (Channel 2) | 8 | 4 |
| Timer RD status register_2 | TRDSR_2 | 8 | H'FFF697 | Timer RD (Channel 2) | 8 | 4 |
| Timer RD interrupt enable register_2 | TRDIER_2 | 8 | H'FFF698 | Timer RD (Channel 2) | 8 | 4 |

| Register Name | Abbreviation | Bit No. | Address | Module Name | Data Bus Width | Access State |
|---|---|---|---|---|---|---|
| PWM mode output level control register_2 | POCR_2 | 8 | H'FFF699 | Timer RD (Channel 2) | 8 | 4 |
| Timer RD digital filtering function select register_2 | TRDDF_2 | 8 | H'FFF69A | Timer RD (Channel 2) | 8 | 4 |
| Timer RD control register_3 | TRDCR_3 | 8 | H'FFF69B | Timer RD (Channel 3) | 8 | 4 |
| Timer RD I/O control register A_3 | TRDIORA_3 | 8 | H'FFF69C | Timer RD (Channel 3) | 8 | 4 |
| Timer RD I/O control register C_3 | TDRIORC_3 | 8 | H'FFF69D | Timer RD (Channel 3) | 8 | 4 |
| Timer RD status register_3 | TRDSR_3 | 8 | H'FFF69E | Timer RD (Channel 3) | 8 | 4 |
| Timer RD interrupt enable register_3 | TRDIER_3 | 8 | H'FFF69F | Timer RD (Channel 3) | 8 | 4 |
| PWM mode output level control register_3 | POCR_3 | 8 | H'FFF6A0 | Timer RD (Channel 3) | 8 | 4 |
| Timer RD digital filtering function select register_3 | TRDDF_3 | 8 | H'FFF6A1 | Timer RD (Channel 3) | 8 | 4 |
| Timer RD start register_23 | TRDSTR_23 | 8 | H'FFF6A2 | Timer RD (Channel 2 and 3 common) | 8 | 4 |
| Timer RD mode register_23 | TRDMDR_23 | 8 | H'FFF6A3 | Timer RD (Channel 2 and 3 common) | 8 | 4 |
| Timer RD PWM mode register_23 | TRDPMR_23 | 8 | H'FFF6A4 | Timer RD (Channel 2 and 3 common) | 8 | 4 |
| Timer RD function control register_23 | TRDFCR_23 | 8 | H'FFF6A5 | Timer RD (Channel 2 and 3 common) | 8 | 4 |
| Timer RD output master enable register 1_23 | TRDOER1_23 | 8 | H'FFF6A6 | Timer RD (Channel 2 and 3 common) | 8 | 4 |
| Timer RD output master enable register 2_23 | TRDOER2_23 | 8 | H'FFF6A7 | Timer RD (Channel 2 and 3 common) | 8 | 4 |

| Register Name | Abbreviation | Bit No. | Address | Module Name | Data Bus Width | Access State |
|---|---|---|---|---|---|---|
| Timer RD output control register_23 | TRDOCR_23 | 8 | H'FFF6A8 | Timer RD (Channel 2 and 3 common) | 8 | 4 |
| Timer RC mode register | TRCMR | 8 | H'FFF6CA | Timer RC | 8 | 4 |
| Timer RC control register 1 | TRCCR1 | 8 | H'FFF6CB | Timer RC | 8 | 4 |
| Timer RC interrupt enable register | TRCIER | 8 | H'FFF6CC | Timer RC | 8 | 4 |
| Timer RC status register | TRCSR | 8 | H'FFF6CD | Timer RC | 8 | 4 |
| Timer RC I/O control register 0 | TRCIOR0 | 8 | H'FFF6CE | Timer RC | 8 | 4 |
| Timer RC I/O control register 1 | TRCIOR1 | 8 | H'FFF6CF | Timer RC | 8 | 4 |
| Timer RC control register 2 | TRCCR2 | 8 | H'FFF6D0 | Timer RC | 8 | 4 |
| Timer RC digital filtering function select register | TRCDF | 8 | H'FFF6D1 | Timer RC | 8 | 4 |
| Timer RC output enable register | TRCOER | 8 | H'FFF6D2 | Timer RC | 8 | 4 |
| Second data register/free running counter data register | RSECDR | 8 | H'FFF728 | RTC | 8 | 2 |
| Minute data register | RMINDR | 8 | H'FFF729 | RTC | 8 | 2 |
| Hour data register | RHRDR | 8 | H'FFF72A | RTC | 8 | 2 |
| Day-of-week data register | RWKDR | 8 | H'FFF72B | RTC | 8 | 2 |
| RTC control register 1 | RTCCR1 | 8 | H'FFF72C | RTC | 8 | 2 |
| RTC control register 2 | RTCCR2 | 8 | H'FFF72D | RTC | 8 | 2 |
| Clock source select register | RTCCSR | 8 | H'FFF72F | RTC | 8 | 2 |
| Low-voltage-detection control register | LVDCR | 8 | H'FFF730 | LVD | 8 | 2 |
| Low-voltage-detection status register | LVDSR | 8 | H'FFF731 | LVD | 8 | 2 |
| Clock control status register | CKCSR | 8 | H'FFF734 | Clock pulse generator | 8 | 2 |
| RC control register | RCCR | 8 | H'FFF738 | On-chip oscillator | 8 | 2 |
| RC trimming data protect register | RCTRMDPR | 8 | H'FFF739 | On-chip oscillator | 8 | 2 |
| RC trimming register | RCTRMDR | 8 | H'FFF73A | On-chip oscillator | 8 | 2 |
| Interrupt control register A | ICRA | 8 | H'FFF73C | Interrupt | 8 | 2 |

| Register Name | Abbreviation | Bit No. | Address | Module Name | Data Bus Width | Access State |
|---|---|---|---|---|---|---|
| Interrupt control register B | ICRB | 8 | H'FFF73D | Interrupt | 8 | 2 |
| Interrupt control register C | ICRC | 8 | H'FFF73E | Interrupt | 8 | 2 |
| Interrupt control register D | ICRD | 8 | H'FFF73F | Interrupt | 8 | 2 |
| Serial mode register_2 | SMR_2 | 8 | H'FFF740 | SCI3_2 | 8 | 3 |
| Bit rate register_2 | BRR_2 | 8 | H'FFF741 | SCI3_2 | 8 | 3 |
| Serial control register 3_2 | SCR3_2 | 8 | H'FFF742 | SCI3_2 | 8 | 3 |
| Transmit data register_2 | TDR_2 | 8 | H'FFF743 | SCI3_2 | 8 | 3 |
| Serial status register_2 | SSR_2 | 8 | H'FFF744 | SCI3_2 | 8 | 3 |
| Receive data register_2 | RDR_2 | 8 | H'FFF745 | SCI3_2 | 8 | 3 |
| $I^2C$ bus control register 1 | ICCR1 | 8 | H'FFF748 | IIC2 | 8 | 2 |
| $I^2C$ bus control register 2 | ICCR2 | 8 | H'FFF749 | IIC2 | 8 | 2 |
| $I^2C$ bus mode register | ICMR | 8 | H'FFF74A | IIC2 | 8 | 2 |
| $I^2C$ bus interrupt enable register | ICIER | 8 | H'FFF74B | IIC2 | 8 | 2 |
| $I^2C$ bus status register | ICSR | 8 | H'FFF74C | IIC2 | 8 | 2 |
| Slave address register | SAR | 8 | H'FFF74D | IIC2 | 8 | 2 |
| $I^2C$ bus transmit data register | ICDRT | 8 | H'FFF74E | IIC2 | 8 | 2 |
| $I^2C$ bus receive data register | ICDRR | 8 | H'FFF74F | IIC2 | 8 | 2 |
| Timer mode register B1 | TMB1 | 8 | H'FFF760 | Timer B1 | 8 | 2 |
| Timer counter B1 | TCB1 | 8 | H'FFF761 | Timer B1 | 8 | 2 |
| Timer load register B1 | TLB1 | 8 | H'FFF761 | Timer B1 | 8 | 2 |
| Flash memory control register 1 | FLMCR1 | 8 | H'FFFF90 | ROM | 8 | 2 |
| Flash memory control register 2 | FLMCR2 | 8 | H'FFFF91 | ROM | 8 | 2 |
| Flash memory power control register | FLPWCR | 8 | H'FFFF92 | ROM | 8 | 2 |
| Erase block register 1 | EBR1 | 8 | H'FFFF93 | ROM | 8 | 2 |
| Flash memory enable register | FENR | 8 | H'FFFF9B | ROM | 8 | 2 |
| Timer control register V0 | TCRV0 | 8 | H'FFFFA0 | Timer V | 8 | 3 |
| Timer control/status register V | TCSRV | 8 | H'FFFFA1 | Timer V | 8 | 3 |
| Time constant register A | TCORA | 8 | H'FFFFA2 | Timer V | 8 | 3 |
| Time constant register B | TCORB | 8 | H'FFFFA3 | Timer V | 8 | 3 |
| Timer counter V | TCNTV | 8 | H'FFFFA4 | Timer V | 8 | 3 |

| Register Name | Abbreviation | Bit No. | Address | Module Name | Data Bus Width | Access State |
|---|---|---|---|---|---|---|
| Timer control register V1 | TCRV1 | 8 | H'FFFFA5 | Timer V | 8 | 3 |
| Serial mode register | SMR | 8 | H'FFFFA8 | SCI3 | 8 | 3 |
| Bit rate register | BRR | 8 | H'FFFFA9 | SCI3 | 8 | 3 |
| Serial control register 3 | SCR3 | 8 | H'FFFFAA | SCI3 | 8 | 3 |
| Transmit data register | TDR | 8 | H'FFFFAB | SCI3 | 8 | 3 |
| Serial status register | SSR | 8 | H'FFFFAC | SCI3 | 8 | 3 |
| Receive data register | RDR | 8 | H'FFFFAD | SCI3 | 8 | 3 |
| PWM data register L | PWDRL | 8 | H'FFFFBC | 14-bit PWM | 8 | 2 |
| PWM data register U | PWDRU | 8 | H'FFFFBD | 14-bit PWM | 8 | 2 |
| PWM control register | PWCR | 8 | H'FFFFBE | 14-bit PWM | 8 | 2 |
| Timer control/status register WD | TCSRWD | 8 | H'FFFFC0 | WD*[2] | 8 | 2 |
| Timer counter WD | TCWD | 8 | H'FFFFC1 | WD*[2] | 8 | 2 |
| Timer mode register WD | TMWD | 8 | H'FFFFC2 | WD*[2] | 8 | 2 |
| Address break control register | ABRKCR | 8 | H'FFFFC8 | Address break | 8 | 2 |
| Address break status register | ABRKSR | 8 | H'FFFFC9 | Address break | 8 | 2 |
| Break address register H | BARH | 8 | H'FFFFCA | Address break | 8 | 2 |
| Break address register L | BARL | 8 | H'FFFFCB | Address break | 8 | 2 |
| Break data register H | BDRH | 8 | H'FFFFCC | Address break | 8 | 2 |
| Break data register L | BDRL | 8 | H'FFFFCD | Address break | 8 | 2 |
| Break address register E | BARE | 8 | H'FFFFCF | Address break | 8 | 2 |
| Port pull-up control register 1 | PUCR1 | 8 | H'FFFFD0 | I/O Port | 8 | 2 |
| Port pull-up control register 5 | PUCR5 | 8 | H'FFFFD1 | I/O Port | 8 | 2 |
| Port data register 1 | PDR1 | 8 | H'FFFFD4 | I/O Port | 8 | 2 |
| Port data register 2 | PDR2 | 8 | H'FFFFD5 | I/O Port | 8 | 2 |
| Port data register 3 | PDR3 | 8 | H'FFFFD6 | I/O Port | 8 | 2 |
| Port data register 5 | PDR5 | 8 | H'FFFFD8 | I/O Port | 8 | 2 |
| Port data register 7 | PDR7 | 8 | H'FFFFDA | I/O Port | 8 | 2 |
| Port data register 8 | PDR8 | 8 | H'FFFFDB | I/O Port | 8 | 2 |
| Port data register C | PDRC | 8 | H'FFFFDE | I/O Port | 8 | 2 |
| Port mode register 1 | PMR1 | 8 | H'FFFFE0 | I/O Port | 8 | 2 |
| Port mode register 5 | PMR5 | 8 | H'FFFFE1 | I/O Port | 8 | 2 |

| Register Name | Abbreviation | Bit No. | Address | Module Name | Data Bus Width | Access State |
|---|---|---|---|---|---|---|
| Port mode register 3 | PMR3 | 8 | H'FFFFE2 | I/O Port | 8 | 2 |
| Port control register 1 | PCR1 | 8 | H'FFFFE4 | I/O Port | 8 | 2 |
| Port control register 2 | PCR2 | 8 | H'FFFFE5 | I/O Port | 8 | 2 |
| Port control register 3 | PCR3 | 8 | H'FFFFE6 | I/O Port | 8 | 2 |
| Port control register 5 | PCR5 | 8 | H'FFFFE8 | I/O Port | 8 | 2 |
| Port control register 7 | PCR7 | 8 | H'FFFFEA | I/O Port | 8 | 2 |
| Port control register 8 | PCR8 | 8 | H'FFFFEB | I/O Port | 8 | 2 |
| Port control register C | PCRC | 8 | H'FFFFEE | I/O Port | 8 | 2 |
| System control register 3 | SYSCR3 | 8 | H'FFFFEF | Power-down modes | 8 | 2 |
| System control register 1 | SYSCR1 | 8 | H'FFFFF0 | Power-down modes | 8 | 2 |
| System control register 2 | SYSCR2 | 8 | H'FFFFF1 | Power-down modes | 8 | 2 |
| Interrupt edge select register 1 | IEGR1 | 8 | H'FFFFF2 | Interrupt | 8 | 2 |
| Interrupt edge select register 2 | IEGR2 | 8 | H'FFFFF3 | Interrupt | 8 | 2 |
| Interrupt enable register 1 | IENR1 | 8 | H'FFFFF4 | Interrupt | 8 | 2 |
| Interrupt enable register 2 | IENR2 | 8 | H'FFFFF5 | Interrupt | 8 | 2 |
| Interrupt flag register 1 | IRR1 | 8 | H'FFFFF6 | Interrupt | 8 | 2 |
| Interrupt flag register 2 | IRR2 | 8 | H'FFFFF7 | Interrupt | 8 | 2 |
| Wakeup interrupt flag register | IWPR | 8 | H'FFFFF8 | Interrupt | 8 | 2 |
| Module standby control register 1 | MSTCR1 | 8 | H'FFFFF9 | Power-down modes | 8 | 2 |
| Module standby control register 2 | MSTCR2 | 8 | H'FFFFFA | Power-down modes | 8 | 2 |

Notes: 1. These registers can be accessed by word size only.
2. WDT: Watchdog timer

## 22.2 Register Bits

The addresses and bit names of the registers in the on-chip peripheral modules are listed below. The 16-bit register is indicated in two rows, 8 bits for each row.

| Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module Name |
|---|---|---|---|---|---|---|---|---|---|
| TRDCNT_0 | TCNT0H7 | TCNT0H6 | TCNT0H5 | TCNT0H4 | TCNT0H3 | TCNT0H2 | TCNT0H1 | TCNT0H0 | Timer RD (Channel 0) |
| | TCNT0L7 | TCNT0L6 | TCNT0L5 | TCNT0L4 | TCNT0L3 | TCNT0L2 | TCNT0L1 | TCNT0L0 | |
| GRA_0 | GRA0H7 | GRA0H6 | GRA0H5 | GRA0H4 | GRA0H3 | GRA0H2 | GRA0H1 | GRA0H0 | |
| | GRA0L7 | GRA0L6 | GRA0L5 | GRA0L4 | GRA0L3 | GRA0L2 | GRA0L1 | GRA0L0 | |
| GRB_0 | GRB0H7 | GRB0H6 | GRB0H5 | GRB0H4 | GRB0H3 | GRB0H2 | GRB0H1 | GRB0H0 | |
| | GRB0L7 | GRB0L6 | GRB0L5 | GRB0L4 | GRB0L3 | GRB0L2 | GRB0L1 | GRB0L0 | |
| GRC_0 | GRC0H7 | GRC0H6 | GRC0H5 | GRC0H4 | GRC0H3 | GRC0H2 | GRC0H1 | GRC0H0 | |
| | GRC0L7 | GRC0L6 | GRC0L5 | GRC0L4 | GRC0L3 | GRC0L2 | GRC0L1 | GRC0L0 | |
| GRD_0 | GRD0H7 | GRD0H6 | GRD0H5 | GRD0H4 | GRD0H3 | GRD0H2 | GRD0H1 | GRD0H0 | |
| | GRD0L7 | GRD0L6 | GRD0L5 | GRD0L4 | GRD0L3 | GRD0L2 | GRD0L1 | GRD0L0 | |
| TRDCNT_1 | TCNT1H7 | TCNT1H6 | TCNT1H5 | TCNT1H4 | TCNT1H3 | TCNT1H2 | TCNT1H1 | TCNT1H0 | Timer RD (Channel 1) |
| | TCNT1L7 | TCNT1L6 | TCNT1L5 | TCNT1L4 | TCNT1L3 | TCNT1L2 | TCNT1L1 | TCNT1L0 | |
| GRA_1 | GRA1H7 | GRA1H6 | GRA1H5 | GRA1H4 | GRA1H3 | GRA1H2 | GRA1H1 | GRA1H0 | |
| | GRA1L7 | GRA1L6 | GRA1L5 | GRA1L4 | GRA1L3 | GRA1L2 | GRA1L1 | GRA1L0 | |
| GRB_1 | GRB1H7 | GRB1H6 | GRB1H5 | GRB1H4 | GRB1H3 | GRB1H2 | GRB1H1 | GRB1H0 | |
| | GRB1L7 | GRB1L6 | GRB1L5 | GRB1L4 | GRB1L3 | GRB1L2 | GRB1L1 | GRB1L0 | |
| GRC_1 | GRC1H7 | GRC1H6 | GRC1H5 | GRC1H4 | GRC1H3 | GRC1H2 | GRC1H1 | GRC1H0 | |
| | GRC1L7 | GRC1L6 | GRC1L5 | GRC1L4 | GRC1L3 | GRC1L2 | GRC1L1 | GRC1L0 | |
| GRD_1 | GRD1H7 | GRD1H6 | GRD1H5 | GRD1H4 | GRD1H3 | GRD1H2 | GRD1H1 | GRD1H0 | |
| | GRD1L7 | GRD1L6 | GRD1L5 | GRD1L4 | GRD1L3 | GRD1L2 | GRD1L1 | GRD1L0 | |
| TRDCNT_2 | TCNT2H7 | TCNT2H6 | TCNT2H5 | TCNT2H4 | TCNT2H3 | TCNT2H2 | TCNT2H1 | TCNT2H0 | Timer RD (Channel 2) |
| | TCNT2L7 | TCNT2L6 | TCNT2L5 | TCNT2L4 | TCNT2L3 | TCNT2L2 | TCNT2L1 | TCNT2L0 | |
| GRA_2 | GRA2H7 | GRA2H6 | GRA2H5 | GRA2H4 | GRA2H3 | GRA2H2 | GRA2H1 | GRA2H0 | |
| | GRA2L7 | GRA2L6 | GRA2L5 | GRA2L4 | GRA2L3 | GRA2L2 | GRA2L1 | GRA2L0 | |
| GRB_2 | GRB2H7 | GRB2H6 | GRB2H5 | GRB2H4 | GRB2H3 | GRB2H2 | GRB2H1 | GRB2H0 | |
| | GRB2L7 | GRB2L6 | GRB2L5 | GRB2L4 | GRB2L3 | GRB2L2 | GRB2L1 | GRB2L0 | |

| Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module Name |
|---|---|---|---|---|---|---|---|---|---|
| GRC_2 | GRC2H7 | GRC2H6 | GRC2H5 | GRC2H4 | GRC2H3 | GRC2H2 | GRC2H1 | GRC2H0 | Timer RD (Channel 2) |
| | GRC2L7 | GRC2L6 | GRC2L5 | GRC2L4 | GRC2L3 | GRC2L2 | GRC2L1 | GRC2L0 | |
| GRD_2 | GRD2H7 | GRD2H6 | GRD2H5 | GRD2H4 | GRD2H3 | GRD2H2 | GRD2H1 | GRD2H0 | |
| | GRD2L7 | GRD2L6 | GRD2L5 | GRD2L4 | GRD2L3 | GRD2L2 | GRD2L1 | GRD2L0 | |
| TRDCNT_3 | TCNT3H7 | TCNT3H6 | TCNT3H5 | TCNT3H4 | TCNT3H3 | TCNT3H2 | TCNT3H1 | TCNT3H0 | Timer RD (Channel 3) |
| | TCNT3L7 | TCNT3L6 | TCNT3L5 | TCNT3L4 | TCNT3L3 | TCNT3L2 | TCNT3L1 | TCNT3L0 | |
| GRA_3 | GRA3H7 | GRA3H6 | GRA3H5 | GRA3H4 | GRA3H3 | GRA3H2 | GRA3H1 | GRA3H0 | |
| | GRA3L7 | GRA3L6 | GRA3L5 | GRA3L4 | GRA3L3 | GRA3L2 | GRA3L1 | GRA3L0 | |
| GRB_3 | GRB3H7 | GRB3H6 | GRB3H5 | GRB3H4 | GRB3H3 | GRB3H2 | GRB3H1 | GRB3H0 | |
| | GRB3L7 | GRB3L6 | GRB3L5 | GRB3L4 | GRB3L3 | GRB3L2 | GRB3L1 | GRB3L0 | |
| GRC_3 | GRC3H7 | GRC3H6 | GRC3H5 | GRC3H4 | GRC3H3 | GRC3H2 | GRC3H1 | GRC3H0 | |
| | GRC3L7 | GRC3L6 | GRC3L5 | GRC3L4 | GRC3L3 | GRC3L2 | GRC3L1 | GRC3L0 | |
| GRD_3 | GRD3H7 | GRD3H6 | GRD3H5 | GRD3H4 | GRD3H3 | GRD3H2 | GRD3H1 | GRD3H0 | |
| | GRD3L7 | GRD3L6 | GRD3L5 | GRD3L4 | GRD3L3 | GRD3L2 | GRD3L1 | GRD3L0 | |
| TRCCNT | TCNTH7 | TCNTH6 | TCNTH5 | TCNTH4 | TCNTH3 | TCNTH2 | TCNTH1 | TCNTH0 | Timer RC |
| | TCNTL7 | TCNTL6 | TCNTL5 | TCNTL4 | TCNTL3 | TCNTL2 | TCNTL1 | TCNTL0 | |
| GRA | GRAH7 | GRAH6 | GRAH5 | GRAH4 | GRAH3 | GRAH2 | GRAH1 | GRAH0 | |
| | GRAL7 | GRAL6 | GRAL5 | GRAL4 | GRAL3 | GRAL2 | GRAL1 | GRAL0 | |
| GRB | GRBH7 | GRBH6 | GRBH5 | GRBH4 | GRBH3 | GRBH2 | GRBH1 | GRBH0 | |
| | GRBL7 | GRBL6 | GRBL5 | GRBL4 | GRBL3 | GRBL2 | GRBL1 | GRBL0 | |
| GRC | GRCH7 | GRCH6 | GRCH5 | GRCH4 | GRCH3 | GRCH2 | GRCH1 | GRCH0 | |
| | GRCL7 | GRCL6 | GRCL5 | GRCL4 | GRCL3 | GRCL2 | GRCL1 | GRCL0 | |
| GRD | GRDH7 | GRDH6 | GRDH5 | GRDH4 | GRDH3 | GRDH2 | GRDH1 | GRDH0 | |
| | GRDL7 | GRDL6 | GRDL5 | GRDL4 | GRDL3 | GRDL2 | GRDL1 | GRDL0 | |
| SMR_3 | COM | CHR | PE | PM | STOP | MP | CKS1 | CKS0 | SCI3_3 |
| BRR_3 | BRR7 | BRR6 | BRR5 | BRR4 | BRR3 | BRR2 | BRR1 | BRR0 | |
| SCR3_3 | TIE | RIE | TE | RE | MPIE | TEIE | CKE1 | CKE0 | |
| TDR_3 | TDR7 | TDR6 | TDR5 | TDR4 | TDR3 | TDR2 | TDR1 | TDR0 | |
| SSR_3 | TDRE | RDRF | OER | FER | PER | TEND | MPBR | MPBT | |
| RDR_3 | RDR7 | RDR6 | RDR5 | RDR4 | RDR3 | RDR2 | RDR1 | RDR0 | |
| SMCR_3 | — | — | — | — | — | NFEN_3 | TXD_3 | MSTS3_3 | |

| Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module Name |
|---|---|---|---|---|---|---|---|---|---|
| ADDRA | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | A/D converter |
| | AD1 | AD0 | — | — | — | — | — | — | |
| ADDRB | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | |
| | AD1 | AD0 | — | — | — | — | — | — | |
| ADDRC | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | |
| | AD1 | AD0 | — | — | — | — | — | — | |
| ADDRD | AD9 | AD8 | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | |
| | AD1 | AD0 | — | — | — | — | — | — | |
| ADCSR | ADF | ADIE | ADST | SCAN | CKS | CH2 | CH1 | CH0 | |
| ADCR | TRGE | — | — | — | — | — | — | CH3 | |
| PDRD | PD7 | PD6 | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 | I/O port |
| PDRE | PE7 | PE6 | PE5 | PE4 | PE3 | PE2 | PE1 | PE0 | |
| PDRF | PF7 | PF6 | PF5 | PF4 | PF3 | PF2 | PF1 | PF0 | |
| PDRG | PG7 | PG6 | PG5 | PG4 | PG3 | PG2 | PG1 | PG0 | |
| PDRH | PH7 | PH6 | PH5 | PH4 | PH3 | PH2 | PH1 | PH0 | |
| PDRJ | — | — | — | — | — | — | PJ1 | PJ0 | |
| PMRF | — | — | — | — | — | — | — | PMRF0 | |
| PMRG | PMRG7 | PMRG6 | PMRG5 | — | PMRG3 | PMRG2 | PMRG1 | PMRG0 | |
| PCRD | PCRD7 | PCRD6 | PCRD5 | PCRD4 | PCRD3 | PCRD2 | PCRD1 | PCRD0 | |
| PCRE | PCRE7 | PCRE6 | PCRE5 | PCRE4 | PCRE3 | PCRE2 | PCRE1 | PCRE0 | |
| PCRG | PCRG7 | PCRG6 | PCRG5 | PCRG4 | PCRG3 | PCRG2 | PCRG1 | PCRG0 | |
| PCRH | PCRH7 | PCRH6 | PCRH5 | PCRH4 | PCRH3 | PCRH2 | PCRH1 | PCRH0 | |
| PCRJ | — | — | — | — | — | — | PCRJ1 | PCRJ0 | |
| MSTCR4 | MSTTRC | MSTAD | MSTTRD0 | MSTTRD1 | — | — | — | — | Power-down modes |
| TRDCR_0 | CCLR2 | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TPSC2 | TPSC1 | TPSC0 | Timer RD (Channel 0) |
| TRDIORA_0 | — | IOB2 | IOB1 | IOB0 | — | IOA2 | IOA1 | IOA0 | |
| TRDIORC_0 | IOD3 | IOD2 | IOD1 | IOD0 | IOC3 | IOC2 | IOC1 | IOC0 | |
| TRDSR_0 | — | — | — | OVF | IMFD | IMFC | IMFB | IMFA | |
| TRDIER_0 | — | — | — | OVIE | IMIED | IMIEC | IMIEB | IMIEA | |
| POCR_0 | — | — | — | — | — | POLD | POLC | POLB | |
| TRDDF_0 | DFCK1 | DFCK0 | — | — | DFD | DFC | DFB | DFA | |

| Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module Name |
|---|---|---|---|---|---|---|---|---|---|
| TRDCR_1 | CCLR2 | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TPSC2 | TPSC1 | TPSC0 | Timer RD (Channel 1) |
| TRDIORA_1 | — | IOB2 | IOB1 | IOB0 | — | IOA2 | IOA1 | IOA0 | |
| TRDIORC_1 | IOD3 | IOD2 | IOD1 | IOD0 | IOC3 | IOC2 | IOC1 | IOC0 | |
| TRDSR_1 | — | — | UDF | OVF | IMFD | IMFC | IMFB | IMFA | |
| TRDIER_1 | — | — | — | OVIE | IMIED | IMIEC | IMIEB | IMIEA | |
| POCR_1 | — | — | — | — | — | POLD | POLC | POLB | |
| TRDDF_1 | DFCK1 | DFCK0 | — | — | DFD | DFC | DFB | DFA | |
| TRDSTR_01 | — | — | — | — | CSTPN1 | CSTPN0 | STR1 | STR0 | Timer RD (Channel 0 and 1 common) |
| TRDMDR_01 | BFD1 | BFC1 | BFD0 | BFC0 | — | — | — | SYNC | |
| TRDPMR_01 | — | PWMD1 | PWMC1 | PWMB1 | — | PWMD0 | PWMC0 | PWMB0 | |
| TRDFCR_01 | PWM3 | STCLK | ADEG | ADTRG | OLS1 | OLS0 | CMD1 | CMD0 | |
| TRDOER1_01 | ED1 | EC1 | EB1 | EA1 | ED0 | EC0 | EB0 | EA0 | |
| TRDOER2_01 | PTO | — | — | — | — | — | — | — | |
| TRDOCR_01 | TOD1 | TOC1 | TOB1 | TOA1 | TOD0 | TOC0 | TOB0 | TOA0 | |
| TRDCR_2 | CCLR2 | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TPSC2 | TPSC1 | TPSC0 | Timer RD (Channel 2) |
| TRDIORA_2 | — | IOB2 | IOB1 | IOB0 | — | IOA2 | IOA1 | IOA0 | |
| TRDIORC_2 | IOD3 | IOD2 | IOD1 | IOD0 | IOC3 | IOC2 | IOC1 | IOC0 | |
| TRDSR_2 | — | — | — | OVF | IMFD | IMFC | IMFB | IMFA | |
| TRDIER_2 | — | — | — | OVIE | IMIED | IMIEC | IMIEB | IMIEA | |
| POCR_2 | — | — | — | — | — | POLD | POLC | POLB | |
| TRDDF_2 | DFCK1 | DFCK0 | — | — | DFD | DFC | DFB | DFA | |
| TRDCR_3 | CCLR2 | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TPSC2 | TPSC1 | TPSC0 | Timer RD (Channel 3) |
| TRDIORA_3 | — | IOB2 | IOB1 | IOB0 | — | IOA2 | IOA1 | IOA0 | |
| TRDIORC_3 | IOD3 | IOD2 | IOD1 | IOD0 | IOC3 | IOC2 | IOC1 | IOC0 | |
| TRDSR_3 | — | — | UDF | OVF | IMFD | IMFC | IMFB | IMFA | |
| TRDIER_3 | — | — | — | OVIE | IMIED | IMIEC | IMIEB | IMIEA | |
| POCR_3 | — | — | — | — | — | POLD | POLC | POLB | |
| TRDDF_3 | DFCK1 | DFCK0 | — | — | DFD | DFC | DFB | DFA | |
| TRDSTR_23 | — | — | — | — | CSTPN1 | CSTPN0 | STR1 | STR0 | Timer RD (Channel 2 and 3 common) |
| TRDMDR_23 | BFD1 | BFC1 | BFD0 | BFC0 | — | — | — | SYNC | |
| TRDPMR_23 | — | PWMD1 | PWMC1 | PWMB1 | — | PWMD0 | PWMC0 | PWMB0 | |

| Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module Name |
|---|---|---|---|---|---|---|---|---|---|
| TRDFCR_23 | PWM3 | STCLK | ADEG | ADTRG | OLS1 | OLS0 | CMD1 | CMD0 | Timer RD (Channel 2 and 3 common) |
| TRDOER1_23 | ED1 | EC1 | EB1 | EA1 | ED0 | EC0 | EB0 | EA0 | |
| TRDOER2_23 | PTO | — | — | — | — | — | — | — | |
| TRDOCR_23 | TOD1 | TOC1 | TOB1 | TOA1 | TOD0 | TOC0 | TOB0 | TOA0 | |
| TRCMR | CTS | — | BUFEB | BUFEA | PWM2 | PWMD | PWMC | PWMB | Timer RC |
| TRCCR1 | CCLR | CKS2 | CKS1 | CKS0 | TOD | TOC | TOB | TOA | |
| TRCIER | OVIE | — | — | — | IMIED | IMIEC | IMIEB | IMIEA | |
| TRCSR | OVF | — | — | — | IMFD | IMFC | IMFB | IMFA | |
| TRCIOR0 | — | IOB2 | IOB1 | IOB0 | — | IOA2 | IOA1 | IOA0 | |
| TRCIOR1 | — | IOD2 | IOD1 | IOD0 | — | IOC2 | IOC1 | IOC0 | |
| TRCCR2 | TCEG1 | TCEG0 | CSTP | — | — | — | — | — | |
| TRCDF | DFCK1 | DFCK0 | — | DFTRG | DFD | DFC | DFB | DFA | |
| TRCOER | PTO | — | — | — | ED | EC | EB | EA | |
| RSECDR | BSY | SC12 | SC11 | SC10 | SC03 | SC02 | SC01 | SC00 | RTC |
| RMINDR | BSY | MN12 | MN11 | MN10 | MN03 | MN02 | MN01 | MN00 | |
| RHRDR | BSY | — | HR11 | HR10 | HR03 | HR02 | HR01 | HR00 | |
| RWKDR | BSY | — | — | — | — | WK2 | WK1 | WK0 | |
| RTCCR1 | RUN | 12/24 | PM | RST | INT | — | — | — | |
| RTCCR2 | — | — | FOIE | WKIE | DYIE | HRIE | MNIE | SEIE | |
| RTCCSR | — | RCS6 | RCS5 | — | RCS3 | RCS2 | RCS1 | RCS0 | |
| LVDCR | — | — | — | — | LVDSEL | — | LVDDE | LVDUE | LVD (optional) |
| LVDSR | — | — | — | — | — | — | LVDDF | LVDUF | |
| CKCSR | PMRJ1 | PMRJ0 | — | OSCSEL | CKSWIE | CKSWIF | — | CKSTA | Clock pulse generator |
| RCCR | RCSTP | FSEL | VCLSEL | — | — | — | RCPSC1 | RCPSC0 | On-chip oscillator |
| RCTRMDPR | WRI | PRWE | LOCKDW | TRMDRWE | — | — | — | — | |
| RCTRMDR | TRMD7 | TRMD6 | TRMD5 | TRMD4 | TRMD3 | TRMD2 | TRMD1 | TRMD0 | |
| ICRA | ICRA7 | ICRA6 | ICRA5 | ICRA4 | ICRA3 | ICRA2 | ICRA1 | — | Interrupt |
| ICRB | — | ICRB6 | ICRB5 | ICRB4 | — | — | — | — | |
| ICRC | ICRC7 | — | — | ICRC4 | — | ICRC2 | ICRC1 | ICRC0 | |
| ICRD | ICRD7 | ICRD6 | ICRD5 | ICRD4 | ICRD3 | — | — | — | |

| Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module Name |
|---|---|---|---|---|---|---|---|---|---|
| SMR_2 | COM | CHR | PE | PM | STOP | MP | CKS1 | CKS0 | SCI3_2 |
| BRR_2 | BRR7 | BRR6 | BRR5 | BRR4 | BRR3 | BRR2 | BRR1 | BRR0 | |
| SCR3_2 | TIE | RIE | TE | RE | MPIE | TEIE | CKE1 | CKE0 | |
| TDR_2 | TDR7 | TRD6 | TDR5 | TDR4 | TRD3 | TRD2 | TRD1 | TRD0 | |
| SSR_2 | TDRE | RDRF | OER | FER | PER | TEND | MPBR | MPBT | |
| RDR_2 | RDR7 | RDR6 | RDR5 | RDR4 | RDR3 | RDR2 | RDR1 | RDR0 | |
| ICCR1 | ICE | RCVD | MST | TRS | CKS3 | CKS2 | CKS1 | CKS0 | IIC2 |
| ICCR2 | BBSY | SCP | SDAO | SDAOP | SCLO | — | IICRST | — | |
| ICMR | MLS | WAIT | — | — | BCWP | BC2 | BC1 | BC0 | |
| ICIER | TIE | TEIE | RIE | NAKIE | STIE | ACKE | ACKBR | ACKBT | |
| ICSR | TDRE | TEND | RDRF | NACKF | STOP | AL/OVE | AAS | ADZ | |
| SAR | SVA6 | SAV5 | SAV4 | SAV3 | SVA2 | SAV1 | SAV0 | FS | |
| ICDRT | ICDRT7 | ICDRT6 | ICDRT5 | ICDRT4 | ICDRT3 | ICDRT2 | ICDRT1 | ICDRT0 | |
| ICDRR | ICDRR7 | ICDRR6 | ICDRR5 | ICDRR4 | ICDRR3 | ICDRR2 | ICDRR1 | ICDRR0 | |
| TMB1 | TMB17 | — | — | — | — | TMB12 | TMB11 | TMB10 | Timer B1 |
| TCB1 | TCB17 | TCB16 | TCB15 | TCB14 | TCB13 | TCB12 | TCB11 | TCB10 | |
| TLB1 | TLB17 | TLB16 | TLB15 | TLB14 | TLB13 | TLB12 | TLB11 | TLB10 | |
| FLMCR1 | — | SWE | ESU | PSU | EV | PV | E | P | ROM |
| FLMCR2 | FLER | — | — | — | — | — | — | — | |
| FLPWCR | PDWND | — | — | — | — | — | — | — | |
| EBR1 | EB7 | EB6 | EB5 | EB4 | EB3 | EB2 | EB1 | EB0 | |
| FENR | FLSHE | — | — | — | — | — | — | — | |
| TCRV0 | CMIEB | CMIEA | OVIE | CCLR1 | CCLR0 | CKS2 | CKS1 | CKS0 | Timer V |
| TCSRV | CMFB | CMFA | OVF | — | OS3 | OS2 | OS1 | OS0 | |
| TCORA | TCORA7 | TCORA6 | TCORA5 | TCORA4 | TCORA3 | TCORA2 | TCORA1 | TCORA0 | |
| TCORB | TCORB7 | TCORB6 | TCORB5 | TCORB4 | TCORB3 | TCORB2 | TCORB1 | TCORB0 | |
| TCNTV | TCNTV7 | TCNTV6 | TCNTV5 | TCNTV4 | TCNTV3 | TCNTV2 | TCNTV1 | TCNTV0 | |
| TCRV1 | — | — | — | TVEG1 | TVEG0 | TRGE | — | ICKS0 | |
| SMR | COM | CHR | PE | PM | STOP | MP | CKS1 | CKS0 | SCI3 |
| BRR | BRR7 | BRR6 | BRR5 | BRR4 | BRR3 | BRR2 | BRR1 | BRR0 | |
| SCR3 | TIE | RIE | TE | RE | MPIE | TEIE | CKE1 | CKE0 | |

| Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module Name |
|---|---|---|---|---|---|---|---|---|---|
| TDR | TDR7 | TRD6 | TDR5 | TDR4 | TRD3 | TRD2 | TRD1 | TRD0 | SCI3 |
| SSR | TDRE | RDRF | OER | FER | PER | TEND | MPBR | MPBT | |
| RDR | RDR7 | RDR6 | RDR5 | RDR4 | RDR3 | RDR2 | RDR1 | RDR0 | |
| PWDRL | PWDRL7 | PWDRL6 | PWDRL5 | PWDRL4 | PWDRL3 | PWDRL2 | PWDRL1 | PWDRL0 | 14-bit PWM |
| PWDRU | — | — | PWDRU5 | PWDRU4 | PWDRU3 | PWDRU2 | PWDRU1 | PWDRU0 | |
| PWCR | — | — | — | — | — | — | — | PWCR0 | |
| TCSRWD | B6WI | TCWE | B4WI | TCSRWE | B2WI | WDON | BOWI | WRST | WDT* |
| TCWD | TCWD7 | TCWD6 | TCWD5 | TCWD4 | TCWD3 | TCWD2 | TCWD1 | TCWD0 | |
| TMWD | — | — | — | — | CKS3 | CKS2 | CKS1 | CKS0 | |
| ABRKCR | RTINTE | CSEL1 | CSEL0 | ACMP2 | ACMP1 | ACMP0 | DCMP1 | DCMP0 | Address break |
| ABRKSR | ABIF | ABIE | — | — | — | — | — | — | |
| BARH | BARH7 | BARH6 | BARH5 | BARH4 | BARH3 | BARH2 | BARH1 | BARH0 | |
| BARL | BARL7 | BARL6 | BARL5 | BARL4 | BARL3 | BARL2 | BARL1 | BARL0 | |
| BDRH | BDRH7 | BDRH6 | BDRH5 | BDRH4 | BDRH3 | BDRH2 | BDRH1 | BDRH0 | |
| BDRL | BDRL7 | BDRL6 | BDRL5 | BDRL4 | BDRL3 | BDRL2 | BDRL1 | BDRL0 | |
| BARE | BARE7 | BARE6 | BARE5 | BARE4 | BARE3 | BARE2 | BARE1 | BARE0 | |
| PUCR1 | PUCR17 | PUCR16 | PUCR15 | PUCR14 | — | PUCR12 | PUCR11 | PUCR10 | I/O port |
| PUCR5 | — | — | PUCR55 | PUCR54 | PUCR53 | PUCR52 | PUCR51 | PUCR50 | |
| PDR1 | P17 | P16 | P15 | P14 | — | P12 | P11 | P10 | |
| PDR2 | P27 | P26 | P25 | P24 | P23 | P22 | P21 | P20 | |
| PDR3 | P37 | P36 | P35 | P34 | P33 | P32 | P31 | P30 | |
| PDR5 | P57 | P56 | P55 | P54 | P53 | P52 | P51 | P50 | |
| PDR7 | P77 | P76 | P75 | P74 | — | P72 | P71 | P70 | |
| PDR8 | P87 | P86 | P85 | — | — | — | — | — | |
| PDRC | — | — | — | — | PC3 | PC2 | PC1 | PC0 | |
| PMR1 | IRQ3 | IRQ2 | IRQ1 | IRQ0 | TXD2 | PWM | TXD | TMOW | |
| PMR5 | POF57 | POF56 | WKP5 | WKP4 | WKP3 | WKP2 | WKP1 | WKP0 | |
| PMR3 | POF27 | POF26 | POF25 | POF24 | POF23 | — | — | — | |
| PCR1 | PCR17 | PCR16 | PCR15 | PCR14 | — | PCR12 | PCR11 | PCR10 | |
| PCR2 | PCR27 | PCR26 | PCR25 | PCR24 | PCR23 | PCR22 | PCR21 | PCR20 | |
| PCR3 | PCR37 | PCR36 | PCR35 | PCR34 | PCR33 | PCR32 | PCR31 | PCR30 | |

| Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module Name |
|---|---|---|---|---|---|---|---|---|---|
| PCR5 | PCR57 | PCR56 | PCR55 | PCR54 | PCR53 | PCR52 | PCR51 | PCR50 | I/O port |
| PCR7 | PCR77 | PCR76 | PCR75 | PCR74 | — | PCR72 | PCR71 | PCR70 | |
| PCR8 | PCR87 | PCR86 | PCR85 | — | — | — | — | — | |
| PCRC | — | — | — | — | PCRC3 | PCRC2 | PCRC1 | PCRC0 | |
| SYSCR3 | — | — | — | — | — | — | — | STS3 | Power-down modes |
| SYSCR1 | SSBY | STS2 | STS1 | STS0 | NESEL | — | — | — | |
| SYSCR2 | SMSEL | LSON | DTON | MA2 | MA1 | MA0 | SA1 | SA0 | |
| IEGR1 | NMIEG | — | — | — | IEG3 | IEG2 | IEG1 | IEG0 | Address break |
| IEGR2 | — | — | WPEG5 | WPEG4 | WPEG3 | WPEG2 | WPEG1 | WPEG0 | |
| IENR1 | IENDT | IENTA | IENWP | — | IEN3 | IEN2 | IEN1 | IEN0 | |
| IENR2 | — | — | IENTB1 | — | — | — | — | — | |
| IRR1 | IRRDT | IRRTA | — | — | IRRI3 | IRRI2 | IRRI1 | IRRI0 | |
| IRR2 | — | — | IRRTB1 | — | — | — | — | — | |
| IWPR | — | — | IWPF5 | IWPF4 | IWPF3 | IWPF2 | IWPF1 | IWPF0 | |
| MSTCR1 | — | MSTIIC | MSTS3 | — | MSTWD | — | MSTTV | MSTTA | Power-down modes |
| MSTCR2 | MSTS3_2 | — | — | MSTTB1 | — | — | — | MSTPWM | |

Note: * WDT: Watchdog timer

## 22.3 Register States in Each Operating Mode

| Register Name | Reset | Active | Sleep | Subactive | Subsleep | Standby | Module |
|---|---|---|---|---|---|---|---|
| TRDCNT_0 | Initialized | — | — | — | — | — | Timer RD |
| GRA_0 | Initialized | — | — | — | — | — | (Channel 0) |
| GRB_0 | Initialized | — | — | — | — | — | |
| GRC_0 | Initialized | — | — | — | — | — | |
| GRD_0 | Initialized | — | — | — | — | — | |
| TRDCNT_1 | Initialized | — | — | — | — | — | Timer RD |
| GRA_1 | Initialized | — | — | — | — | — | (Channel 1) |
| GRB_1 | Initialized | — | — | — | — | — | |
| GRC_1 | Initialized | — | — | — | — | — | |
| GRD_1 | Initialized | — | — | — | — | — | |
| TRDCNT_2 | Initialized | — | — | — | — | — | Timer RD |
| GRA_2 | Initialized | — | — | — | — | — | (Channel 2) |
| GRB_2 | Initialized | — | — | — | — | — | |
| GRC_2 | Initialized | — | — | — | — | — | |
| GRD_2 | Initialized | — | — | — | — | — | |
| TRDCNT_3 | Initialized | — | — | — | — | — | Timer RD |
| GRA_3 | Initialized | — | — | — | — | — | (Channel 3) |
| GRB_3 | Initialized | — | — | — | — | — | |
| GRC_3 | Initialized | — | — | — | — | — | |
| GRD_3 | Initialized | — | — | — | — | — | |
| TRCCNT | Initialized | — | — | — | — | — | Timer RC |
| GRA | Initialized | — | — | — | — | — | |
| GRB | Initialized | — | — | — | — | — | |
| GRC | Initialized | — | — | — | — | — | |
| GRD | Initialized | — | — | — | — | — | |

| Register Name | Reset | Active | Sleep | Subactive | Subsleep | Standby | Module |
|---|---|---|---|---|---|---|---|
| SMR_3 | Initialized | — | — | Initialized | Initialized | Initialized | SCI3_3 |
| BRR_3 | Initialized | — | — | Initialized | Initialized | Initialized | |
| SCR3_3 | Initialized | — | — | Initialized | Initialized | Initialized | |
| TDR_3 | Initialized | — | — | Initialized | Initialized | Initialized | |
| SSR_3 | Initialized | — | — | Initialized | Initialized | Initialized | |
| RDR_3 | Initialized | — | — | Initialized | Initialized | Initialized | |
| SMCR_3 | Initialized | — | — | Initialized | Initialized | Initialized | |
| ADDRA | Initialized | — | — | Initialized | Initialized | Initialized | AD converter |
| ADDRB | Initialized | — | — | Initialized | Initialized | Initialized | |
| ADDRC | Initialized | — | — | Initialized | Initialized | Initialized | |
| ADDRD | Initialized | — | — | Initialized | Initialized | Initialized | |
| ADCSR | Initialized | — | — | Initialized | Initialized | Initialized | |
| ADCR | Initialized | — | — | Initialized | Initialized | Initialized | |
| PDRD | Initialized | — | — | — | — | — | I/O port |
| PDRE | Initialized | — | — | — | — | — | |
| PDRF | Initialized | — | — | — | — | — | |
| PDRG | Initialized | — | — | — | — | — | |
| PDRH | Initialized | — | — | — | — | — | |
| PDRJ | Initialized | — | — | — | — | — | |
| PMRF | Initialized | — | — | — | — | — | |
| PMRG | Initialized | — | — | — | — | — | |
| PCRD | Initialized | — | — | — | — | — | |
| PCRE | Initialized | — | — | — | — | — | |
| PCRG | Initialized | — | — | — | — | — | |
| PCRH | Initialized | — | — | — | — | — | |
| PCRJ | Initialized | — | — | — | — | — | |
| MSTCR4 | Initialized | — | — | — | — | — | Power-down modes |

| Register Name | Reset | Active | Sleep | Subactive | Subsleep | Standby | Module |
|---|---|---|---|---|---|---|---|
| TRDCR_0 | Initialized | — | — | — | — | — | Timer RD |
| TRDIORA_0 | Initialized | — | — | — | — | — | (Channel 0) |
| TRDIORC_0 | Initialized | — | — | — | — | — | |
| TRDSR_0 | Initialized | — | — | — | — | — | |
| TRDIER_0 | Initialized | — | — | — | — | — | |
| POCR_0 | Initialized | — | — | — | — | — | |
| TRDDF_0 | Initialized | — | — | — | — | — | |
| TRDCR1 | Initialized | — | — | — | — | — | Timer RD |
| TRDIORA_1 | Initialized | — | — | — | — | — | (Channel 1) |
| TRDIORC_1 | Initialized | — | — | — | — | — | |
| TRDSR_1 | Initialized | — | — | — | — | — | |
| TRDIER_1 | Initialized | — | — | — | — | — | |
| POCR_1 | Initialized | — | — | — | — | — | |
| TRDDF_1 | Initialized | — | — | — | — | — | |
| TRDSTR_01 | Initialized | — | — | — | — | — | Timer RD (Channel 0 and 1 common) |
| TRDMDR_01 | Initialized | — | — | — | — | — | |
| TRDPMR_01 | Initialized | — | — | — | — | — | |
| TRDFCR_01 | Initialized | — | — | — | — | — | |
| TRDOER1_01 | Initialized | — | — | — | — | — | |
| TRDOER2_01 | Initialized | — | — | — | — | — | |
| TRDOCR_01 | Initialized | — | — | — | — | — | |
| TRDCR_2 | Initialized | — | — | — | — | — | Timer RD |
| TRDIORA_2 | Initialized | — | — | — | — | — | (Channel 2) |
| TRDIORC_2 | Initialized | — | — | — | — | — | |
| TRDSR_2 | Initialized | — | — | — | — | — | |
| TRDIER_2 | Initialized | — | — | — | — | — | |
| POCR_2 | Initialized | — | — | — | — | — | |
| TRDDF_2 | Initialized | — | — | — | — | — | |

| Register Name | Reset | Active | Sleep | Subactive | Subsleep | Standby | Module |
|---|---|---|---|---|---|---|---|
| TRDCR_3 | Initialized | — | — | — | — | — | Timer RD |
| TRDIORA_3 | Initialized | — | — | — | — | — | (Channel 3) |
| TRDIORC_3 | Initialized | — | — | — | — | — | |
| TRDSR_3 | Initialized | — | — | — | — | — | |
| TRDIER_3 | Initialized | — | — | — | — | — | |
| POCR_3 | Initialized | — | — | — | — | — | |
| TRDDF_3 | Initialized | — | — | — | — | — | |
| TRDSTR_23 | Initialized | — | — | — | — | — | Timer RD (Channel 2 and 3 common) |
| TRDMDR_23 | Initialized | — | — | — | — | — | |
| TRDPMR_23 | Initialized | — | — | — | — | — | |
| TRDFCR_23 | Initialized | — | — | — | — | — | |
| TRDOER1_23 | Initialized | — | — | — | — | — | |
| TRDOER2_23 | Initialized | — | — | — | — | — | |
| TRDOCR_23 | Initialized | — | — | — | — | — | |
| TRCMR | Initialized | — | — | — | — | — | Timer RC |
| TRCCR1 | Initialized | — | — | — | — | — | |
| TRCIER | Initialized | — | — | — | — | — | |
| TRCSR | Initialized | — | — | — | — | — | |
| TRCIOR0 | Initialized | — | — | — | — | — | |
| TRCIOR1 | Initialized | — | — | — | — | — | |
| TRCCR2 | Initialized | — | — | — | — | — | |
| TRCDF | Initialized | — | — | — | — | — | |
| TRCOER | Initialized | — | — | — | — | — | |
| RSECDR | Initialized | — | — | — | — | — | RTC |
| RMINDR | Initialized | — | — | — | — | — | |
| RHRDR | Initialized | — | — | — | — | — | |
| RWKDR | Initialized | — | — | — | — | — | |
| RTCCR1 | Initialized | — | — | — | — | — | |
| RTCCR2 | Initialized | — | — | — | — | — | |
| RTCCSR | Initialized | — | — | — | — | — | |

| Register Name | Reset | Active | Sleep | Subactive | Subsleep | Standby | Module |
|---|---|---|---|---|---|---|---|
| LVDCR | Initialized | — | — | — | — | — | LVD (optional) |
| LVDSR | Initialized | — | — | — | — | — | |
| CKCSR | Initialized | — | — | — | — | — | Clock pulse generator |
| RCCR | Initialized | — | — | — | — | — | On-chip oscillator |
| RCTRMDPR | Initialized | — | — | — | — | — | |
| RCTRMDR | Initialized | — | — | — | — | — | |
| ICRA | Initialized | — | — | — | — | — | Interrupt |
| ICRB | Initialized | — | — | — | — | — | |
| ICRC | Initialized | — | — | — | — | — | |
| ICRD | Initialized | — | — | — | — | — | |
| SMR_2 | Initialized | — | — | Initialized | Initialized | Initialized | SCI3_2 |
| BRR_2 | Initialized | — | — | Initialized | Initialized | Initialized | |
| SCR3_2 | Initialized | — | — | Initialized | Initialized | Initialized | |
| TDR_2 | Initialized | — | — | Initialized | Initialized | Initialized | |
| SSR_2 | Initialized | — | — | Initialized | Initialized | Initialized | |
| RDR_2 | Initialized | — | — | Initialized | Initialized | Initialized | |
| ICCR1 | Initialized | — | — | — | — | — | IIC2 |
| ICCR2 | Initialized | — | — | — | — | — | |
| ICMR | Initialized | — | — | — | — | — | |
| ICIER | Initialized | — | — | — | — | — | |
| ICSR | Initialized | — | — | — | — | — | |
| SAR | Initialized | — | — | — | — | — | |
| ICDRT | Initialized | — | — | — | — | — | |
| ICDRR | Initialized | — | — | — | — | — | |
| TMB1 | Initialized | — | — | — | — | — | Timer B1 |
| TCB1 | Initialized | — | — | — | — | — | |
| TLB1 | Initialized | — | — | — | — | — | |

| Register Name | Reset | Active | Sleep | Subactive | Subsleep | Standby | Module |
|---|---|---|---|---|---|---|---|
| FLMCR1 | Initialized | — | — | Initialized | Initialized | Initialized | ROM |
| FLMCR2 | Initialized | — | — | — | — | — | |
| FLPWCR | Initialized | — | — | — | — | — | |
| EBR1 | Initialized | — | — | Initialized | Initialized | Initialized | |
| FENR | Initialized | — | — | — | — | — | |
| TCRV0 | Initialized | — | — | Initialized | Initialized | Initialized | Timer V |
| TCSRV | Initialized | — | — | Initialized | Initialized | Initialized | |
| TCORA | Initialized | — | — | Initialized | Initialized | Initialized | |
| TCORB | Initialized | — | — | Initialized | Initialized | Initialized | |
| TCNTV | Initialized | — | — | Initialized | Initialized | Initialized | |
| TCRV1 | Initialized | — | — | Initialized | Initialized | Initialized | |
| SMR | Initialized | — | — | Initialized | Initialized | Initialized | SCI3 |
| BRR | Initialized | — | — | Initialized | Initialized | Initialized | |
| SCR3 | Initialized | — | — | Initialized | Initialized | Initialized | |
| TDR | Initialized | — | — | Initialized | Initialized | Initialized | |
| SSR | Initialized | — | — | Initialized | Initialized | Initialized | |
| RDR | Initialized | — | — | Initialized | Initialized | Initialized | |
| PWDRL | Initialized | — | — | — | — | — | 14-bit PWM |
| PWDRU | Initialized | — | — | — | — | — | |
| PWCR | Initialized | — | — | — | — | — | |
| TCSRWD | Initialized | — | — | — | — | — | WDT* |
| TCWD | Initialized | — | — | — | — | — | |
| TMWD | Initialized | — | — | — | — | — | |
| ABRKCR | Initialized | — | — | — | — | — | Address break |
| ABRKSR | Initialized | — | — | — | — | — | |
| BARH | Initialized | — | — | — | — | — | |
| BARL | Initialized | — | — | — | — | — | |
| BDRH | Initialized | — | — | — | — | — | |
| BDRL | Initialized | — | — | — | — | — | |
| BARE | Initialized | — | — | — | — | — | |

| Register Name | Reset | Active | Sleep | Subactive | Subsleep | Standby | Module |
|---|---|---|---|---|---|---|---|
| PUCR1 | Initialized | — | — | — | — | — | I/O port |
| PUCR5 | Initialized | — | — | — | — | — | |
| PDR1 | Initialized | — | — | — | — | — | |
| PDR2 | Initialized | — | — | — | — | — | |
| PDR3 | Initialized | — | — | — | — | — | |
| PDR5 | Initialized | — | — | — | — | — | |
| PDR7 | Initialized | — | — | — | — | — | |
| PDR8 | Initialized | — | — | — | — | — | |
| PDRC | Initialized | — | — | — | — | — | |
| PMR1 | Initialized | — | — | — | — | — | |
| PMR5 | Initialized | — | — | — | — | — | |
| PMR3 | Initialized | — | — | — | — | — | |
| PCR1 | Initialized | — | — | — | — | — | |
| PCR2 | Initialized | — | — | — | — | — | |
| PCR3 | Initialized | — | — | — | — | — | |
| PCR5 | Initialized | — | — | — | — | — | |
| PCR7 | Initialized | — | — | — | — | — | |
| PCR8 | Initialized | — | — | — | — | — | |
| PCRC | Initialized | — | — | — | — | — | |
| SYSCR3 | Initialized | — | — | — | — | — | Power-down modes |
| SYSCR1 | Initialized | — | — | — | — | — | |
| SYSCR2 | Initialized | — | — | — | — | — | |
| IEGR1 | Initialized | — | — | — | — | — | Interrupt |
| IEGR2 | Initialized | — | — | — | — | — | |
| IENR1 | Initialized | — | — | — | — | — | |
| IENR2 | Initialized | — | — | — | — | — | |
| IRR1 | Initialized | — | — | — | — | — | |
| IRR2 | Initialized | — | — | — | — | — | |
| IWPR | Initialized | — | — | — | — | — | |

| Register Name | Reset | Active | Sleep | Subactive | Subsleep | Standby | Module |
|---|---|---|---|---|---|---|---|
| MSTCR1 | Initialized | — | — | — | — | — | Power-down modes |
| MSTCR2 | Initialized | — | — | — | — | — | |

Notes: — is not initialized

* WDT: Watchdog timer

# Section 23 Electrical Characteristics

## 23.1 Absolute Maximum Ratings

**Table 23.1 Absolute Maximum Ratings**

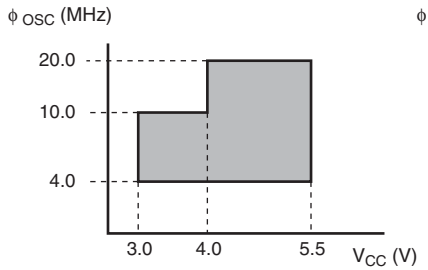
| Item | | Symbol | Value | Unit | Notes |
|---|---|---|---|---|---|
| Power supply voltage | | $V_{CC}$ | –0.3 to +7.0 | V | * |
| Analog power supply voltage | | $AV_{CC}$ | –0.3 to +7.0 | V | |
| Input voltage | Ports other than ports F, G, and X1 | $V_{IN}$ | –0.3 to $V_{CC}$ +0.3 | V | |
| | Ports F, G | | –0.3 to $AV_{CC}$ +0.3 | V | |
| | X1 | | –0.3 to 4.3 | V | |
| Operating temperature | | $T_{opr}$ | Regular specifications: –20 to +75 | °C | |
| | | | Wide-range specifications: –40 to +85 | °C | |
| Storage temperature | | $T_{stg}$ | –55 to +125 | °C | |

Note: * Permanent damage may result if maximum ratings are exceeded. Normal operation should be under the conditions specified in Electrical Characteristics. Exceeding these values can result in incorrect operation and reduced reliability.

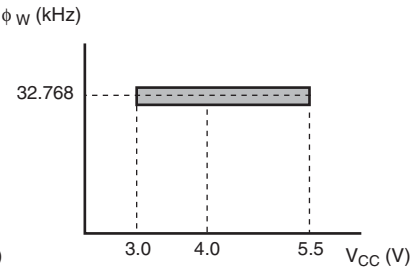
# 23.2 Electrical Characteristics

## 23.2.1 Power Supply Voltage and Operating Ranges

### (1) Power Supply Voltage and External Oscillation Frequency Range

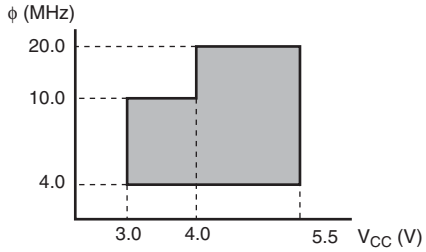



- $AV_{CC}$ = 3.0 to 5.5 V
- Active mode
- Sleep mode



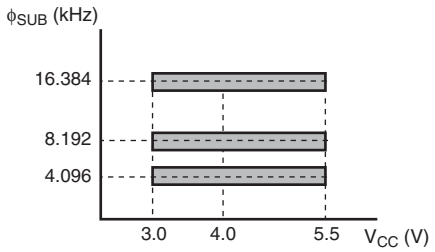

- $AV_{CC}$ = 3.0 to 5.5 V
- All operating modes

## (2) Power Supply Voltage and Operating Frequency Range

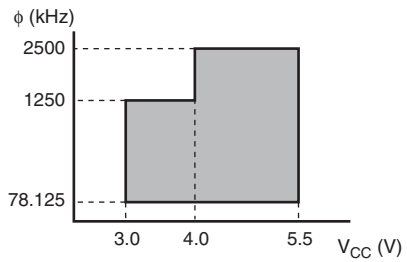



- $AV_{CC}$ = 3.0 to 5.5 V
- Active mode
- Sleep mode

(When MA2 in SYSCR2 = 0 )



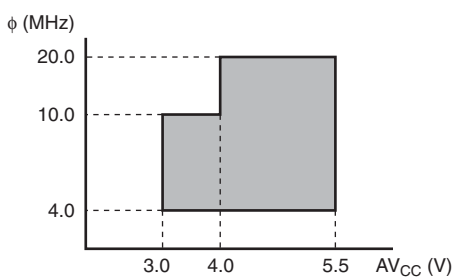

- $AV_{CC}$ = 3.0 to 5.5 V
- Subactive mode
- Subsleep mode




- $AV_{CC}$ = 3.0 to 5.5 V
- Active mode
- Sleep mode

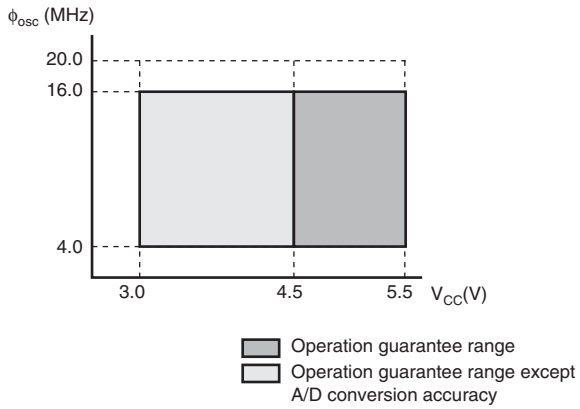
(When MA2 in SYSCR2 = 1 )

## (3) Analog Power Supply Voltage and A/D Converter Accuracy Guarantee Range




- $V_{CC}$ = 3.0 to 5.5 V
- Active mode
- Sleep mode

### (4) Range of Power Supply Voltage and Oscillation Frequency when Low-Voltage Detection Circuit is Used




## 23.2.2 DC Characteristics

### Table 23.2 DC Characteristics (1)

$V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0.0 V, $T_a$ = –20 to +75°C/–40 to +85°C, unless otherwise indicated.

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Input high voltage | $V_{IH}$ | $\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TMIB1, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, FTIOA2 to FTIOD2, FTIOA3 to FTIOD3, FTIOA to FTIOD, SCK3, SCK3_2, SCK3_3, TRGV, FTCI, TRGC, $\overline{TRCOI}$, $\overline{TRDOI_0}$, $\overline{TRDOI_1}$ | $V_{CC}$ = 4.0 to 5.5 V | $V_{CC}$ × 0.8 | — | $V_{CC}$ + 0.3 | V | |
| | | | | $V_{CC}$ × 0.9 | — | $V_{CC}$ + 0.3 | V | |

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Input high voltage | $V_{IH}$ | RXD, RXD_2, RXD_3, SCL, SDA, P10 to P12, P14 to P17, P20 to P27, P30 to P37, P50 to P57, P70 to P72, P74 to P77, P85 to P87, PC0 to PC3, PD0 to PD7, PE0 to PE7, PH0 to PH7, PJ0, PJ1 | $V_{CC}$ = 4.0 to 5.5 V | $V_{CC} \times 0.7$ | — | $V_{CC}$ + 0.3 | V | |
| | | | | $V_{CC} \times 0.8$ | — | $V_{CC}$ + 0.3 | V | |
| | | PF0 to PF7, PG0 to PG7 | $AV_{CC}$ = 4.0 to 5.5 V | $AV_{CC} \times 0.7$ | — | $AV_{CC}$ + 0.3 | V | |
| | | | $AV_{CC}$ = 3.0 to 5.5 V | $AV_{CC} \times 0.8$ | — | $AV_{CC}$ + 0.3 | V | |
| | | OSC1 | $V_{CC}$ = 4.0 to 5.5 V | $V_{CC}$ – 0.5 | — | $V_{CC}$ + 0.3 | V | |
| | | | | $V_{CC}$ – 0.3 | — | $V_{CC}$ + 0.3 | V | |

Note: Connect the TEST pin to Vss.

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Input low voltage | $V_{IL}$ | $\overline{RES}$, $\overline{NMI}$, $\overline{WKP0}$ to $\overline{WKP5}$, $\overline{IRQ0}$ to $\overline{IRQ3}$, $\overline{ADTRG}$, TMIB1, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, FTIOA2 to FTIOD2, FTIOA3 to FTIOD3, FTIOA to FTIOD, SCK3, SCK3_2, SCK3_3, TRGV, FTCI, TRGC, $\overline{TRCOI}$, $\overline{TRDOI_0}$, $\overline{TRODI_1}$ | $V_{cc}$ = 4.0 to 5.5 V | –0.3 | — | $V_{cc}$ × 0.2 | V | |
| | | | | –0.3 | — | $V_{cc}$ × 0.1 | V | |
| | | RXD, RXD_2, RXD_3, SCL, SDA, P10 to P12, P14 to P17, P20 to P27, P30 to P37, P50 to P57, P70 to P72, P74 to P77, P85 to P87, PC0 to PC3 PD0 to PD7 PE0 to PE7 PH0 to PH7 PJ0, PJ1 | $V_{cc}$ = 4.0 to 5.5 V | –0.3 | — | $V_{cc}$ × 0.3 | V | |
| | | | | –0.3 | — | $V_{cc}$ × 0.2 | V | |
| | | PF0 to PF7 PG0 to PG7 | $AV_{cc}$ = 4.0 to 5.5 V | –0.3 | — | $AV_{cc}$ × 0.3 | V | |
| | | | $AV_{cc}$ = 3.0 to 5.5 V | –0.3 | — | $AV_{cc}$ × 0.2 | V | |
| | | OSC1 | $V_{cc}$ = 4.0 to 5.5 V | –0.3 | — | 0.5 | V | |
| | | | | –0.3 | — | 0.3 | V | |

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Output high voltage | $V_{OH}$ | P10 to P12, P14 to P17, P20 to P27, P30 to P37, P50 to P55, P70 to P72, P74 to P77, P85 to P87, PC0 to PC3, PD0 to PD7, PE0 to PE7, PH0 to PH7, PJ0, PJ1 | $V_{CC}$ = 4.0 to 5.5 V<br>$-I_{OH}$ = 5.0 mA | $V_{CC}$ – 1.0 | — | — | V | |
| | | | $-I_{OH}$ = 0.1 mA | $V_{CC}$ – 0.5 | — | — | V | |
| | | PG0 to PG7 | $-I_{OH}$ = 0.1 mA | $AV_{CC}$ – 0.5 | — | — | V | |
| | | P56, P57 | 4.0 V ≤ $V_{CC}$ ≤ 5.5 V<br>$-I_{OH}$ = 0.1 mA | $V_{CC}$ – 2.5 | — | — | V | |
| | | | 3.0 V ≤ $V_{CC}$ < 4.0 V<br>$-I_{OH}$ = 0.1 mA | $V_{CC}$ – 2.2 | — | — | V | |
| Output low voltage | $V_{OL}$ | P10 to P12, P14 to P17, P20 to P27, P30 to P37, P50 to P57, P70 to P72, P74 to P77, P85 to P87, PC0 to PC3, PH0 to PH3, PJ0, PJ1 | $V_{CC}$ = 4.0 to 5.5 V<br>$I_{OL}$ = 1.6 mA | — | — | 0.6 | V | |
| | | | $I_{OL}$ = 0.2 mA | — | — | 0.4 | V | |
| | | PG0 to PG7 | $I_{OL}$ = 0.2 mA | — | — | 0.4 | V | |
| | | PD0 to PD7, PE0 to PE7, PH4 to PH7 | $V_{CC}$ = 4.0 to 5.5 V<br>$I_{OL}$ = 20.0 mA | — | — | 1.5 | V | |
| | | | $V_{CC}$ = 4.0 to 5.5 V<br>$I_{OL}$ = 10.0 mA | — | — | 1.0 | V | |
| | | | $V_{CC}$ = 4.0 to 5.5 V<br>$I_{OL}$ = 1.6 mA | — | — | 0.4 | V | |
| | | | $I_{OL}$ = 0.4 mA | — | — | 0.4 | V | |
| | | SCL, SDA | $V_{CC}$ = 4.0 to 5.5 V<br>$I_{OL}$ = 6.0 mA | — | — | 0.6 | V | |
| | | | $I_{OL}$ = 3.0 mA | — | — | 0.4 | V | |

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Values Typ. | Values Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Input/ output leakage current | $\lvert I_{IL} \rvert$ | OSC1, $\overline{\text{RES}}$, $\overline{\text{NMI}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{ADTRG}}$, TRGV, TMRIV, TMCIV, FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, FTIOA2 to FTIOD2, FTIOA3 to FTIOD3, FTIOA to FTIOD, RXD, SCK3, RXD_2, SCK3_2, RXD_3, SCK3_3, SCL, SDA, TMIB1, FTCI, TRGC, $\overline{\text{TRCOI}}$, $\overline{\text{TRDOI_0}}$, $\overline{\text{TRDOI_1}}$ | $V_{IN}$ = 0.5 V or higher ($V_{CC}$ – 0.5 V) | — | — | 1.0 | µA | |
| | | P10 to P12, P14 to P17, P20 to P27, P30 to P37, P50 to P57, P70 to P72, P74 to P77, P85 to P87, PC0 to PC3, PD0 to PD7, PE0 to PE7, PH0 to PH7, PJ0, PJ1 | $V_{IN}$ = 0.5 V or higher ($V_{CC}$ – 0.5 V) | — | — | 1.0 | µA | |
| | | PF0 to PF7, PG0 to PG7 | $V_{IN}$ = 0.5 V or higher ($AV_{CC}$ – 0.5 V) | — | — | 1.0 | µA | |
| Pull-up MOS current | $-I_p$ | P10 to P12, P14 to P17, P50 to P55 | $V_{CC}$ = 5.0 V, $V_{IN}$ = 0.0 V | 50.0 | — | 300.0 | µA | |
| | | | $V_{CC}$ = 3.0 V, $V_{IN}$ = 0.0 V | — | 60.0 | — | µA | Reference value |
| Input capacitance | $C_{in}$ | All input pins except power supply pins | f = 1 MHz, $V_{IN}$ = 0.0 V, $T_a$ = 25°C | — | — | 15.0 | pF | |

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Active mode supply current | $I_{OPE1}$ | $V_{CC}$ | Active mode 1 $V_{CC}$ = 5.0 V, $f_{OSC}$ = 20 MHz | — | 33.0 | 40.0 | mA | * |
| | | | Active mode 1 $V_{CC}$ = 3.0 V, $f_{OSC}$ = 10 MHz | — | 15.0 | — | | * Reference value |
| | $I_{OPE2}$ | $V_{CC}$ | Active mode 2 $V_{CC}$ = 5.0 V, $f_{OSC}$ = 20 MHz | — | 6.0 | 7.5 | mA | * |
| | | | Active mode 2 $V_{CC}$ = 3.0 V, $f_{OSC}$ = 10 MHz | — | 4.5 | — | | * Reference value |
| Sleep mode supply current | $I_{SLEEP1}$ | $V_{CC}$ | Sleep mode 1 $V_{CC}$ = 5.0 V, $f_{OSC}$ = 20 MHz | — | 22.0 | 30.0 | mA | * |
| | | | Sleep mode 1 $V_{CC}$ = 3.0 V, $f_{OSC}$ = 10 MHz | — | 12.0 | — | | * Reference value |
| | $I_{SLEEP2}$ | $V_{CC}$ | Sleep mode 2 $V_{CC}$ = 5.0 V, $f_{OSC}$ = 20 MHz | — | 5.0 | 6.5 | mA | * |
| | | | Sleep mode 2 $V_{CC}$ = 3.0 V, $f_{OSC}$ = 10 MHz | — | 4.5 | — | | * Reference value |
| Subactive mode supply current | $I_{SUB}$ | $V_{CC}$ | $V_{CC}$ = 3.0 V 32-kHz crystal resonator used ($\phi_{SUB} = \phi_W/2$) | — | 130 | 150 | µA | * Optional |
| | | | | — | 50 | 70 | | * |
| | | | $V_{CC}$ = 3.0 V 32-kHz crystal resonator not used ($\phi_{SUB} = \phi_W/8$) | — | 100 | — | | Reference value Optional |
| | | | | — | 40 | — | | * |
| Subsleep mode supply current | $I_{SUBSP1}$ | $V_{CC}$ | Subsleep mode 1 $V_{CC}$ = 3.0 V 32-kHz crystal resonator used ($\phi_{SUB} = \phi_W/2$) | — | 110 | 140 | µA | * Optional |
| | | | | — | 40 | 50 | | * |
| | $I_{SUBSP2}$ | $V_{CC}$ | Subsleep mode 2 $V_{CC}$ = 3.0 V 32-kHz crystal resonator not used | — | 110 | 135 | | * Optional |
| | | | | — | — | 6.0 | | * |

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Standby mode supply current | $I_{STBY}$ | $V_{CC}$ | 32-kHz crystal resonator not used | — | — | 135 | µA | * Optional |
| | | | | — | — | 5.0 | | * |
| RAM data retaining voltage | $V_{RAM}$ | $V_{CC}$ | | 2.0 | — | — | V | |

Note: * Pin states during supply current measurement are given below (excluding current in the pull-up MOS transistors and output buffers).

| Mode | $\overline{RES}$ Pin | Internal State | Other Pins | Oscillator Pins |
|---|---|---|---|---|
| Active mode 1 | $V_{CC}$ | Operates | $V_{CC}$ | Main clock: ceramic or crystal resonator<br>Subclock: Pin X1 = $V_{SS}$ |
| Active mode 2 | | Operates (ϕ/64) | | |
| Sleep mode 1 | $V_{CC}$ | Only timers operate | $V_{CC}$ | |
| Sleep mode 2 | | Only timers operate (ϕ/64) | | |
| Subactive mode | $V_{CC}$ | Operates | $V_{CC}$ | Main clock: ceramic or crystal resonator |
| Subsleep mode 1 | | Only timers operate | $V_{CC}$ | Subclock: crystal resonator<br>On-chip oscillator stop |
| Subsleep mode 2 | $V_{CC}$ | CPU and timers both stop | $V_{CC}$ | Main clock: ceramic or crystal resonator |
| Standby mode | | | | Subclock: Pin X1 = $V_{SS}$<br>On-chip oscillator stop |

## Table 23.2 DC Characteristics (2)

$V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0.0 V, $T_a$ = –20 to +75°C/–40 to +85°C, unless otherwise indicated.

| | | | | Values | | | |
|---|---|---|---|---|---|---|---|
| **Item** | **Symbol** | **Applicable Pins** | **Test Condition** | **Min.** | **Typ.** | **Max.** | **Unit** |
| Allowable output low current (per pin) | $I_{OL}$ | Output pins except ports D, E, G, PH4 to PH7, SCL, and SDA | $V_{CC}$ = 4.0 to 5.5 V | — | — | 2.0 | mA |
| | | Ports D, E, PH4 to PH7 | | — | — | 20.0 | |
| | | Output pins except ports D, E, G, PH4 to PH7, SCL, and SDA | | — | — | 0.5 | |
| | | Ports D, E, PH4 to PH7 | | — | — | 10.0 | |
| | | SCL, SDA | | — | — | 6.0 | |
| | | Port G | | — | — | 0.4 | |
| Allowable output low current (total) | $\sum I_{OL}$ | Output pins except ports D, E, PH4 to PH7, SCL, and SDA | $V_{CC}$ = 4.0 to 5.5 V | — | — | 40.0 | mA |
| | | Ports D, E, PH4 to PH7, SCL, and SDA | | — | — | 120.0 | |
| | | Output pins except ports D, E, G, PH4 to PH7 | | — | — | 20.0 | |
| | | Ports D, E, PH4 to PH7, SCL, and SDA | | — | — | 60.0 | |
| | | Port G | | — | — | 3.2 | |
| Allowable output high current (per pin) | $\lvert -I_{OH} \rvert$ | All output pins except P56, P57, and port G | $V_{CC}$ = 4.0 to 5.5 V | — | — | 5.0 | mA |
| | | | | — | — | 0.2 | |
| | | P56, P57 | $V_{CC}$ = 4.0 to 5.5 V | — | — | 0.4 | |
| | | | | — | — | 0.2 | |
| | | Port G | | — | — | 0.2 | |
| Allowable output high current (total) | $\lvert -\sum I_{OH} \rvert$ | All output pins | $V_{CC}$ = 4.0 to 5.5 V | — | — | 50.0 | mA |
| | | | | — | — | 8.0 | |
| | | Port G | | — | — | 1.6 | |

### 23.2.3 AC Characteristics

### Table 23.3 AC Characteristics

$V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0.0 V, $T_a$ = –20 to +75°C/–40 to +85°C, unless otherwise indicated.

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|---|
| System clock oscillation frequency | $f_{OSC}$ | OSC1, OSC2 | $V_{CC}$ = 4.0 to 5.5 V | 4.0 | — | 20.0 | MHz | |
| | | | | 4.0 | — | 10.0 | | |
| System clock (ϕ)cycle time | $t_{cyc}$ | | | 1 | — | 64 | $t_{OSC}$ | * |
| | | | | — | — | 12.8 | µs | |
| Subclock oscillation frequency | $f_W$ | X1, X2 | | — | 32.768 | — | kHz | |
| Watch clock ($\phi_W$) cycle time | $t_W$ | X1, X2 | | — | 30.5 | — | µs | |
| Subclock ($\phi_{SUB}$) cycle time | $t_{subcyc}$ | | | 2 | — | 8 | $t_W$ | * |
| Instruction cycle time | | | | 2 | — | — | $t_{cyc}$ $t_{subcyc}$ | |
| Oscillation stabilization time (crystal resonator) | $t_{rc}$ | OSC1, OSC2 | | — | — | 10.0 | ms | |
| Oscillation stabilization time (ceramic resonator) | $t_{rc}$ | OSC1, OSC2 | | — | — | 5.0 | ms | |
| Oscillation stabilization time | $t_{rcx}$ | X1, X2 | | — | — | 2.0 | s | |
| External clock high width | $t_{CPH}$ | OSC1 | $V_{CC}$ = 4.0 to 5.5 V | 20.0 | — | — | ns | Figure 23.1 |
| | | | | 40.0 | — | — | | |
| External clock low width | $t_{CPL}$ | OSC1 | $V_{CC}$ = 4.0 to 5.5 V | 20.0 | — | — | ns | |
| | | | | 40.0 | — | — | | |
| External clock rise time | $t_{CPr}$ | OSC1 | $V_{CC}$ = 4.0 to 5.5 V | — | — | 10.0 | ns | |
| | | | | — | — | 15.0 | | |
| External clock fall time | $t_{CPf}$ | OSC1 | $V_{CC}$ = 4.0 to 5.5 V | — | — | 10.0 | ns | |
| | | | | — | — | 15.0 | | |
| $\overline{RES}$ pin low width | $t_{REL}$ | $\overline{RES}$ | At power-on and in modes other than those below | $t_{rc}$ | — | — | ms | Figure 23.2 |
| | | | In active mode and sleep mode operation | 1500 | — | — | ns | |

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|---|
| $\overline{\text{NMI}}$ pin high width | $t_{IH}$ | $\overline{\text{NMI}}$ | | 2tcyc + 1500 ns<br>2tsubcyc + 1500 ns | — | — | ns | Figure 23.3 |
| $\overline{\text{NMI}}$ pin low width | $t_{IL}$ | $\overline{\text{NMI}}$ | | 2tcyc + 1500 ns<br>2tsubcyc + 1500 ns | — | — | ns | |
| Input pin high width | $t_{IH}$ | TMBI1, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$, | | 3 | — | — | $t_{cyc}$<br>$t_{subcyc}$ | Figure 23.3 |
| | | FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, FTIOA2 to FTIOD2, FTIOA3 to FTIOD3, FTIOA to FTIOD, FTCI, TRGC, $\overline{\text{TRCOI}}$, $\overline{\text{TRDOI_0}}$, $\overline{\text{TRDOI_1}}$ | | 3 | — | — | $t_{cyc}$<br>$t_{subcyc}$<br>$\phi$40M | |
| Input pin low width | $t_{IL}$ | TMBI1, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ3}}$, $\overline{\text{WKP0}}$ to $\overline{\text{WKP5}}$, TMCIV, TMRIV, TRGV, $\overline{\text{ADTRG}}$ | | 3 | — | — | $t_{cyc}$<br>$t_{subcyc}$ | |
| | | FTIOA0 to FTIOD0, FTIOA1 to FTIOD1, FTIOA2 to FTIOD2, FTIOA3 to FTIOD3, FTIOA to FTIOD, FTCI, TRGC, $\overline{\text{TRCOI}}$, $\overline{\text{TRDOI_0}}$, $\overline{\text{TRDOI_1}}$ | | 3 | — | — | $t_{cyc}$<br>$t_{subcyc}$<br>$\phi$40M | |

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|---|
| On-chip oscillator oscillation frequency | $f_{RC}$ | | $V_{cc}$ = 4.0 to 5.5V<br>Ta = 25°C<br>FSEL = 1<br>VCLSEL = 0 | 39.40 | 40.00 | 40.60 | MHz | |
| | | | Ta = 25°C<br>FSEL = 1<br>VCLSEL = 0 | 39.20 | 40.00 | 40.80 | MHz | |
| | | | $V_{cc}$ = 4.0 to 5.5V<br>Ta = -20°C to +75°C<br>FSEL = 1<br>VCLSEL = 0 | 38.80 | 40.00 | 41.20 | MHz | |
| | | | $V_{cc}$ = 4.0 to 5.5V<br>Ta = -40°C to +85°C<br>FSEL = 1<br>VCLSEL = 0 | 38.40 | 40.00 | 41.60 | MHz | |
| | | | Ta = -20°C to +75°C<br>FSEL = 1<br>VCLSEL = 0 | 38.40 | 40.00 | 41.60 | MHz | |
| | | | Ta = -40°C to +85°C<br>FSEL = 1<br>VCLSEL = 0 | 38.00 | 40.00 | 42.00 | MHz | |
| | | | $V_{cc}$ = 4.0 to 5.5V<br>Ta = 25°C<br>FSEL = 0<br>VCLSEL = 0 | 31.52 | 32.00 | 32.48 | MHz | |
| | | | Ta = 25°C<br>FSEL = 0<br>VCLSEL = 0 | 31.36 | 32.00 | 32.64 | MHz | |
| | | | $V_{cc}$ = 4.0 to 5.5V<br>Ta = -20°C to +75°C<br>FSEL = 0<br>VCLSEL = 0 | 31.04 | 32.00 | 32.96 | MHz | |
| | | | $V_{cc}$ = 4.0 to 5.5V<br>Ta = -40°C to +85°C<br>FSEL = 0<br>VCLSEL = 0 | 30.72 | 32.00 | 33.28 | MHz | |
| | | | Ta = -20°C to +75°C<br>FSEL = 0<br>VCLSEL = 0 | 30.72 | 32.00 | 33.28 | MHz | |

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|---|
| On-chip oscillator oscillation frequency | $f_{RC}$ | | Ta = -40°C to +85°C<br>FSEL = 0<br>VCLSEL = 0 | 30.40 | 32.00 | 33.60 | MHz | |

Note: * Determined by the MA2, MA1, MA0, SA1, and SA0 bits in the system control register 2 (SYSCR2).

### Table 23.4 $I^2C$ Bus Interface Timing

$V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0.0 V, $T_a$ = –20 to +75°C/–40 to +85°C, unless otherwise indicated.

| Item | Symbol | Test Condition | Values Min. | Typ. | Max. | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|
| SCL input cycle time | $t_{SCL}$ | | $12t_{cyc}$ + 600 | — | — | ns | Figure 23.4 |
| SCL input high width | $t_{SCLH}$ | | $3t_{cyc}$ + 300 | — | — | ns | |
| SCL input low width | $t_{SCLL}$ | | $5t_{cyc}$ + 300 | — | — | ns | |
| SCL and SDA input fall time | $t_{Sf}$ | | — | — | 300 | ns | |
| SCL and SDA input spike pulse removal time | $t_{SP}$ | | — | — | $1t_{cyc}$ | ns | |
| SDA input bus-free time | $t_{BUF}$ | | $5t_{cyc}$ | — | — | ns | |
| Start condition input hold time | $t_{STAH}$ | | $3t_{cyc}$ | — | — | ns | |
| Retransmission start condition input setup time | $t_{STAS}$ | | $3t_{cyc}$ | — | — | ns | |
| Setup time for stop condition input | $t_{STOS}$ | | $3t_{cyc}$ | — | — | ns | |
| Data-input setup time | $t_{SDAS}$ | | $1t_{cyc}$+20 | — | — | ns | |
| Data-input hold time | $t_{SDAH}$ | | 0 | — | — | ns | |
| Capacitive load of SCL and SDA | $c_b$ | | 0 | — | 400 | pF | |
| SCL and SDA output fall time | $t_{Sf}$ | $V_{CC}$ = 4.0 to 5.5 V | — | — | 250 | ns | |
| | | | — | — | 300 | | |

## Table 23.5 Serial Communication Interface (SCI) Timing

$V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0.0 V, $T_a$ = –20 to +75°C/–40 to +85°C, unless otherwise indicated.

| Item | | Symbol | Applicable Pins | Test Condition | Values | | | Unit | Reference Figure |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | Min. | Typ. | Max. | | |
| Input clock cycle | Asynchronous | $t_{Scyc}$ | SCK3 | | 4 | — | — | $t_{cyc}$ | Figure 23.5 |
| | Clocked synchronous | | | | 6 | — | — | | |
| Input clock pulse width | | $t_{SCKW}$ | SCK3 | | 0.4 | — | 0.6 | $t_{Scyc}$ | |
| Transmit data delay time (clocked synchronous) | | $t_{TXD}$ | TXD | $V_{CC}$ = 4.0 to 5.5 V | — | — | 1 | $t_{cyc}$ | Figure 23.6 |
| | | | | | — | — | 1 | | |
| Receive data setup time (clocked synchronous) | | $t_{RXS}$ | RXD | $V_{CC}$ = 4.0 to 5.5 V | 50.0 | — | — | ns | |
| | | | | | 100.0 | — | — | | |
| Receive data hold time (clocked synchronous) | | $t_{RXH}$ | RXD | $V_{CC}$ = 4.0 to 5.5 V | 50.0 | — | — | ns | |
| | | | | | 100.0 | — | — | | |

### 23.2.4 A/D Converter Characteristics

**Table 23.6 A/D Converter Characteristics**

$V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0.0 V, $T_a$ = –20 to +75°C/–40 to +85°C, unless otherwise indicated.

| Item | Symbol | Applicable Pins | Test Condition | Values | | | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| | | | | Min. | Typ. | Max. | | |
| Analog power supply voltage | $AV_{CC}$ | $AV_{CC}$ | | 3.0 | $V_{CC}$ | 5.5 | V | *1 |
| Analog input voltage | $AV_{IN}$ | AN0 to AN15 | | $V_{SS}$ – 0.3 | — | $AV_{CC}$ + 0.3 | V | |
| Analog power supply current | $AI_{OPE}$ | $AV_{CC}$ | $AV_{CC}$ = 5.0 V $f_{OSC}$ = 20 MHz | — | — | 2.0 | mA | |
| | $AI_{STOP1}$ | $AV_{CC}$ | | — | 50 | — | µA | *2 Reference value |
| | $AI_{STOP2}$ | $AV_{CC}$ | | — | — | 5.0 | µA | *3 |
| Analog input capacitance | $C_{AIN}$ | AN0 to AN15 | | — | — | 30.0 | pF | |
| Allowable signal source impedance | $R_{AIN}$ | AN0 to AN15 | | — | — | 5.0 | kΩ | |
| Resolution (data length) | | | | 10 | 10 | 10 | Bit | |
| Conversion time (single mode) | | AN0 to AN15 | $AV_{CC}$ = 3.0 to 5.5 V | 134 | — | — | $t_{cyc}$ | |
| Nonlinearity error | | | | — | — | ±7.5 | LSB | |
| Offset error | | | | — | — | ±7.5 | LSB | |
| Full-scale error | | | | — | — | ±7.5 | LSB | |
| Quantization error | | | | — | — | ±0.5 | LSB | |
| Absolute accuracy | | | | — | — | ±8.0 | LSB | |
| Conversion time (single mode) | | AN0 to AN15 | $AV_{CC}$ = 4.0 to 5.5 V | 70 | — | — | $t_{cyc}$ | |
| Nonlinearity error | | | | — | — | ±7.5 | LSB | |
| Offset error | | | | — | — | ±7.5 | LSB | |
| Full-scale error | | | | — | — | ±7.5 | LSB | |
| Quantization error | | | | — | — | ±0.5 | LSB | |
| Absolute accuracy | | | | — | — | ±8.0 | LSB | |

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Conversion time (single mode) | | AN0 to AN7 | $AV_{CC}$ = 4.0 to 5.5 V | 134 | — | — | $t_{cyc}$ | |
| Nonlinearity error | | | | — | — | ±3.5 | LSB | |
| Offset error | | | | — | — | ±3.5 | LSB | |
| Full-scale error | | | | — | — | ±3.5 | LSB | |
| Quantization error | | | | — | — | ±0.5 | LSB | |
| Absolute accuracy | | | | — | — | ±4.0 | LSB | |
| Conversion time (single mode) | | AN8 to AN15 | $AV_{CC}$ = 4.0 to 5.5 V | 134 | — | — | $t_{cyc}$ | |
| Nonlinearity error | | | | — | — | ±5.5 | LSB | |
| Offset error | | | | — | — | ±5.5 | LSB | |
| Full-scale error | | | | — | — | ±5.5 | LSB | |
| Quantization error | | | | — | — | ±0.5 | LSB | |
| Absolute accuracy | | | | — | — | ±6.0 | LSB | |

Notes: 1. Set $AV_{CC} = V_{CC}$ when the A/D converter is not used.
2. $AI_{STOP1}$ is the current in active and sleep modes while the A/D converter is idle.
3. $AI_{STOP2}$ is the current at reset and in standby, subactive, and subsleep modes while the A/D converter is idle.

### 23.2.5 Watchdog Timer Characteristics

**Table 23.7 Watchdog Timer Characteristics**

$V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0.0 V, $T_a$ = –20 to +75°C/–40 to +85°C, unless otherwise indicated.

| Item | Symbol | Applicable Pins | Test Condition | Values Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|---|---|
| Internal oscillator overflow time | $t_{OVF}$ | | | 0.2 | 0.4 | — | s | * |

Note: * Shows the time to count from 0 to 255, at which point an internal reset is generated, when the internal oscillator is selected.

### 23.2.6 Flash Memory Characteristics

**Table 23.8 Flash Memory Characteristics**

$V_{CC}$ = 3.0 to 5.5 V, $V_{SS}$ = 0.0 V, $T_a$ = –20 to +75°C/–40 to +85°C, unless otherwise indicated.

| Item | | Symbol | Test Condition | Values Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| Programming time (per 128 bytes)*1*2*4 | | $t_P$ | | — | 7 | 200 | ms |
| Erase time (per block) *1*3*6 | | $t_E$ | | — | 100 | 1200 | ms |
| Reprogramming count | | $N_{WEC}$ | | 1000 | 10000 | — | Times |
| Programming | Wait time after SWE bit setting*1 | x | | 1 | — | — | µs |
| | Wait time after PSU bit setting*1 | y | | 50 | — | — | µs |
| | Wait time after P bit setting*1*4 | z1 | 1 ≤ n ≤ 6 | 28 | 30 | 32 | µs |
| | | z2 | 7 ≤ n ≤ 1000 | 198 | 200 | 202 | µs |
| | | z3 | Additional-programming | 8 | 10 | 12 | µs |
| | Wait time after P bit clear*1 | α | | 5 | — | — | µs |
| | Wait time after PSU bit clear*1 | β | | 5 | — | — | µs |
| | Wait time after PV bit setting*1 | γ | | 4 | — | — | µs |
| | Wait time after dummy write*1 | ε | | 2 | — | — | µs |
| | Wait time after PV bit clear*1 | η | | 2 | — | — | µs |
| | Wait time after SWE bit clear*1 | θ | | 100 | — | — | µs |
| | Maximum programming count *1*4*5 | N | | — | — | 1000 | Times |

| Item | | Symbol | Test Condition | Values Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| Erasing | Wait time after SWE bit setting*[1] | x | | 1 | — | — | µs |
| | Wait time after ESU bit setting*[1] | y | | 100 | — | — | µs |
| | Wait time after E bit setting*[1]*[6] | z | | 10 | — | 100 | ms |
| | Wait time after E bit clear*[1] | α | | 10 | — | — | µs |
| | Wait time after ESU bit clear*[1] | β | | 10 | — | — | µs |
| | Wait time after EV bit setting*[1] | γ | | 20 | — | — | µs |
| | Wait time after dummy write*[1] | ε | | 2 | — | — | µs |
| | Wait time after EV bit clear*[1] | η | | 4 | — | — | µs |
| | Wait time after SWE bit clear*[1] | θ | | 100 | — | — | µs |
| | Maximum erase count *[1]*[6]*[7] | N | | — | — | 120 | Times |

Notes:
1. Make the time settings in accordance with the program/erase algorithms.
2. The programming time for 128 bytes. (Indicates the total time for which the P bit in the flash memory control register 1 (FLMCR1) is set. The program-verify time is not included.)
3. The time required to erase one block. (Indicates the time for which the E bit in the flash memory control register 1 (FLMCR1) is set. The erase-verify time is not included.)
4. Maximum programming time ($t_P$ (max.)) = wait time after P bit setting (z) × maximum programming count (N)
5. Set the maximum programming count (N) according to the actual set values of z1, z2, and z3, so that it does not exceed the maximum programming time ($t_P$ (max.)). The wait time after P bit setting (z1, z2) should be changed as follows according to the value of the programming count (n).

   Programming count (n)

   $1 \leq n \leq 6$ z1 = 30 µs

   $7 \leq n \leq 1000$ z2 = 200 µs
6. Maximum erase time ($t_E$ (max.)) = wait time after E bit setting (z) × maximum erase count (N)
7. Set the maximum erase count (N) according to the actual set value of (z), so that it does not exceed the maximum erase time ($t_E$ (max.)).

### 23.2.7 Power-Supply-Voltage Detection Circuit Characteristics (Optional)

**Table 23.9 Power-Supply-Voltage Detection Circuit Characteristics**

$V_{SS}$ = 0.0 V, $T_a$ = –20 to +75°C/–40 to +85°C, unless otherwise indicated.

| Item | Symbol | Test Condition | Values | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| Power-supply falling detection voltage | Vint (D) | LVDSEL = 0 | 3.5 | 3.7 | — | V |
| Power-supply rising detection voltage | Vint (U) | LVDSEL = 0 | — | 4.1 | 4.3 | V |
| Reset detection voltage 1*[1] | Vreset1 | LVDSEL = 0 | — | 2.3 | 2.6 | V |
| Reset detection voltage 2*[2] | Vreset2 | LVDSEL = 1 | 3.3 | 3.6 | 3.9 | V |
| Lower-limit voltage of LVDR operation | $V_{LVDRmin}$ | | 1.0 | — | — | V |

Notes: 1. This voltage should be used when the falling and rising voltage detection function is used.
2. Select the low-voltage reset 2 when only the low-voltage detection reset is used.

### 23.2.8 Power-On Reset Circuit Characteristics (Optional)

**Table 23.10 Power-On Reset Circuit Characteristics**

$V_{SS}$ = 0.0 V, $T_a$ = –20 to +75°C/–40 to +85°C, unless otherwise indicated.

| Item | Symbol | Test Condition | Values | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| Pull-up resistance of $\overline{RES}$ pin | $R_{RES}$ | | 100 | 150 | — | kΩ |
| Power-on reset start voltage* | $V_{por}$ | — | — | — | 100 | mV |

Note: * The power-supply voltage (Vcc) must fall below Vpor = 100 mV and then rise after charge of the $\overline{RES}$ pin is removed completely. In order to remove charge of the $\overline{RES}$ pin, it is recommended that the diode be placed in the Vcc side. If the power-supply voltage (Vcc) rises from the point over 100 mV, a power-on reset may not occur.

## 23.3 Operation Timing

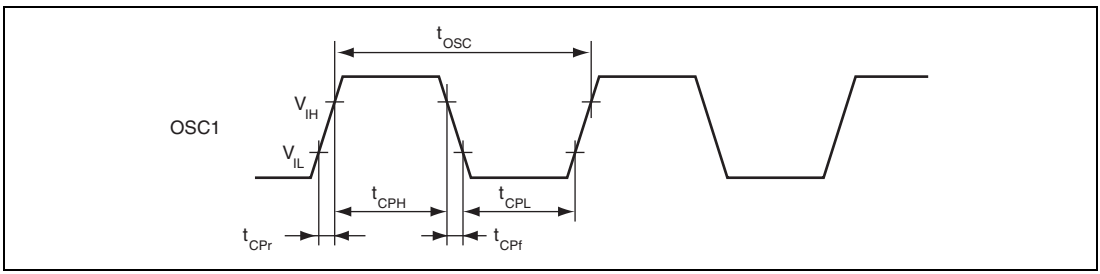



**Figure 23.1 System Clock Input Timing**

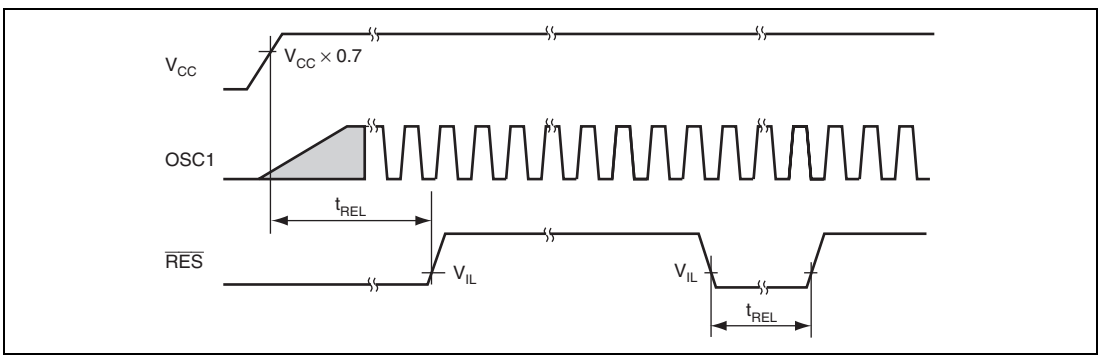



**Figure 23.2 $\overline{RES}$ Low Width Timing**

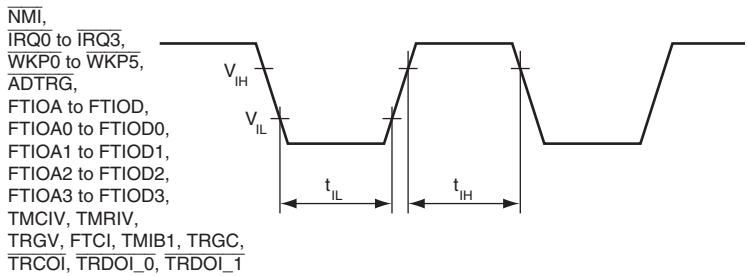
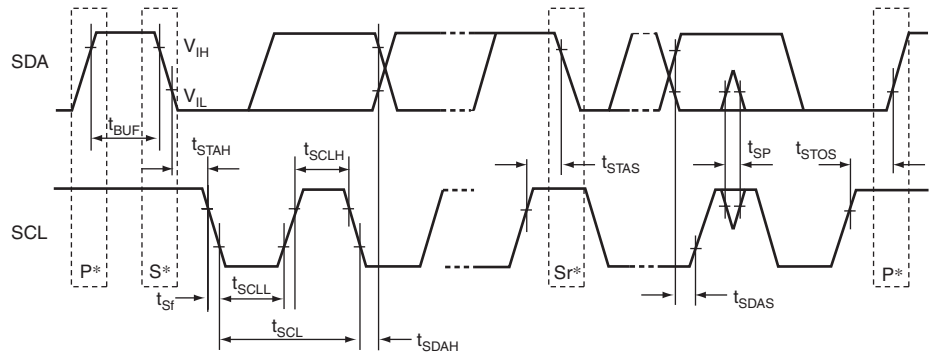



**Figure 23.3 Input Timing**




Note: * S, P, and Sr represent the following:
S: Start condition
P: Stop condition
Sr: Retransmission start condition

**Figure 23.4 I²C Bus Interface Input/Output Timing**

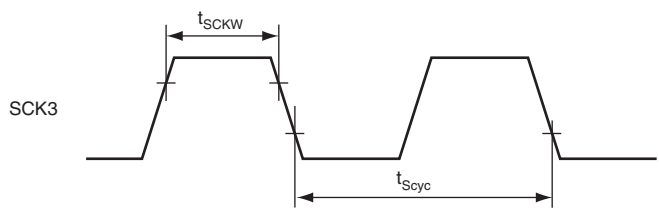
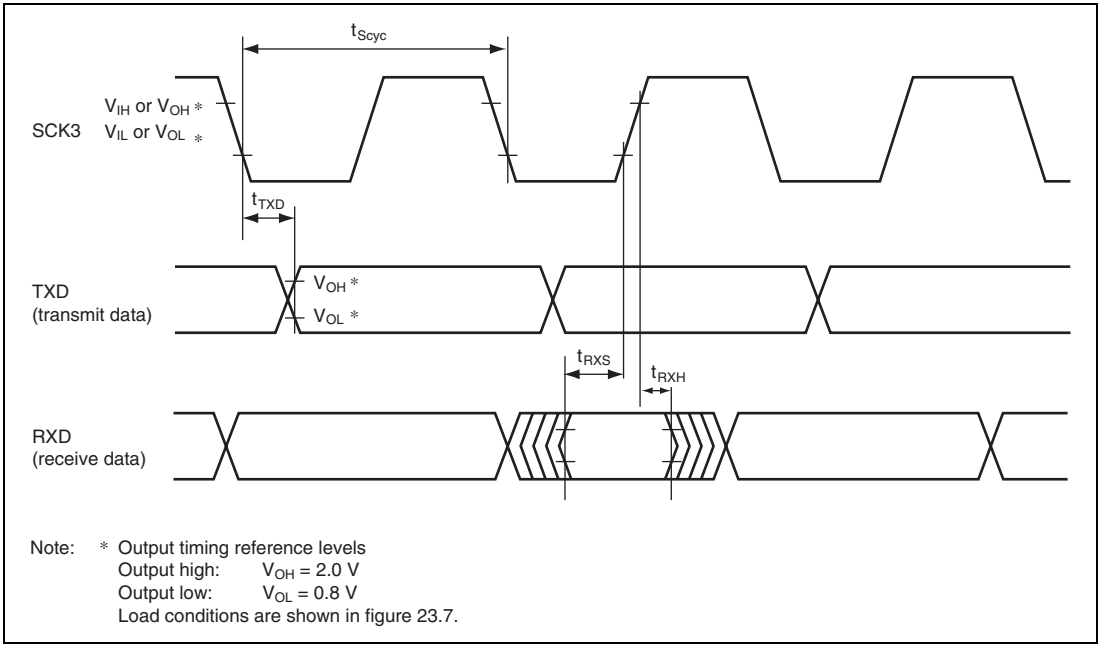



**Figure 23.5 SCK3 Input Clock Timing**




Note: * Output timing reference levels
Output high: $V_{OH}$ = 2.0 V
Output low: $V_{OL}$ = 0.8 V
Load conditions are shown in figure 23.7.

**Figure 23.6 SCI Input/Output Timing in Clocked Synchronous Mode**

## 23.4 Output Load Condition

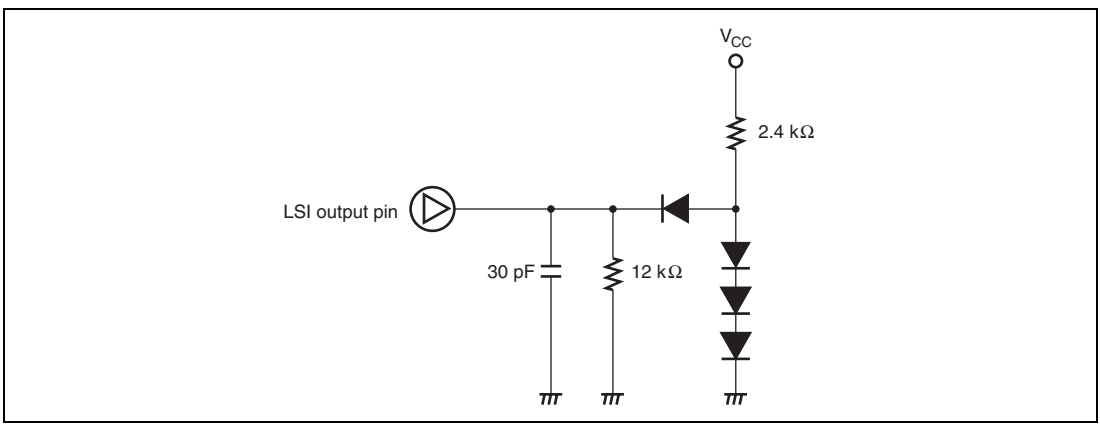



**Figure 23.7 Output Load Circuit**

# Appendix

## A. Instruction Set

### A.1 Instruction List

#### Condition Code

| Symbol | Description |
|---|---|
| Rd | General destination register |
| Rs | General source register |
| Rn | General register |
| ERd | General destination register (address register or 32-bit register) |
| ERs | General source register (address register or 32-bit register) |
| ERn | General register (32-bit register) |
| (EAd) | Destination operand |
| (EAs) | Source operand |
| PC | Program counter |
| SP | Stack pointer |
| CCR | Condition-code register |
| N | N (negative) flag in CCR |
| Z | Z (zero) flag in CCR |
| V | V (overflow) flag in CCR |
| C | C (carry) flag in CCR |
| disp | Displacement |
| → | Transfer from the operand on the left to the operand on the right, or transition from the state on the left to the state on the right |
| + | Addition of the operands on both sides |
| – | Subtraction of the operand on the right from the operand on the left |
| × | Multiplication of the operands on both sides |
| ÷ | Division of the operand on the left by the operand on the right |

| Symbol | Description |
|---|---|
| ∧ | Logical AND of the operands on both sides |
| ∨ | Logical OR of the operands on both sides |
| ⊕ | Logical exclusive OR of the operands on both sides |
| ¬ | NOT (logical complement) |
| ( ), < > | Contents of operand |
| ↕ | Changed according to execution result |
| * | Undetermined (no guaranteed value) |
| 0 | Cleared to 0 |
| 1 | Set to 1 |
| — | Not affected by execution of the instruction |
| Δ | Varies depending on conditions, described in notes |

Note: General registers include 8-bit registers (R0H to R7H and R0L to R7L) and 16-bit registers (R0 to R7 and E0 to E7).

## Table A.1 Instruction Set

### 1. Data Transfer Instructions

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | \| | | I | H | N | Z | V | C | Normal | Advanced |
| MOV | MOV.B #xx:8, Rd | B | 2 | | | | | | | | | #xx:8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | MOV.B Rs, Rd | B | | 2 | | | | | | | | Rs8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | MOV.B @ERs, Rd | B | | | 2 | | | | | | | @ERs → Rd8 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | MOV.B @(d:16, ERs), Rd | B | | | | 4 | | | | | | @(d:16, ERs) → Rd8 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.B @(d:24, ERs), Rd | B | | | | 8 | | | | | | @(d:24, ERs) → Rd8 | — | — | ↕ | ↕ | 0 | — | 10 | |
| | MOV.B @ERs+, Rd | B | | | | | 2 | | | | | @ERs → Rd8<br>ERs32+1 → ERs32 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.B @aa:8, Rd | B | | | | | | 2 | | | | @aa:8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | MOV.B @aa:16, Rd | B | | | | | | 4 | | | | @aa:16 → Rd8 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.B @aa:24, Rd | B | | | | | | 6 | | | | @aa:24 → Rd8 | — | — | ↕ | ↕ | 0 | — | 8 | |
| | MOV.B Rs, @ERd | B | | | 2 | | | | | | | Rs8 → @ERd | — | — | ↕ | ↕ | 0 | — | 4 | |
| | MOV.B Rs, @(d:16, ERd) | B | | | | 4 | | | | | | Rs8 → @(d:16, ERd) | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.B Rs, @(d:24, ERd) | B | | | | 8 | | | | | | Rs8 → @(d:24, ERd) | — | — | ↕ | ↕ | 0 | — | 10 | |
| | MOV.B Rs, @–ERd | B | | | | | 2 | | | | | ERd32–1 → ERd32<br>Rs8 → @ERd | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.B Rs, @aa:8 | B | | | | | | 2 | | | | Rs8 → @aa:8 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | MOV.B Rs, @aa:16 | B | | | | | | 4 | | | | Rs8 → @aa:16 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.B Rs, @aa:24 | B | | | | | | 6 | | | | Rs8 → @aa:24 | — | — | ↕ | ↕ | 0 | — | 8 | |
| | MOV.W #xx:16, Rd | W | 4 | | | | | | | | | #xx:16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | MOV.W Rs, Rd | W | | 2 | | | | | | | | Rs16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | MOV.W @ERs, Rd | W | | | 2 | | | | | | | @ERs → Rd16 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | MOV.W @(d:16, ERs), Rd | W | | | | 4 | | | | | | @(d:16, ERs) → Rd16 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.W @(d:24, ERs), Rd | W | | | | 8 | | | | | | @(d:24, ERs) → Rd16 | — | — | ↕ | ↕ | 0 | — | 10 | |
| | MOV.W @ERs+, Rd | W | | | | | 2 | | | | | @ERs → Rd16<br>ERs32+2 → @ERd32 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.W @aa:16, Rd | W | | | | | | 4 | | | | @aa:16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.W @aa:24, Rd | W | | | | | | 6 | | | | @aa:24 → Rd16 | — | — | ↕ | ↕ | 0 | — | 8 | |
| | MOV.W Rs, @ERd | W | | | 2 | | | | | | | Rs16 → @ERd | — | — | ↕ | ↕ | 0 | — | 4 | |
| | MOV.W Rs, @(d:16, ERd) | W | | | | 4 | | | | | | Rs16 → @(d:16, ERd) | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.W Rs, @(d:24, ERd) | W | | | | 8 | | | | | | Rs16 → @(d:24, ERd) | — | — | ↕ | ↕ | 0 | — | 10 | |

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States※1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | — | | I | H | N | Z | V | C | Normal | Advanced |
| MOV | MOV.W Rs, @–ERd | W | | | | | 2 | | | | | ERd32–2 → ERd32<br>Rs16 → @ERd | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.W Rs, @aa:16 | W | | | | | | 4 | | | | Rs16 → @aa:16 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.W Rs, @aa:24 | W | | | | | | 6 | | | | Rs16 → @aa:24 | — | — | ↕ | ↕ | 0 | — | 8 | |
| | MOV.L #xx:32, ERd | L | 6 | | | | | | | | | #xx:32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | MOV.L ERs, ERd | L | | 2 | | | | | | | | ERs32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | MOV.L @ERs, ERd | L | | | 4 | | | | | | | @ERs → ERd32 | — | — | ↕ | ↕ | 0 | — | 8 | |
| | MOV.L @(d:16, ERs), ERd | L | | | | 6 | | | | | | @(d:16, ERs) → ERd32 | — | — | ↕ | ↕ | 0 | — | 10 | |
| | MOV.L @(d:24, ERs), ERd | L | | | | 10 | | | | | | @(d:24, ERs) → ERd32 | — | — | ↕ | ↕ | 0 | — | 14 | |
| | MOV.L @ERs+, ERd | L | | | | | 4 | | | | | @ERs → ERd32<br>ERs32+4 → ERs32 | — | — | ↕ | ↕ | 0 | — | 10 | |
| | MOV.L @aa:16, ERd | L | | | | | | 6 | | | | @aa:16 → ERd32 | — | — | ↕ | ↕ | 0 | — | 10 | |
| | MOV.L @aa:24, ERd | L | | | | | | 8 | | | | @aa:24 → ERd32 | — | — | ↕ | ↕ | 0 | — | 12 | |
| | MOV.L ERs, @ERd | L | | | 4 | | | | | | | ERs32 → @ERd | — | — | ↕ | ↕ | 0 | — | 8 | |
| | MOV.L ERs, @(d:16, ERd) | L | | | | 6 | | | | | | ERs32 → @(d:16, ERd) | — | — | ↕ | ↕ | 0 | — | 10 | |
| | MOV.L ERs, @(d:24, ERd) | L | | | | 10 | | | | | | ERs32 → @(d:24, ERd) | — | — | ↕ | ↕ | 0 | — | 14 | |
| | MOV.L ERs, @–ERd | L | | | | | 4 | | | | | ERd32–4 → ERd32<br>ERs32 → @ERd | — | — | ↕ | ↕ | 0 | — | 10 | |
| | MOV.L ERs, @aa:16 | L | | | | | | 6 | | | | ERs32 → @aa:16 | — | — | ↕ | ↕ | 0 | — | 10 | |
| | MOV.L ERs, @aa:24 | L | | | | | | 8 | | | | ERs32 → @aa:24 | — | — | ↕ | ↕ | 0 | — | 12 | |
| POP | POP.W Rn | W | | | | | | | | | 2 | @SP → Rn16<br>SP+2 → SP | — | — | ↕ | ↕ | 0 | — | 6 | |
| | POP.L ERn | L | | | | | | | | | 4 | @SP → ERn32<br>SP+4 → SP | — | — | ↕ | ↕ | 0 | — | 10 | |
| PUSH | PUSH.W Rn | W | | | | | | | | | 2 | SP–2 → SP<br>Rn16 → @SP | — | — | ↕ | ↕ | 0 | — | 6 | |
| | PUSH.L ERn | L | | | | | | | | | 4 | SP–4 → SP<br>ERn32 → @SP | — | — | ↕ | ↕ | 0 | — | 10 | |
| MOVFPE | MOVFPE @aa:16, Rd | B | | | | | | 4 | | | | Cannot be used in this LSI | Cannot be used in this LSI | | | | | | | |
| MOVTPE | MOVTPE Rs, @aa:16 | B | | | | | | 4 | | | | Cannot be used in this LSI | Cannot be used in this LSI | | | | | | | |

## 2. Arithmetic Instructions

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | \| | | I | H | N | Z | V | C | Normal | Advanced |
| ADD | ADD.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8+#xx:8 → Rd8 | — | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| | ADD.B Rs, Rd | B | | 2 | | | | | | | | Rd8+Rs8 → Rd8 | — | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| | ADD.W #xx:16, Rd | W | 4 | | | | | | | | | Rd16+#xx:16 → Rd16 | — | (1) | ↕ | ↕ | ↕ | ↕ | 4 | |
| | ADD.W Rs, Rd | W | | 2 | | | | | | | | Rd16+Rs16 → Rd16 | — | (1) | ↕ | ↕ | ↕ | ↕ | 2 | |
| | ADD.L #xx:32, ERd | L | 6 | | | | | | | | | ERd32+#xx:32 → ERd32 | — | (2) | ↕ | ↕ | ↕ | ↕ | 6 | |
| | ADD.L ERs, ERd | L | | 2 | | | | | | | | ERd32+ERs32 → ERd32 | — | (2) | ↕ | ↕ | ↕ | ↕ | 2 | |
| ADDX | ADDX.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8+#xx:8 +C → Rd8 | — | ↕ | ↕ | (3) | ↕ | ↕ | 2 | |
| | ADDX.B Rs, Rd | B | | 2 | | | | | | | | Rd8+Rs8 +C → Rd8 | — | ↕ | ↕ | (3) | ↕ | ↕ | 2 | |
| ADDS | ADDS.L #1, ERd | L | | 2 | | | | | | | | ERd32+1 → ERd32 | — | — | — | — | — | — | 2 | |
| | ADDS.L #2, ERd | L | | 2 | | | | | | | | ERd32+2 → ERd32 | — | — | — | — | — | — | 2 | |
| | ADDS.L #4, ERd | L | | 2 | | | | | | | | ERd32+4 → ERd32 | — | — | — | — | — | — | 2 | |
| INC | INC.B Rd | B | | 2 | | | | | | | | Rd8+1 → Rd8 | — | — | ↕ | ↕ | ↕ | — | 2 | |
| | INC.W #1, Rd | W | | 2 | | | | | | | | Rd16+1 → Rd16 | — | — | ↕ | ↕ | ↕ | — | 2 | |
| | INC.W #2, Rd | W | | 2 | | | | | | | | Rd16+2 → Rd16 | — | — | ↕ | ↕ | ↕ | — | 2 | |
| | INC.L #1, ERd | L | | 2 | | | | | | | | ERd32+1 → ERd32 | — | — | ↕ | ↕ | ↕ | — | 2 | |
| | INC.L #2, ERd | L | | 2 | | | | | | | | ERd32+2 → ERd32 | — | — | ↕ | ↕ | ↕ | — | 2 | |
| DAA | DAA Rd | B | | 2 | | | | | | | | Rd8 decimal adjust → Rd8 | — | * | ↕ | ↕ | * | — | 2 | |
| SUB | SUB.B Rs, Rd | B | | 2 | | | | | | | | Rd8–Rs8 → Rd8 | — | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| | SUB.W #xx:16, Rd | W | 4 | | | | | | | | | Rd16–#xx:16 → Rd16 | — | (1) | ↕ | ↕ | ↕ | ↕ | 4 | |
| | SUB.W Rs, Rd | W | | 2 | | | | | | | | Rd16–Rs16 → Rd16 | — | (1) | ↕ | ↕ | ↕ | ↕ | 2 | |
| | SUB.L #xx:32, ERd | L | 6 | | | | | | | | | ERd32–#xx:32 → ERd32 | — | (2) | ↕ | ↕ | ↕ | ↕ | 6 | |
| | SUB.L ERs, ERd | L | | 2 | | | | | | | | ERd32–ERs32 → ERd32 | — | (2) | ↕ | ↕ | ↕ | ↕ | 2 | |
| SUBX | SUBX.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8–#xx:8–C → Rd8 | — | ↕ | ↕ | (3) | ↕ | ↕ | 2 | |
| | SUBX.B Rs, Rd | B | | 2 | | | | | | | | Rd8–Rs8–C → Rd8 | — | ↕ | ↕ | (3) | ↕ | ↕ | 2 | |
| SUBS | SUBS.L #1, ERd | L | | 2 | | | | | | | | ERd32–1 → ERd32 | — | — | — | — | — | — | 2 | |
| | SUBS.L #2, ERd | L | | 2 | | | | | | | | ERd32–2 → ERd32 | — | — | — | — | — | — | 2 | |
| | SUBS.L #4, ERd | L | | 2 | | | | | | | | ERd32–4 → ERd32 | — | — | — | — | — | — | 2 | |
| DEC | DEC.B Rd | B | | 2 | | | | | | | | Rd8–1 → Rd8 | — | — | ↕ | ↕ | ↕ | — | 2 | |
| | DEC.W #1, Rd | W | | 2 | | | | | | | | Rd16–1 → Rd16 | — | — | ↕ | ↕ | ↕ | — | 2 | |
| | DEC.W #2, Rd | W | | 2 | | | | | | | | Rd16–2 → Rd16 | — | — | ↕ | ↕ | ↕ | — | 2 | |

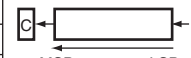
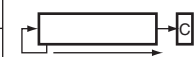
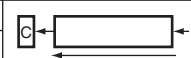
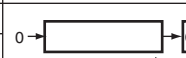
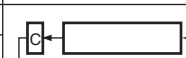
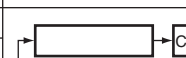
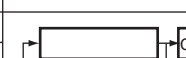
| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | \| | | I | H | N | Z | V | C | Normal | Advanced |
| DEC | DEC.L #1, ERd | L | | 2 | | | | | | | | ERd32–1 → ERd32 | — | — | ↕ | ↕ | ↕ | — | 2 | |
| | DEC.L #2, ERd | L | | 2 | | | | | | | | ERd32–2 → ERd32 | — | — | ↕ | ↕ | ↕ | — | 2 | |
| DAS | DAS.Rd | B | | 2 | | | | | | | | Rd8 decimal adjust → Rd8 | — | * | ↕ | ↕ | * | — | 2 | |
| MULXU | MULXU. B Rs, Rd | B | | 2 | | | | | | | | Rd8 × Rs8 → Rd16 (unsigned multiplication) | — | — | — | — | — | — | 14 | |
| | MULXU. W Rs, ERd | W | | 2 | | | | | | | | Rd16 × Rs16 → ERd32 (unsigned multiplication) | — | — | — | — | — | — | 22 | |
| MULXS | MULXS. B Rs, Rd | B | | 4 | | | | | | | | Rd8 × Rs8 → Rd16 (signed multiplication) | — | — | ↕ | ↕ | — | — | 16 | |
| | MULXS. W Rs, ERd | W | | 4 | | | | | | | | Rd16 × Rs16 → ERd32 (signed multiplication) | — | — | ↕ | ↕ | — | — | 24 | |
| DIVXU | DIVXU. B Rs, Rd | B | | 2 | | | | | | | | Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (unsigned division) | — | — | (6) | (7) | — | — | 14 | |
| | DIVXU. W Rs, ERd | W | | 2 | | | | | | | | ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division) | — | — | (6) | (7) | — | — | 22 | |
| DIVXS | DIVXS. B Rs, Rd | B | | 4 | | | | | | | | Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division) | — | — | (8) | (7) | — | — | 16 | |
| | DIVXS. W Rs, ERd | W | | 4 | | | | | | | | ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division) | — | — | (8) | (7) | — | — | 24 | |
| CMP | CMP.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8–#xx:8 | — | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| | CMP.B Rs, Rd | B | | 2 | | | | | | | | Rd8–Rs8 | — | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| | CMP.W #xx:16, Rd | W | 4 | | | | | | | | | Rd16–#xx:16 | — | (1) | ↕ | ↕ | ↕ | ↕ | 4 | |
| | CMP.W Rs, Rd | W | | 2 | | | | | | | | Rd16–Rs16 | — | (1) | ↕ | ↕ | ↕ | ↕ | 2 | |
| | CMP.L #xx:32, ERd | L | 6 | | | | | | | | | ERd32–#xx:32 | — | (2) | ↕ | ↕ | ↕ | ↕ | 4 | |
| | CMP.L ERs, ERd | L | | 2 | | | | | | | | ERd32–ERs32 | — | (2) | ↕ | ↕ | ↕ | ↕ | 2 | |

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | — | | I | H | N | Z | V | C | Normal | Advanced |
| NEG | NEG.B Rd | B | | 2 | | | | | | | | 0–Rd8 → Rd8 | — | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| | NEG.W Rd | W | | 2 | | | | | | | | 0–Rd16 → Rd16 | — | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| | NEG.L ERd | L | | 2 | | | | | | | | 0–ERd32 → ERd32 | — | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| EXTU | EXTU.W Rd | W | | 2 | | | | | | | | 0 → (<bits 15 to 8> of Rd16) | — | — | 0 | ↕ | 0 | — | 2 | |
| | EXTU.L ERd | L | | 2 | | | | | | | | 0 → (<bits 31 to 16> of ERd32) | — | — | 0 | ↕ | 0 | — | 2 | |
| EXTS | EXTS.W Rd | W | | 2 | | | | | | | | (<bit 7> of Rd16) → (<bits 15 to 8> of Rd16) | — | — | ↕ | ↕ | 0 | — | 2 | |
| | EXTS.L ERd | L | | 2 | | | | | | | | (<bit 15> of ERd32) → (<bits 31 to 16> of ERd32) | — | — | ↕ | ↕ | 0 | — | 2 | |

## 3. Logic Instructions

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | — | | I | H | N | Z | V | C | Normal | Advanced |
| AND | AND.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8∧#xx:8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | AND.B Rs, Rd | B | | 2 | | | | | | | | Rd8∧Rs8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | AND.W #xx:16, Rd | W | 4 | | | | | | | | | Rd16∧#xx:16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | AND.W Rs, Rd | W | | 2 | | | | | | | | Rd16∧Rs16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | AND.L #xx:32, ERd | L | 6 | | | | | | | | | ERd32∧#xx:32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | AND.L ERs, ERd | L | | 4 | | | | | | | | ERd32∧ERs32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 4 | |
| OR | OR.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8∨#xx:8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | OR.B Rs, Rd | B | | 2 | | | | | | | | Rd8∨Rs8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | OR.W #xx:16, Rd | W | 4 | | | | | | | | | Rd16∨#xx:16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | OR.W Rs, Rd | W | | 2 | | | | | | | | Rd16∨Rs16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | OR.L #xx:32, ERd | L | 6 | | | | | | | | | ERd32∨#xx:32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | OR.L ERs, ERd | L | | 4 | | | | | | | | ERd32∨ERs32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 4 | |
| XOR | XOR.B #xx:8, Rd | B | 2 | | | | | | | | | Rd8⊕#xx:8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | XOR.B Rs, Rd | B | | 2 | | | | | | | | Rd8⊕Rs8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | XOR.W #xx:16, Rd | W | 4 | | | | | | | | | Rd16⊕#xx:16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 4 | |
| | XOR.W Rs, Rd | W | | 2 | | | | | | | | Rd16⊕Rs16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | XOR.L #xx:32, ERd | L | 6 | | | | | | | | | ERd32⊕#xx:32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 6 | |
| | XOR.L ERs, ERd | L | | 4 | | | | | | | | ERd32⊕ERs32 → ERd32 | — | — | ↕ | ↕ | 0 | — | 4 | |
| NOT | NOT.B Rd | B | | 2 | | | | | | | | ¬ Rd8 → Rd8 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | NOT.W Rd | W | | 2 | | | | | | | | ¬ Rd16 → Rd16 | — | — | ↕ | ↕ | 0 | — | 2 | |
| | NOT.L ERd | L | | 2 | | | | | | | | ¬ Rd32 → Rd32 | — | — | ↕ | ↕ | 0 | — | 2 | |

# 4. Shift Instructions

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | — | | I | H | N | Z | V | C | Normal | Advanced |
| SHAL | SHAL.B Rd | B | | 2 | | | | | | | |  | — | — | ↕ | ↕ | ↕ | ↕ | 2 | |
| | SHAL.W Rd | W | | 2 | | | | | | | | | — | — | ↕ | ↕ | ↕ | ↕ | 2 | |
| | SHAL.L ERd | L | | 2 | | | | | | | | | — | — | ↕ | ↕ | ↕ | ↕ | 2 | |
| SHAR | SHAR.B Rd | B | | 2 | | | | | | | |  | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | SHAR.W Rd | W | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | SHAR.L ERd | L | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| SHLL | SHLL.B Rd | B | | 2 | | | | | | | |  | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | SHLL.W Rd | W | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | SHLL.L ERd | L | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| SHLR | SHLR.B Rd | B | | 2 | | | | | | | |  | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | SHLR.W Rd | W | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | SHLR.L ERd | L | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| ROTXL | ROTXL.B Rd | B | | 2 | | | | | | | |  | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | ROTXL.W Rd | W | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | ROTXL.L ERd | L | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| ROTXR | ROTXR.B Rd | B | | 2 | | | | | | | |  | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | ROTXR.W Rd | W | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | ROTXR.L ERd | L | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| ROTL | ROTL.B Rd | B | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | ROTL.W Rd | W | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | ROTL.L ERd | L | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| ROTR | ROTR.B Rd | B | | 2 | | | | | | | |  | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | ROTR.W Rd | W | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |
| | ROTR.L ERd | L | | 2 | | | | | | | | | — | — | ↕ | ↕ | 0 | ↕ | 2 | |

## 5. Bit-Manipulation Instructions

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | \| | | I | H | N | Z | V | C | Normal | Advanced |
| BSET | BSET #xx:3, Rd | B | | 2 | | | | | | | | (#xx:3 of Rd8) ← 1 | — | — | — | — | — | — | 2 | |
| | BSET #xx:3, @ERd | B | | | 4 | | | | | | | (#xx:3 of @ERd) ← 1 | — | — | — | — | — | — | 8 | |
| | BSET #xx:3, @aa:8 | B | | | | | | 4 | | | | (#xx:3 of @aa:8) ← 1 | — | — | — | — | — | — | 8 | |
| | BSET Rn, Rd | B | | 2 | | | | | | | | (Rn8 of Rd8) ← 1 | — | — | — | — | — | — | 2 | |
| | BSET Rn, @ERd | B | | | 4 | | | | | | | (Rn8 of @ERd) ← 1 | — | — | — | — | — | — | 8 | |
| | BSET Rn, @aa:8 | B | | | | | | 4 | | | | (Rn8 of @aa:8) ← 1 | — | — | — | — | — | — | 8 | |
| BCLR | BCLR #xx:3, Rd | B | | 2 | | | | | | | | (#xx:3 of Rd8) ← 0 | — | — | — | — | — | — | 2 | |
| | BCLR #xx:3, @ERd | B | | | 4 | | | | | | | (#xx:3 of @ERd) ← 0 | — | — | — | — | — | — | 8 | |
| | BCLR #xx:3, @aa:8 | B | | | | | | 4 | | | | (#xx:3 of @aa:8) ← 0 | — | — | — | — | — | — | 8 | |
| | BCLR Rn, Rd | B | | 2 | | | | | | | | (Rn8 of Rd8) ← 0 | — | — | — | — | — | — | 2 | |
| | BCLR Rn, @ERd | B | | | 4 | | | | | | | (Rn8 of @ERd) ← 0 | — | — | — | — | — | — | 8 | |
| | BCLR Rn, @aa:8 | B | | | | | | 4 | | | | (Rn8 of @aa:8) ← 0 | — | — | — | — | — | — | 8 | |
| BNOT | BNOT #xx:3, Rd | B | | 2 | | | | | | | | (#xx:3 of Rd8) ← ¬ (#xx:3 of Rd8) | — | — | — | — | — | — | 2 | |
| | BNOT #xx:3, @ERd | B | | | 4 | | | | | | | (#xx:3 of @ERd) ← ¬ (#xx:3 of @ERd) | — | — | — | — | — | — | 8 | |
| | BNOT #xx:3, @aa:8 | B | | | | | | 4 | | | | (#xx:3 of @aa:8) ← ¬ (#xx:3 of @aa:8) | — | — | — | — | — | — | 8 | |
| | BNOT Rn, Rd | B | | 2 | | | | | | | | (Rn8 of Rd8) ← ¬ (Rn8 of Rd8) | — | — | — | — | — | — | 2 | |
| | BNOT Rn, @ERd | B | | | 4 | | | | | | | (Rn8 of @ERd) ← ¬ (Rn8 of @ERd) | — | — | — | — | — | — | 8 | |
| | BNOT Rn, @aa:8 | B | | | | | | 4 | | | | (Rn8 of @aa:8) ← ¬ (Rn8 of @aa:8) | — | — | — | — | — | — | 8 | |
| BTST | BTST #xx:3, Rd | B | | 2 | | | | | | | | ¬ (#xx:3 of Rd8) → Z | — | — | — | ↕ | — | — | 2 | |
| | BTST #xx:3, @ERd | B | | | 4 | | | | | | | ¬ (#xx:3 of @ERd) → Z | — | — | — | ↕ | — | — | 6 | |
| | BTST #xx:3, @aa:8 | B | | | | | | 4 | | | | ¬ (#xx:3 of @aa:8) → Z | — | — | — | ↕ | — | — | 6 | |
| | BTST Rn, Rd | B | | 2 | | | | | | | | ¬ (Rn8 of @Rd8) → Z | — | — | — | ↕ | — | — | 2 | |
| | BTST Rn, @ERd | B | | | 4 | | | | | | | ¬ (Rn8 of @ERd) → Z | — | — | — | ↕ | — | — | 6 | |
| | BTST Rn, @aa:8 | B | | | | | | 4 | | | | ¬ (Rn8 of @aa:8) → Z | — | — | — | ↕ | — | — | 6 | |
| BLD | BLD #xx:3, Rd | B | | 2 | | | | | | | | (#xx:3 of Rd8) → C | — | — | — | — | — | ↕ | 2 | |

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | \| | | I | H | N | Z | V | C | Normal | Advanced |
| BLD | BLD #xx:3, @ERd | B | | | 4 | | | | | | | (#xx:3 of @ERd) → C | — | — | — | — | — | ↕ | 6 | |
| | BLD #xx:3, @aa:8 | B | | | | | | 4 | | | | (#xx:3 of @aa:8) → C | — | — | — | — | — | ↕ | 6 | |
| BILD | BILD #xx:3, Rd | B | | 2 | | | | | | | | ¬ (#xx:3 of Rd8) → C | — | — | — | — | — | ↕ | 2 | |
| | BILD #xx:3, @ERd | B | | | 4 | | | | | | | ¬ (#xx:3 of @ERd) → C | — | — | — | — | — | ↕ | 6 | |
| | BILD #xx:3, @aa:8 | B | | | | | | 4 | | | | ¬ (#xx:3 of @aa:8) → C | — | — | — | — | — | ↕ | 6 | |
| BST | BST #xx:3, Rd | B | | 2 | | | | | | | | C → (#xx:3 of Rd8) | — | — | — | — | — | — | 2 | |
| | BST #xx:3, @ERd | B | | | 4 | | | | | | | C → (#xx:3 of @ERd24) | — | — | — | — | — | — | 8 | |
| | BST #xx:3, @aa:8 | B | | | | | | 4 | | | | C → (#xx:3 of @aa:8) | — | — | — | — | — | — | 8 | |
| BIST | BIST #xx:3, Rd | B | | 2 | | | | | | | | ¬ C → (#xx:3 of Rd8) | — | — | — | — | — | — | 2 | |
| | BIST #xx:3, @ERd | B | | | 4 | | | | | | | ¬ C → (#xx:3 of @ERd24) | — | — | — | — | — | — | 8 | |
| | BIST #xx:3, @aa:8 | B | | | | | | 4 | | | | ¬ C → (#xx:3 of @aa:8) | — | — | — | — | — | — | 8 | |
| BAND | BAND #xx:3, Rd | B | | 2 | | | | | | | | C∧(#xx:3 of Rd8) → C | — | — | — | — | — | ↕ | 2 | |
| | BAND #xx:3, @ERd | B | | | 4 | | | | | | | C∧(#xx:3 of @ERd24) → C | — | — | — | — | — | ↕ | 6 | |
| | BAND #xx:3, @aa:8 | B | | | | | | 4 | | | | C∧(#xx:3 of @aa:8) → C | — | — | — | — | — | ↕ | 6 | |
| BIAND | BIAND #xx:3, Rd | B | | 2 | | | | | | | | C∧ ¬ (#xx:3 of Rd8) → C | — | — | — | — | — | ↕ | 2 | |
| | BIAND #xx:3, @ERd | B | | | 4 | | | | | | | C∧ ¬ (#xx:3 of @ERd24) → C | — | — | — | — | — | ↕ | 6 | |
| | BIAND #xx:3, @aa:8 | B | | | | | | 4 | | | | C∧ ¬ (#xx:3 of @aa:8) → C | — | — | — | — | — | ↕ | 6 | |
| BOR | BOR #xx:3, Rd | B | | 2 | | | | | | | | C∨(#xx:3 of Rd8) → C | — | — | — | — | — | ↕ | 2 | |
| | BOR #xx:3, @ERd | B | | | 4 | | | | | | | C∨(#xx:3 of @ERd24) → C | — | — | — | — | — | ↕ | 6 | |
| | BOR #xx:3, @aa:8 | B | | | | | | 4 | | | | C∨(#xx:3 of @aa:8) → C | — | — | — | — | — | ↕ | 6 | |
| BIOR | BIOR #xx:3, Rd | B | | 2 | | | | | | | | C∨ ¬ (#xx:3 of Rd8) → C | — | — | — | — | — | ↕ | 2 | |
| | BIOR #xx:3, @ERd | B | | | 4 | | | | | | | C∨ ¬ (#xx:3 of @ERd24) → C | — | — | — | — | — | ↕ | 6 | |
| | BIOR #xx:3, @aa:8 | B | | | | | | 4 | | | | C∨ ¬ (#xx:3 of @aa:8) → C | — | — | — | — | — | ↕ | 6 | |
| BXOR | BXOR #xx:3, Rd | B | | 2 | | | | | | | | C⊕(#xx:3 of Rd8) → C | — | — | — | — | — | ↕ | 2 | |
| | BXOR #xx:3, @ERd | B | | | 4 | | | | | | | C⊕(#xx:3 of @ERd24) → C | — | — | — | — | — | ↕ | 6 | |
| | BXOR #xx:3, @aa:8 | B | | | | | | 4 | | | | C⊕(#xx:3 of @aa:8) → C | — | — | — | — | — | ↕ | 6 | |
| BIXOR | BIXOR #xx:3, Rd | B | | 2 | | | | | | | | C⊕ ¬ (#xx:3 of Rd8) → C | — | — | — | — | — | ↕ | 2 | |
| | BIXOR #xx:3, @ERd | B | | | 4 | | | | | | | C⊕ ¬ (#xx:3 of @ERd24) → C | — | — | — | — | — | ↕ | 6 | |
| | BIXOR #xx:3, @aa:8 | B | | | | | | 4 | | | | C⊕ ¬ (#xx:3 of @aa:8) → C | — | — | — | — | — | ↕ | 6 | |

# 6. Branching Instructions

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | \| | Branch Condition | I | H | N | Z | V | C | Normal | Advanced |
| Bcc | BRA d:8 (BT d:8) | — | | | | | | | 2 | | | If condition is true then PC ← PC+d else next; Always | — | — | — | — | — | — | 4 | |
| | BRA d:16 (BT d:16) | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BRN d:8 (BF d:8) | — | | | | | | | 2 | | | Never | — | — | — | — | — | — | 4 | |
| | BRN d:16 (BF d:16) | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BHI d:8 | — | | | | | | | 2 | | | C∨ Z = 0 | — | — | — | — | — | — | 4 | |
| | BHI d:16 | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BLS d:8 | — | | | | | | | 2 | | | C∨ Z = 1 | — | — | — | — | — | — | 4 | |
| | BLS d:16 | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BCC d:8 (BHS d:8) | — | | | | | | | 2 | | | C = 0 | — | — | — | — | — | — | 4 | |
| | BCC d:16 (BHS d:16) | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BCS d:8 (BLO d:8) | — | | | | | | | 2 | | | C = 1 | — | — | — | — | — | — | 4 | |
| | BCS d:16 (BLO d:16) | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BNE d:8 | — | | | | | | | 2 | | | Z = 0 | — | — | — | — | — | — | 4 | |
| | BNE d:16 | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BEQ d:8 | — | | | | | | | 2 | | | Z = 1 | — | — | — | — | — | — | 4 | |
| | BEQ d:16 | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BVC d:8 | — | | | | | | | 2 | | | V = 0 | — | — | — | — | — | — | 4 | |
| | BVC d:16 | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BVS d:8 | — | | | | | | | 2 | | | V = 1 | — | — | — | — | — | — | 4 | |
| | BVS d:16 | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BPL d:8 | — | | | | | | | 2 | | | N = 0 | — | — | — | — | — | — | 4 | |
| | BPL d:16 | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BMI d:8 | — | | | | | | | 2 | | | N = 1 | — | — | — | — | — | — | 4 | |
| | BMI d:16 | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BGE d:8 | — | | | | | | | 2 | | | N⊕V = 0 | — | — | — | — | — | — | 4 | |
| | BGE d:16 | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BLT d:8 | — | | | | | | | 2 | | | N⊕V = 1 | — | — | — | — | — | — | 4 | |
| | BLT d:16 | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BGT d:8 | — | | | | | | | 2 | | | Z∨ (N⊕V) = 0 | — | — | — | — | — | — | 4 | |
| | BGT d:16 | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |
| | BLE d:8 | — | | | | | | | 2 | | | Z∨ (N⊕V) = 1 | — | — | — | — | — | — | 4 | |
| | BLE d:16 | — | | | | | | | 4 | | | | — | — | — | — | — | — | 6 | |

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | — | | I | H | N | Z | V | C | Normal | Advanced |
| JMP | JMP @ERn | — | | | 2 | | | | | | | PC ← ERn | — | — | — | — | — | — | 4 | |
| | JMP @aa:24 | — | | | | | | 4 | | | | PC ← aa:24 | — | — | — | — | — | — | 6 | |
| | JMP @@aa:8 | — | | | | | | | | 2 | | PC ← @aa:8 | — | — | — | — | — | — | 8 | 10 |
| BSR | BSR d:8 | — | | | | | | | 2 | | | PC → @–SP<br>PC ← PC+d:8 | — | — | — | — | — | — | 6 | 8 |
| | BSR d:16 | — | | | | | | | 4 | | | PC → @–SP<br>PC ← PC+d:16 | — | — | — | — | — | — | 8 | 10 |
| JSR | JSR @ERn | — | | | 2 | | | | | | | PC → @–SP<br>PC ← ERn | — | — | — | — | — | — | 6 | 8 |
| | JSR @aa:24 | — | | | | | | 4 | | | | PC → @–SP<br>PC ← aa:24 | — | — | — | — | — | — | 8 | 10 |
| | JSR @@aa:8 | — | | | | | | | | 2 | | PC → @–SP<br>PC ← @aa:8 | — | — | — | — | — | — | 8 | 12 |
| RTS | RTS | — | | | | | | | | | 2 | PC ← @SP+ | — | — | — | — | — | — | 8 | 10 |

# 7. System Control Instructions

| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States*1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | \| | | I | H | N | Z | V | C | Normal | Advanced |
| TRAPA | TRAPA #x:2 | — | | | | | | | | | 2 | PC → @–SP<br>CCR → @–SP<br><vector> → PC | 1 | — | — | — | — | — | 14 | 16 |
| RTE | RTE | — | | | | | | | | | | CCR ← @SP+<br>PC ← @SP+ | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | 10 | |
| SLEEP | SLEEP | — | | | | | | | | | | Transition to power-down state | — | — | — | — | — | — | 2 | |
| LDC | LDC #xx:8, CCR | B | 2 | | | | | | | | | #xx:8 → CCR | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| | LDC Rs, CCR | B | | 2 | | | | | | | | Rs8 → CCR | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| | LDC @ERs, CCR | W | | | 4 | | | | | | | @ERs → CCR | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | 6 | |
| | LDC @(d:16, ERs), CCR | W | | | | 6 | | | | | | @(d:16, ERs) → CCR | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | 8 | |
| | LDC @(d:24, ERs), CCR | W | | | | 10 | | | | | | @(d:24, ERs) → CCR | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | 12 | |
| | LDC @ERs+, CCR | W | | | | | 4 | | | | | @ERs → CCR<br>ERs32+2 → ERs32 | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | 8 | |
| | LDC @aa:16, CCR | W | | | | | | 6 | | | | @aa:16 → CCR | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | 8 | |
| | LDC @aa:24, CCR | W | | | | | | 8 | | | | @aa:24 → CCR | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | 10 | |
| STC | STC CCR, Rd | B | | 2 | | | | | | | | CCR → Rd8 | — | — | — | — | — | — | 2 | |
| | STC CCR, @ERd | W | | | 4 | | | | | | | CCR → @ERd | — | — | — | — | — | — | 6 | |
| | STC CCR, @(d:16, ERd) | W | | | | 6 | | | | | | CCR → @(d:16, ERd) | — | — | — | — | — | — | 8 | |
| | STC CCR, @(d:24, ERd) | W | | | | 10 | | | | | | CCR → @(d:24, ERd) | — | — | — | — | — | — | 12 | |
| | STC CCR, @–ERd | W | | | | | 4 | | | | | ERd32–2 → ERd32<br>CCR → @ERd | — | — | — | — | — | — | 8 | |
| | STC CCR, @aa:16 | W | | | | | | 6 | | | | CCR → @aa:16 | — | — | — | — | — | — | 8 | |
| | STC CCR, @aa:24 | W | | | | | | 8 | | | | CCR → @aa:24 | — | — | — | — | — | — | 10 | |
| ANDC | ANDC #xx:8, CCR | B | 2 | | | | | | | | | CCR∧#xx:8 → CCR | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| ORC | ORC #xx:8, CCR | B | 2 | | | | | | | | | CCR∨#xx:8 → CCR | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| XORC | XORC #xx:8, CCR | B | 2 | | | | | | | | | CCR⊕#xx:8 → CCR | ↕ | ↕ | ↕ | ↕ | ↕ | ↕ | 2 | |
| NOP | NOP | — | | | | | | | | | 2 | PC ← PC+2 | — | — | — | — | — | — | 2 | |

## 8. Block Transfer Instructions

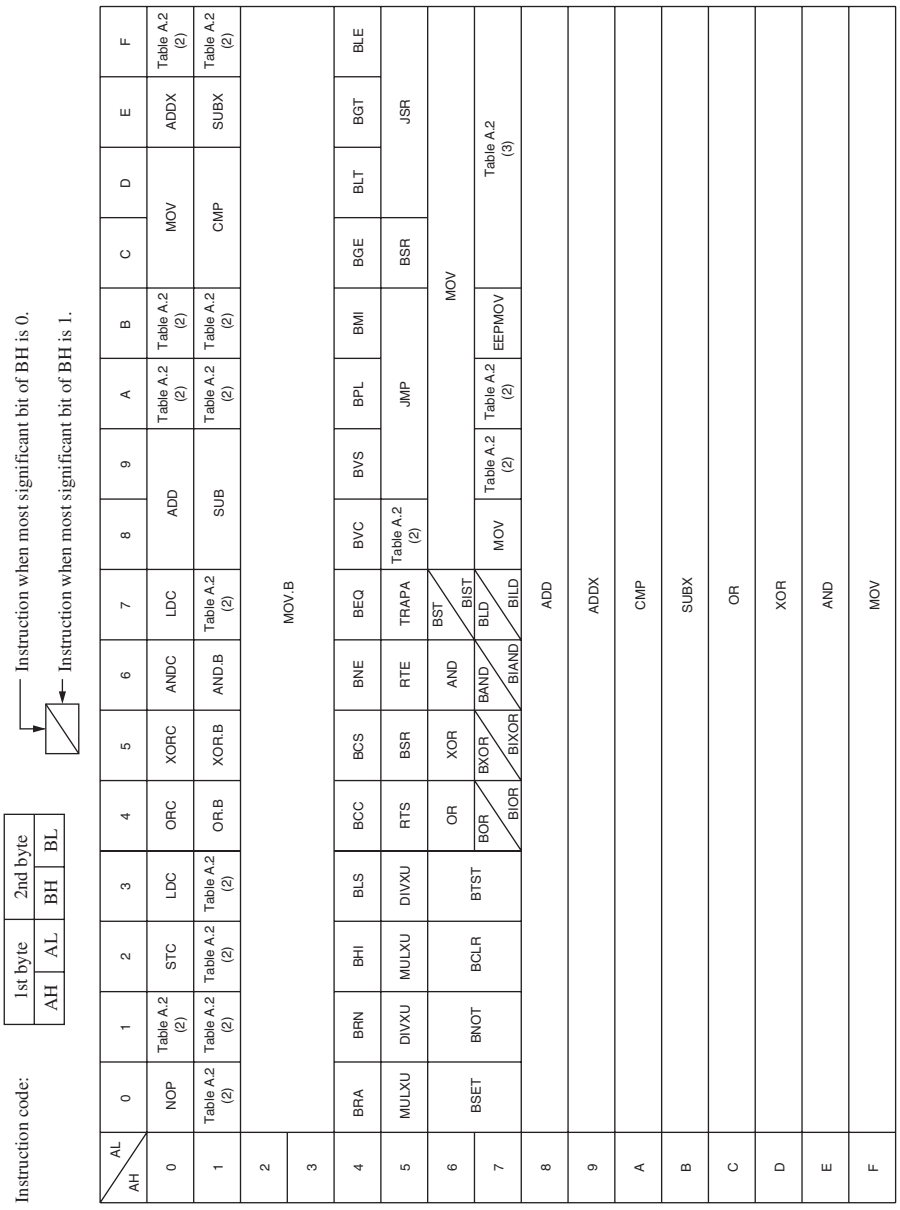
| Mnemonic | | Operand Size | Addressing Mode and Instruction Length (bytes) | | | | | | | | | Operation | Condition Code | | | | | | No. of States※1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | #xx | Rn | @ERn | @(d, ERn) | @–ERn/@ERn+ | @aa | @(d, PC) | @@aa | \| | | I | H | N | Z | V | C | Normal | Advanced |
| EEPMOV | EEPMOV. B | — | | | | | | | | | 4 | if R4L ≠ 0 then<br>repeat @R5 → @R6<br>R5+1 → R5<br>R6+1 → R6<br>R4L–1 → R4L<br>until R4L=0<br>else next | — | — | — | — | — | — | 8+<br>4n※2 | |
| | EEPMOV. W | — | | | | | | | | | 4 | if R4 ≠ 0 then<br>repeat @R5 → @R6<br>R5+1 → R5<br>R6+1 → R6<br>R4–1 → R4<br>until R4=0<br>else next | — | — | — | — | — | — | 8+<br>4n※2 | |

Notes: 1. The number of states in cases where the instruction code and its operands are located in on-chip memory is shown here. For other cases see appendix A.3, Number of Execution States.

2. The value n is set in register R4L or R4.
   (1) Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.
   (2) Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.
   (3) Retains its previous value when the result is zero; otherwise cleared to 0.
   (4) Set to 1 when the adjustment produces a carry; otherwise retains its previous value.
   (5) The number of states required for execution of an instruction that transfers data in synchronization with the E clock is variable.
   (6) Set to 1 when the divisor is negative; otherwise cleared to 0.
   (7) Set to 1 when the divisor is zero; otherwise cleared to 0.
   (8) Set to 1 when the quotient is negative; otherwise cleared to 0.

## A.2 Operation Code Map

### Table A.2 Operation Code Map (1)



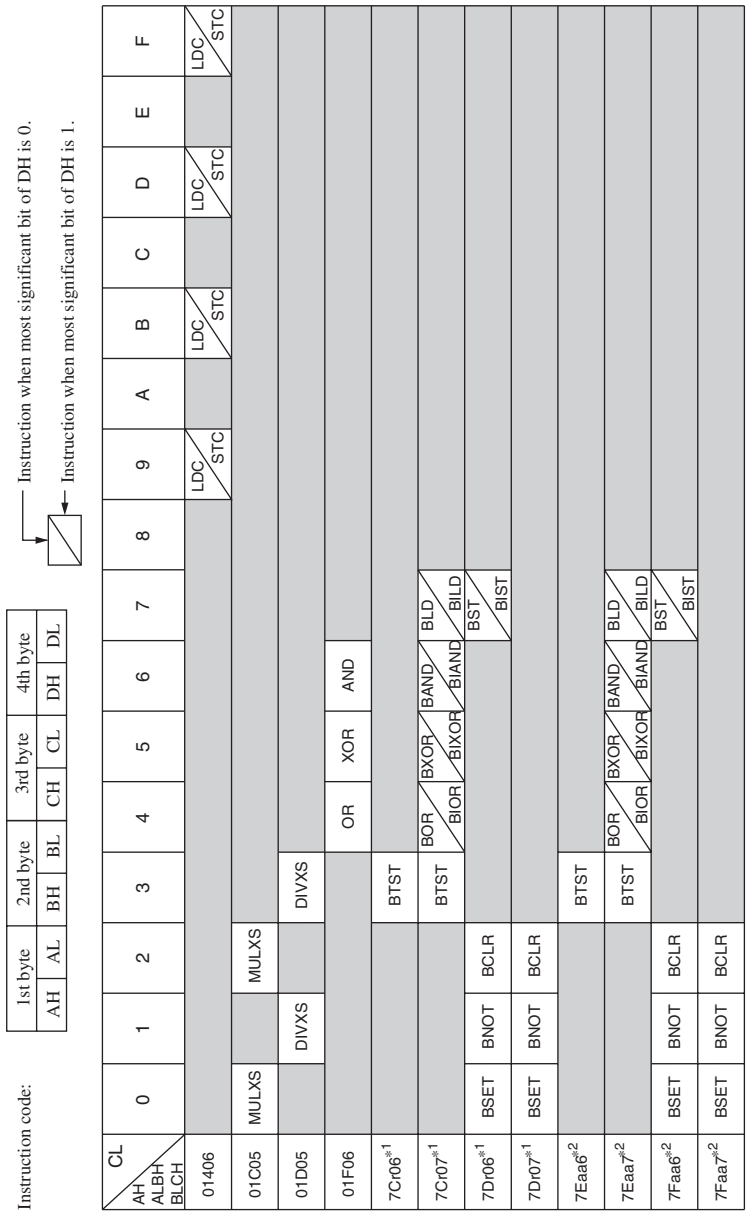
## Table A.2 Operation Code Map (2)

Instruction code:

| 1st byte | | 2nd byte | |
|---|---|---|---|
| AH | AL | BH | BL |

| BH / AH AL | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 01 | MOV | | | | LDC/STC | | | | SLEEP | | | | Table A.2 (3) | Table A.2 (3) | | Table A.2 (3) |
| 0A | INC | | | | | | | | ADD | | | | | | | |
| 0B | ADDS | | | | | INC | | INC | ADDS | | | | | INC | | INC |
| 0F | DAA | | | | | | | | MOV | | | | | | | |
| 10 | SHLL | SHLL | | SHLL | | | | | SHAL | SHAL | | SHAL | | | | |
| 11 | SHLR | SHLR | | SHLR | | | | | SHAR | SHAR | | SHAR | | | | |
| 12 | ROTXL | ROTXL | | ROTXL | | | | | ROTL | ROTL | | ROTL | | | | |
| 13 | ROTXR | ROTXR | | ROTXR | | | | | ROTR | ROTR | | ROTR | | | | |
| 17 | NOT | NOT | | NOT | | EXTU | | EXTU | NEG | NEG | | NEG | | EXTS | | EXTS |
| 1A | DEC | | | | | | | | SUB | | | | | | | |
| 1B | SUBS | | | | | DEC | | DEC | SUB | SUB | | | | DEC | | DEC |
| 1F | DAS | | | | | | | | CMP | | | | | | | |
| 58 | BRA | BRN | BHI | BLS | BCC | BCS | BNE | BEQ | BVC | BVS | BPL | BMI | BGE | BLT | BGT | BLE |
| 79 | MOV | ADD | CMP | SUB | OR | XOR | AND | | | | | | | | | |
| 7A | MOV | ADD | CMP | SUB | OR | XOR | AND | | | | | | | | | |

## Table A.2 Operation Code Map (3)




Notes: 1. r is the register designation field.
2. aa is the absolute address field.

## A.3 Number of Execution States

The status of execution for each instruction of the H8/300H CPU and the method of calculating the number of states required for instruction execution are shown below. Table A.4 shows the number of cycles of each type occurring in each instruction, such as instruction fetch and data read/write. Table A.3 shows the number of states required for each cycle. The total number of states required for execution of an instruction can be calculated by the following expression:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

**Examples:** When instruction is fetched from on-chip ROM, and an on-chip RAM is accessed.

BSET #0, @FF00

From table A.4:
I = L = 2, J = K = M = N= 0

From table A.3:
$S_I = 2$, $S_L = 2$

Number of states required for execution = $2 \times 2 + 2 \times 2 = 8$

When instruction is fetched from on-chip ROM, branch address is read from on-chip ROM, and on-chip RAM is used for stack area.

JSR @@ 30

From table A.4:
I = 2, J = K = 1, L = M = N = 0

From table A.3:
$S_I = S_J = S_K = 2$

Number of states required for execution = $2 \times 2 + 1 \times 2 + 1 \times 2 = 8$

**Table A.3 Number of Cycles in Each Instruction**

| Execution Status (Instruction Cycle) | | Access Location | |
|---|---|---|---|
| | | On-Chip Memory | On-Chip Peripheral Module |
| Instruction fetch | $S_I$ | 2 | — |
| Branch address read | $S_J$ | | |
| Stack operation | $S_K$ | | |
| Byte data access | $S_L$ | | 2 or 3* |
| Word data access | $S_M$ | | 2 or 3* |
| Internal operation | $S_N$ | 1 | |

Note: * Depends on which on-chip peripheral module is accessed. For details, see section 22.1, Register Addresses (Address Order).

## Table A.4 Number of Cycles in Each Instruction

| Instruction | Mnemonic | Instruction Fetch I | Branch Addr. Read J | Stack Operation K | Byte Data Access L | Word Data Access M | Internal Operation N |
|---|---|---|---|---|---|---|---|
| ADD | ADD.B #xx:8, Rd | 1 | | | | | |
| | ADD.B Rs, Rd | 1 | | | | | |
| | ADD.W #xx:16, Rd | 2 | | | | | |
| | ADD.W Rs, Rd | 1 | | | | | |
| | ADD.L #xx:32, ERd | 3 | | | | | |
| | ADD.L ERs, ERd | 1 | | | | | |
| ADDS | ADDS #1/2/4, ERd | 1 | | | | | |
| ADDX | ADDX #xx:8, Rd | 1 | | | | | |
| | ADDX Rs, Rd | 1 | | | | | |
| AND | AND.B #xx:8, Rd | 1 | | | | | |
| | AND.B Rs, Rd | 1 | | | | | |
| | AND.W #xx:16, Rd | 2 | | | | | |
| | AND.W Rs, Rd | 1 | | | | | |
| | AND.L #xx:32, ERd | 3 | | | | | |
| | AND.L ERs, ERd | 2 | | | | | |
| ANDC | ANDC #xx:8, CCR | 1 | | | | | |
| BAND | BAND #xx:3, Rd | 1 | | | | | |
| | BAND #xx:3, @ERd | 2 | | | 1 | | |
| | BAND #xx:3, @aa:8 | 2 | | | 1 | | |
| Bcc | BRA d:8 (BT d:8) | 2 | | | | | |
| | BRN d:8 (BF d:8) | 2 | | | | | |
| | BHI d:8 | 2 | | | | | |
| | BLS d:8 | 2 | | | | | |
| | BCC d:8 (BHS d:8) | 2 | | | | | |
| | BCS d:8 (BLO d:8) | 2 | | | | | |
| | BNE d:8 | 2 | | | | | |
| | BEQ d:8 | 2 | | | | | |
| | BVC d:8 | 2 | | | | | |
| | BVS d:8 | 2 | | | | | |
| | BPL d:8 | 2 | | | | | |
| | BMI d:8 | 2 | | | | | |
| | BGE d:8 | 2 | | | | | |

| Instruction | Mnemonic | Instruction Fetch I | Branch Addr. Read J | Stack Operation K | Byte Data Access L | Word Data Access M | Internal Operation N |
|---|---|---|---|---|---|---|---|
| Bcc | BLT d:8 | 2 | | | | | |
| | BGT d:8 | 2 | | | | | |
| | BLE d:8 | 2 | | | | | |
| | BRA d:16(BT d:16) | 2 | | | | | 2 |
| | BRN d:16(BF d:16) | 2 | | | | | 2 |
| | BHI d:16 | 2 | | | | | 2 |
| | BLS d:16 | 2 | | | | | 2 |
| | BCC d:16(BHS d:16) | 2 | | | | | 2 |
| | BCS d:16(BLO d:16) | 2 | | | | | 2 |
| | BNE d:16 | 2 | | | | | 2 |
| | BEQ d:16 | 2 | | | | | 2 |
| | BVC d:16 | 2 | | | | | 2 |
| | BVS d:16 | 2 | | | | | 2 |
| | BPL d:16 | 2 | | | | | 2 |
| | BMI d:16 | 2 | | | | | 2 |
| | BGE d:16 | 2 | | | | | 2 |
| | BLT d:16 | 2 | | | | | 2 |
| | BGT d:16 | 2 | | | | | 2 |
| | BLE d:16 | 2 | | | | | 2 |
| BCLR | BCLR #xx:3, Rd | 1 | | | | | |
| | BCLR #xx:3, @ERd | 2 | | | 2 | | |
| | BCLR #xx:3, @aa:8 | 2 | | | 2 | | |
| | BCLR Rn, Rd | 1 | | | | | |
| | BCLR Rn, @ERd | 2 | | | 2 | | |
| | BCLR Rn, @aa:8 | 2 | | | 2 | | |
| BIAND | BIAND #xx:3, Rd | 1 | | | | | |
| | BIAND #xx:3, @ERd | 2 | | | 1 | | |
| | BIAND #xx:3, @aa:8 | 2 | | | 1 | | |
| BILD | BILD #xx:3, Rd | 1 | | | | | |
| | BILD #xx:3, @ERd | 2 | | | 1 | | |
| | BILD #xx:3, @aa:8 | 2 | | | 1 | | |

| Instruction | Mnemonic | Instruction Fetch I | Branch Addr. Read J | Stack Operation K | Byte Data Access L | Word Data Access M | Internal Operation N |
|---|---|---|---|---|---|---|---|
| BIOR | BIOR #xx:8, Rd | 1 | | | | | |
| | BIOR #xx:8, @ERd | 2 | | | 1 | | |
| | BIOR #xx:8, @aa:8 | 2 | | | 1 | | |
| BIST | BIST #xx:3, Rd | 1 | | | | | |
| | BIST #xx:3, @ERd | 2 | | | 2 | | |
| | BIST #xx:3, @aa:8 | 2 | | | 2 | | |
| BIXOR | BIXOR #xx:3, Rd | 1 | | | | | |
| | BIXOR #xx:3, @ERd | 2 | | | 1 | | |
| | BIXOR #xx:3, @aa:8 | 2 | | | 1 | | |
| BLD | BLD #xx:3, Rd | 1 | | | | | |
| | BLD #xx:3, @ERd | 2 | | | 1 | | |
| | BLD #xx:3, @aa:8 | 2 | | | 1 | | |
| BNOT | BNOT #xx:3, Rd | 1 | | | | | |
| | BNOT #xx:3, @ERd | 2 | | | 2 | | |
| | BNOT #xx:3, @aa:8 | 2 | | | 2 | | |
| | BNOT Rn, Rd | 1 | | | | | |
| | BNOT Rn, @ERd | 2 | | | 2 | | |
| | BNOT Rn, @aa:8 | 2 | | | 2 | | |
| BOR | BOR #xx:3, Rd | 1 | | | | | |
| | BOR #xx:3, @ERd | 2 | | | 1 | | |
| | BOR #xx:3, @aa:8 | 2 | | | 1 | | |
| BSET | BSET #xx:3, Rd | 1 | | | | | |
| | BSET #xx:3, @ERd | 2 | | | 2 | | |
| | BSET #xx:3, @aa:8 | 2 | | | 2 | | |
| | BSET Rn, Rd | 1 | | | | | |
| | BSET Rn, @ERd | 2 | | | 2 | | |
| | BSET Rn, @aa:8 | 2 | | | 2 | | |
| BSR | BSR d:8 | 2 | | 1 | | | |
| | BSR d:16 | 2 | | 1 | | | 2 |
| BST | BST #xx:3, Rd | 1 | | | | | |
| | BST #xx:3, @ERd | 2 | | | 2 | | |
| | BST #xx:3, @aa:8 | 2 | | | 2 | | |

| Instruction | Mnemonic | Instruction Fetch I | Branch Addr. Read J | Stack Operation K | Byte Data Access L | Word Data Access M | Internal Operation N |
|---|---|---|---|---|---|---|---|
| BTST | BTST #xx:3, Rd | 1 | | | | | |
| | BTST #xx:3, @ERd | 2 | | | 1 | | |
| | BTST #xx:3, @aa:8 | 2 | | | 1 | | |
| | BTST Rn, Rd | 1 | | | | | |
| | BTST Rn, @ERd | 2 | | | 1 | | |
| | BTST Rn, @aa:8 | 2 | | | 1 | | |
| BXOR | BXOR #xx:3, Rd | 1 | | | | | |
| | BXOR #xx:3, @ERd | 2 | | | 1 | | |
| | BXOR #xx:3, @aa:8 | 2 | | | 1 | | |
| CMP | CMP.B #xx:8, Rd | 1 | | | | | |
| | CMP.B Rs, Rd | 1 | | | | | |
| | CMP.W #xx:16, Rd | 2 | | | | | |
| | CMP.W Rs, Rd | 1 | | | | | |
| | CMP.L #xx:32, ERd | 3 | | | | | |
| | CMP.L ERs, ERd | 1 | | | | | |
| DAA | DAA Rd | 1 | | | | | |
| DAS | DAS Rd | 1 | | | | | |
| DEC | DEC.B Rd | 1 | | | | | |
| | DEC.W #1/2, Rd | 1 | | | | | |
| | DEC.L #1/2, ERd | 1 | | | | | |
| DUVXS | DIVXS.B Rs, Rd | 2 | | | | | 12 |
| | DIVXS.W Rs, ERd | 2 | | | | | 20 |
| DIVXU | DIVXU.B Rs, Rd | 1 | | | | | 12 |
| | DIVXU.W Rs, ERd | 1 | | | | | 20 |
| EEPMOV | EEPMOV.B | 2 | | | 2n+2*[1] | | |
| | EEPMOV.W | 2 | | | 2n+2*[1] | | |
| EXTS | EXTS.W Rd | 1 | | | | | |
| | EXTS.L ERd | 1 | | | | | |
| EXTU | EXTU.W Rd | 1 | | | | | |
| | EXTU.L ERd | 1 | | | | | |

| Instruction | Mnemonic | Instruction Fetch I | Branch Addr. Read J | Stack Operation K | Byte Data Access L | Word Data Access M | Internal Operation N |
|---|---|---|---|---|---|---|---|
| INC | INC.B Rd | 1 | | | | | |
| | INC.W #1/2, Rd | 1 | | | | | |
| | INC.L #1/2, ERd | 1 | | | | | |
| JMP | JMP @ERn | 2 | | | | | |
| | JMP @aa:24 | 2 | | | | | 2 |
| | JMP @@aa:8 | 2 | 1 | | | | 2 |
| JSR | JSR @ERn | 2 | | 1 | | | |
| | JSR @aa:24 | 2 | | 1 | | | 2 |
| | JSR @@aa:8 | 2 | 1 | 1 | | | |
| LDC | LDC #xx:8, CCR | 1 | | | | | |
| | LDC Rs, CCR | 1 | | | | | |
| | LDC@ERs, CCR | 2 | | | | 1 | |
| | LDC@(d:16, ERs), CCR | 3 | | | | 1 | |
| | LDC@(d:24,ERs), CCR | 5 | | | | 1 | |
| | LDC@ERs+, CCR | 2 | | | | 1 | 2 |
| | LDC@aa:16, CCR | 3 | | | | 1 | |
| | LDC@aa:24, CCR | 4 | | | | 1 | |
| MOV | MOV.B #xx:8, Rd | 1 | | | | | |
| | MOV.B Rs, Rd | 1 | | | | | |
| | MOV.B @ERs, Rd | 1 | | | 1 | | |
| | MOV.B @(d:16, ERs), Rd | 2 | | | 1 | | |
| | MOV.B @(d:24, ERs), Rd | 4 | | | 1 | | |
| | MOV.B @ERs+, Rd | 1 | | | 1 | | 2 |
| | MOV.B @aa:8, Rd | 1 | | | 1 | | |
| | MOV.B @aa:16, Rd | 2 | | | 1 | | |
| | MOV.B @aa:24, Rd | 3 | | | 1 | | |
| | MOV.B Rs, @Erd | 1 | | | 1 | | |
| | MOV.B Rs, @(d:16, ERd) | 2 | | | 1 | | |
| | MOV.B Rs, @(d:24, ERd) | 4 | | | 1 | | |
| | MOV.B Rs, @-ERd | 1 | | | 1 | | 2 |
| | MOV.B Rs, @aa:8 | 1 | | | 1 | | |

| Instruction | Mnemonic | Instruction Fetch I | Branch Addr. Read J | Stack Operation K | Byte Data Access L | Word Data Access M | Internal Operation N |
|---|---|---|---|---|---|---|---|
| MOV | MOV.B Rs, @aa:16 | 2 | | | 1 | | |
| | MOV.B Rs, @aa:24 | 3 | | | 1 | | |
| | MOV.W #xx:16, Rd | 2 | | | | | |
| | MOV.W Rs, Rd | 1 | | | | | |
| | MOV.W @ERs, Rd | 1 | | | | 1 | |
| | MOV.W @(d:16,ERs), Rd | 2 | | | | 1 | |
| | MOV.W @(d:24,ERs), Rd | 4 | | | | 1 | |
| | MOV.W @ERs+, Rd | 1 | | | | 1 | 2 |
| | MOV.W @aa:16, Rd | 2 | | | | 1 | |
| | MOV.W @aa:24, Rd | 3 | | | | 1 | |
| | MOV.W Rs, @ERd | 1 | | | | 1 | |
| | MOV.W Rs, @(d:16,ERd) | 2 | | | | 1 | |
| | MOV.W Rs, @(d:24,ERd) | 4 | | | | 1 | |
| | MOV.W Rs, @-ERd | 1 | | | | 1 | 2 |
| | MOV.W Rs, @aa:16 | 2 | | | | 1 | |
| | MOV.W Rs, @aa:24 | 3 | | | | 1 | |
| | MOV.L #xx:32, ERd | 3 | | | | | |
| | MOV.L ERs, ERd | 1 | | | | | |
| | MOV.L @ERs, ERd | 2 | | | | 2 | |
| | MOV.L @(d:16,ERs), ERd | 3 | | | | 2 | |
| | MOV.L @(d:24,ERs), ERd | 5 | | | | 2 | |
| | MOV.L @ERs+, ERd | 2 | | | | 2 | 2 |
| | MOV.L @aa:16, ERd | 3 | | | | 2 | |
| | MOV.L @aa:24, ERd | 4 | | | | 2 | |
| | MOV.L ERs,@ERd | 2 | | | | 2 | |
| | MOV.L ERs, @(d:16,ERd) | 3 | | | | 2 | |
| | MOV.L ERs, @(d:24,ERd) | 5 | | | | 2 | |
| | MOV.L ERs, @-ERd | 2 | | | | 2 | 2 |
| | MOV.L ERs, @aa:16 | 3 | | | | 2 | |
| | MOV.L ERs, @aa:24 | 4 | | | | 2 | |
| MOVFPE | MOVFPE @aa:16, Rd*[2] | 2 | | | 1 | | |
| MOVTPE | MOVTPE Rs,@aa:16*[2] | 2 | | | 1 | | |

| Instruction | Mnemonic | Instruction Fetch I | Branch Addr. Read J | Stack Operation K | Byte Data Access L | Word Data Access M | Internal Operation N |
|---|---|---|---|---|---|---|---|
| MULXS | MULXS.B Rs, Rd | 2 | | | | | 12 |
| | MULXS.W Rs, ERd | 2 | | | | | 20 |
| MULXU | MULXU.B Rs, Rd | 1 | | | | | 12 |
| | MULXU.W Rs, ERd | 1 | | | | | 20 |
| NEG | NEG.B Rd | 1 | | | | | |
| | NEG.W Rd | 1 | | | | | |
| | NEG.L ERd | 1 | | | | | |
| NOP | NOP | 1 | | | | | |
| NOT | NOT.B Rd | 1 | | | | | |
| | NOT.W Rd | 1 | | | | | |
| | NOT.L ERd | 1 | | | | | |
| OR | OR.B #xx:8, Rd | 1 | | | | | |
| | OR.B Rs, Rd | 1 | | | | | |
| | OR.W #xx:16, Rd | 2 | | | | | |
| | OR.W Rs, Rd | 1 | | | | | |
| | OR.L #xx:32, ERd | 3 | | | | | |
| | OR.L ERs, ERd | 2 | | | | | |
| ORC | ORC #xx:8, CCR | 1 | | | | | |
| POP | POP.W Rn | 1 | | | | 1 | 2 |
| | POP.L ERn | 2 | | | | 2 | 2 |
| PUSH | PUSH.W Rn | 1 | | | | 1 | 2 |
| | PUSH.L ERn | 2 | | | | 2 | 2 |
| ROTL | ROTL.B Rd | 1 | | | | | |
| | ROTL.W Rd | 1 | | | | | |
| | ROTL.L ERd | 1 | | | | | |
| ROTR | ROTR.B Rd | 1 | | | | | |
| | ROTR.W Rd | 1 | | | | | |
| | ROTR.L ERd | 1 | | | | | |
| ROTXL | ROTXL.B Rd | 1 | | | | | |
| | ROTXL.W Rd | 1 | | | | | |
| | ROTXL.L ERd | 1 | | | | | |

| Instruction | Mnemonic | Instruction Fetch I | Branch Addr. Read J | Stack Operation K | Byte Data Access L | Word Data Access M | Internal Operation N |
|---|---|---|---|---|---|---|---|
| ROTXR | ROTXR.B Rd | 1 | | | | | |
| | ROTXR.W Rd | 1 | | | | | |
| | ROTXR.L ERd | 1 | | | | | |
| RTE | RTE | 2 | | 2 | | | 2 |
| RTS | RTS | 2 | | 1 | | | 2 |
| SHAL | SHAL.B Rd | 1 | | | | | |
| | SHAL.W Rd | 1 | | | | | |
| | SHAL.L ERd | 1 | | | | | |
| SHAR | SHAR.B Rd | 1 | | | | | |
| | SHAR.W Rd | 1 | | | | | |
| | SHAR.L ERd | 1 | | | | | |
| SHLL | SHLL.B Rd | 1 | | | | | |
| | SHLL.W Rd | 1 | | | | | |
| | SHLL.L ERd | 1 | | | | | |
| SHLR | SHLR.B Rd | 1 | | | | | |
| | SHLR.W Rd | 1 | | | | | |
| | SHLR.L ERd | 1 | | | | | |
| SLEEP | SLEEP | 1 | | | | | |
| STC | STC CCR, Rd | 1 | | | | | |
| | STC CCR, @ERd | 2 | | | | 1 | |
| | STC CCR, @(d:16,ERd) | 3 | | | | 1 | |
| | STC CCR, @(d:24,ERd) | 5 | | | | 1 | |
| | STC CCR,@-ERd | 2 | | | | 1 | 2 |
| | STC CCR, @aa:16 | 3 | | | | 1 | |
| | STC CCR, @aa:24 | 4 | | | | 1 | |
| SUB | SUB.B Rs, Rd | 1 | | | | | |
| | SUB.W #xx:16, Rd | 2 | | | | | |
| | SUB.W Rs, Rd | 1 | | | | | |
| | SUB.L #xx:32, ERd | 3 | | | | | |
| | SUB.L ERs, ERd | 1 | | | | | |
| SUBS | SUBS #1/2/4, ERd | 1 | | | | | |

| Instruction | Mnemonic | Instruction Fetch I | Branch Addr. Read J | Stack Operation K | Byte Data Access L | Word Data Access M | Internal Operation N |
|---|---|---|---|---|---|---|---|
| SUBX | SUBX #xx:8, Rd | 1 | | | | | |
| | SUBX. Rs, Rd | 1 | | | | | |
| TRAPA | TRAPA #xx:2 | 2 | 1 | 2 | | | 4 |
| XOR | XOR.B #xx:8, Rd | 1 | | | | | |
| | XOR.B Rs, Rd | 1 | | | | | |
| | XOR.W #xx:16, Rd | 2 | | | | | |
| | XOR.W Rs, Rd | 1 | | | | | |
| | XOR.L #xx:32, ERd | 3 | | | | | |
| | XOR.L ERs, ERd | 2 | | | | | |
| XORC | XORC #xx:8, CCR | 1 | | | | | |

Notes: 1. n: Specified value in R4L and R4. The source and destination operands are accessed n+1 times respectively.
2. Cannot be used in this LSI.

## A.4 Combinations of Instructions and Addressing Modes

### Table A.5 Combinations of Instructions and Addressing Modes

| Functions | Instructions | Addressing Mode | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | #xx | Rn | @ERn | @(d:16.ERn) | @(d:24.ERn) | @ERn+/@ERn | @aa:8 | @aa:16 | @aa:24 | @(d:8.PC) | @(d:16.PC) | @@aa:8 | — |
| Data transfer instructions | MOV | BWL | BWL | BWL | BWL | BWL | BWL | B | BWL | BWL | — | — | — | — |
| | POP, PUSH | — | — | — | — | — | — | — | — | — | — | — | — | WL |
| | MOVFPE, MOVTPE | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Arithmetic operations | ADD, CMP | BWL | BWL | — | — | — | — | — | — | — | — | — | — | — |
| | SUB | WL | BWL | — | — | — | — | — | — | — | — | — | — | — |
| | ADDX, SUBX | B | B | — | — | — | — | — | — | — | — | — | — | — |
| | ADDS, SUBS | — | L | — | — | — | — | — | — | — | — | — | — | — |
| | INC, DEC | — | BWL | — | — | — | — | — | — | — | — | — | — | — |
| | DAA, DAS | — | B | — | — | — | — | — | — | — | — | — | — | — |
| | MULXU, MULXS, DIVXU, DIVXS | — | BW | — | — | — | — | — | — | — | — | — | — | — |
| | NEG | — | BWL | — | — | — | — | — | — | — | — | — | — | — |
| | EXTU, EXTS | — | WL | — | — | — | — | — | — | — | — | — | — | — |
| Logical operations | AND, OR, XOR | — | BWL | — | — | — | — | — | — | — | — | — | — | — |
| | NOT | — | BWL | — | — | — | — | — | — | — | — | — | — | — |
| Shift operations | | — | BWL | — | — | — | — | — | — | — | — | — | — | — |
| Bit manipulations | | — | B | B | — | — | — | B | — | — | — | — | — | — |
| Branching instructions | BCC, BSR | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | JMP, JSR | — | — | ◯ | — | — | — | — | — | — | ◯ | ◯ | — | — |
| | RTS | — | — | — | — | — | — | — | — | ◯ | — | — | ◯ | — |
| System control instructions | TRAPA | — | — | — | — | — | — | — | — | — | — | — | — | ◯ |
| | RTE | — | — | — | — | — | — | — | — | — | — | — | — | ◯ |
| | SLEEP | — | — | — | — | — | — | — | — | — | — | — | — | ◯ |
| | LDC | B | B | W | W | W | W | — | W | W | — | — | — | ◯ |
| | STC | — | B | W | W | W | W | — | W | W | — | — | — | — |
| | ANDC, ORC, XORC | B | — | — | — | — | — | — | — | — | — | — | — | — |
| | NOP | — | — | — | — | — | — | — | — | — | — | — | — | ◯ |
| Block data transfer instructions | | — | — | — | — | — | — | — | — | — | — | — | — | BW |

# B. I/O Port Block Diagrams

## B.1 I/O Port Block Diagrams

$\overline{RES}$ goes low in a reset, and $\overline{SBY}$ goes low at a reset and in standby mode.

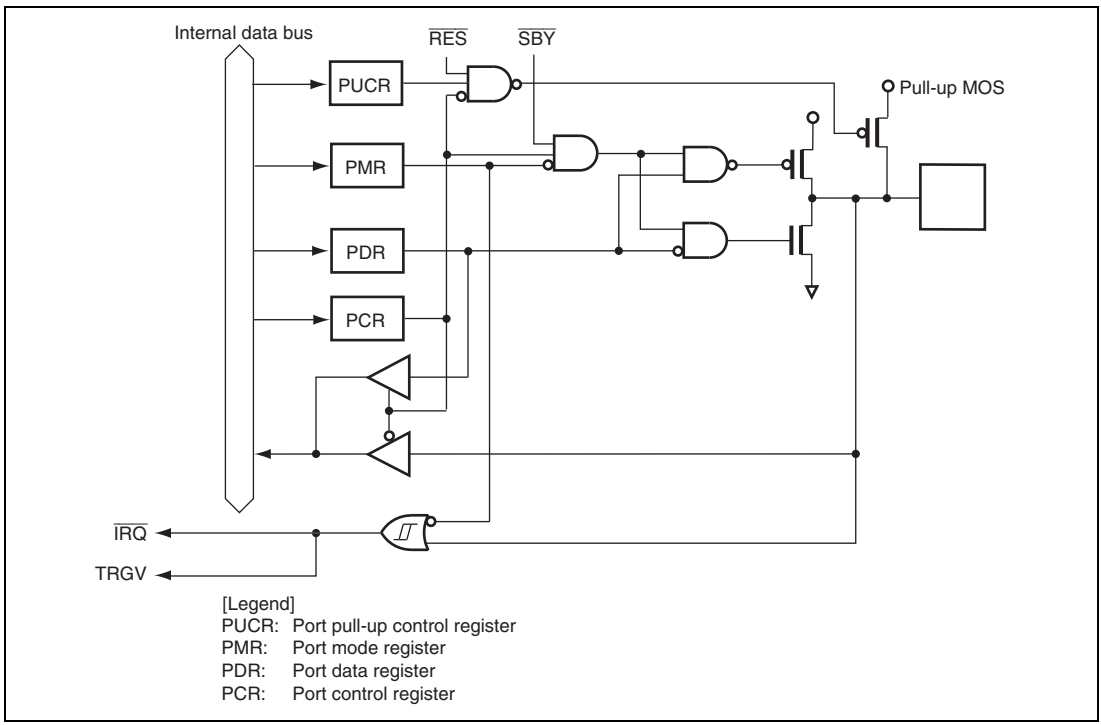



**Figure B.1 Port 1 Block Diagram (P17)**

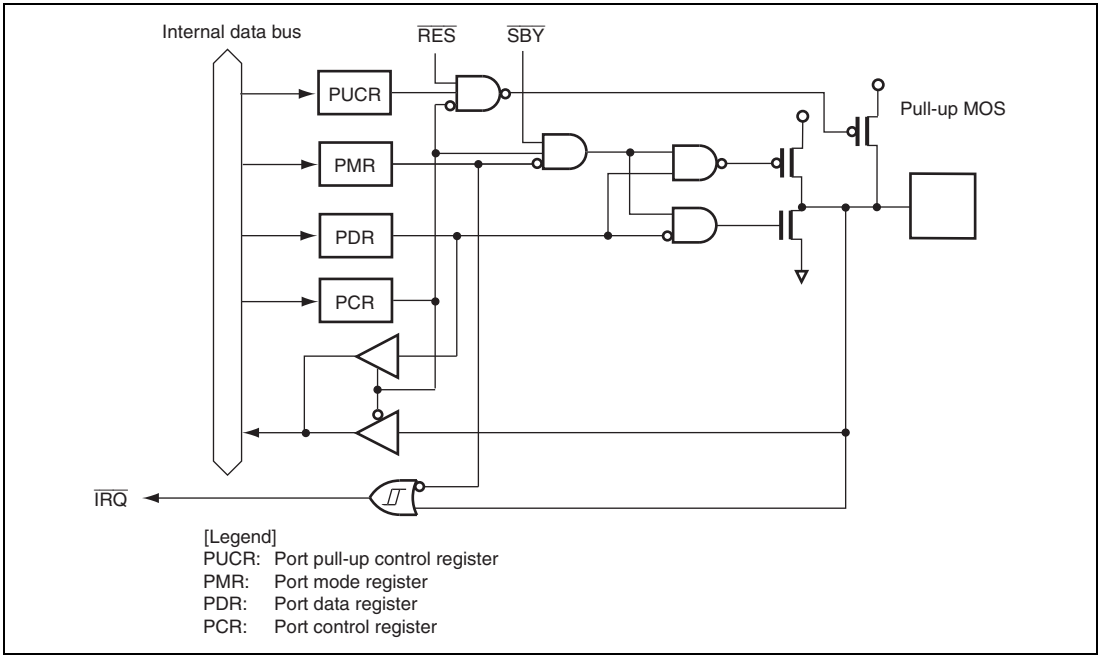



**Figure B.2 Port 1 Block Diagram (P16, P14)**

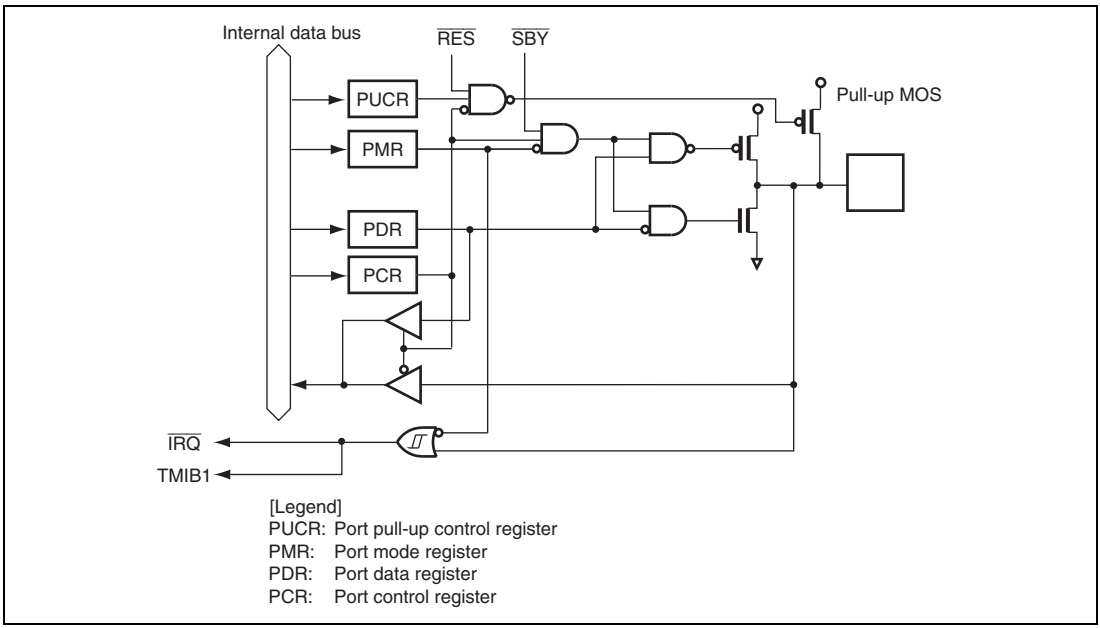



**Figure B.3 Port 1 Block Diagram (P15)**

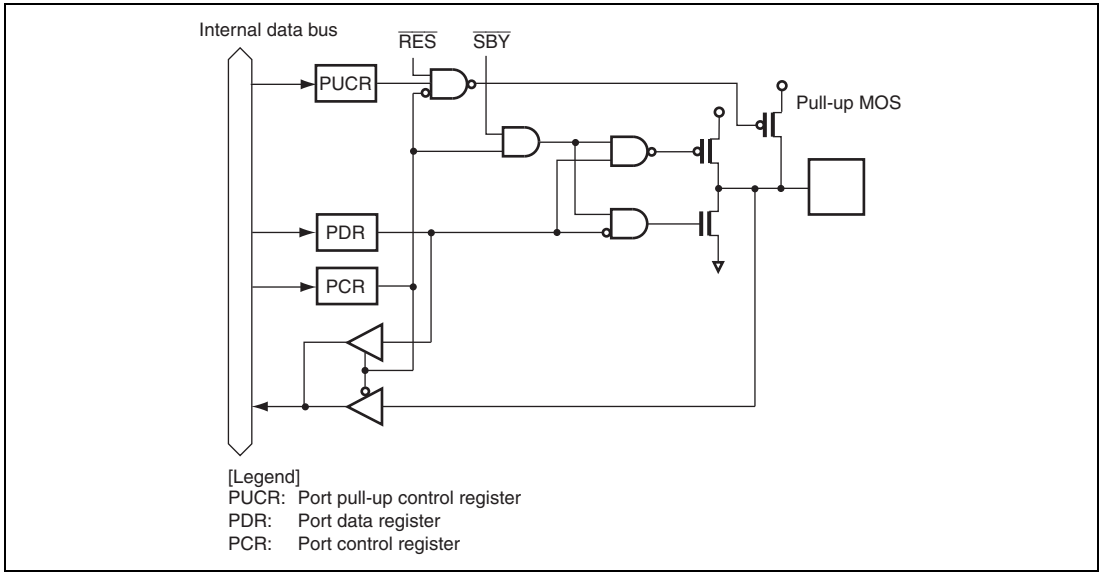



**Figure B.4 Port 1 Block Diagram (P12)**

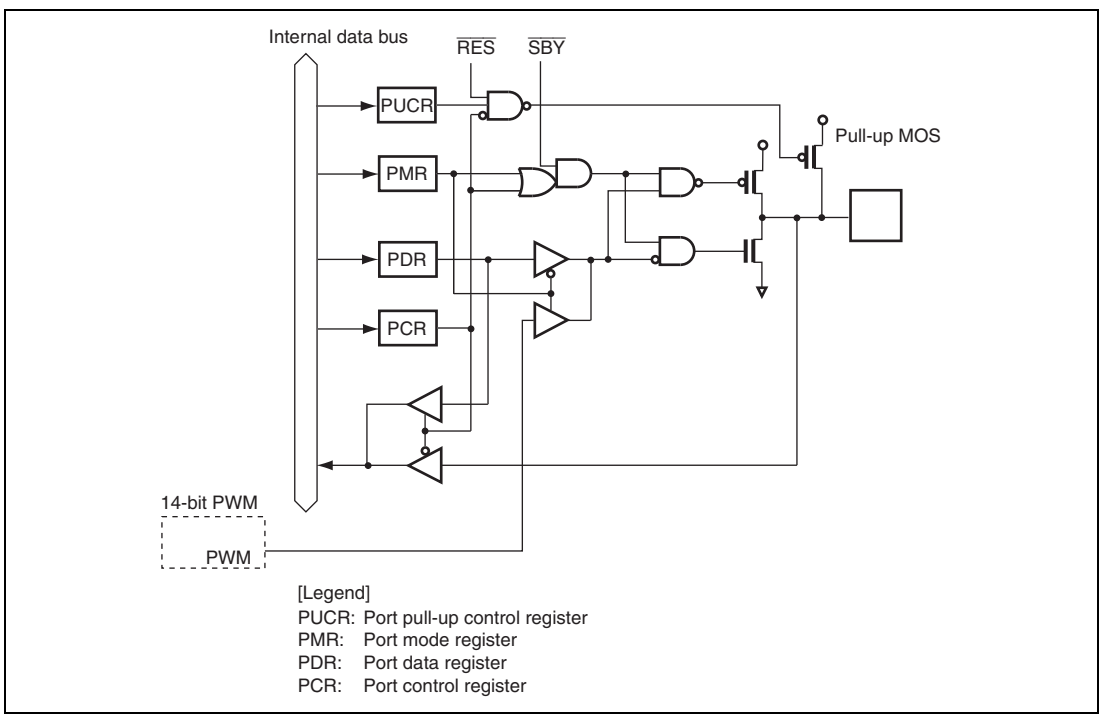



**Figure B.5 Port 1 Block Diagram (P11)**

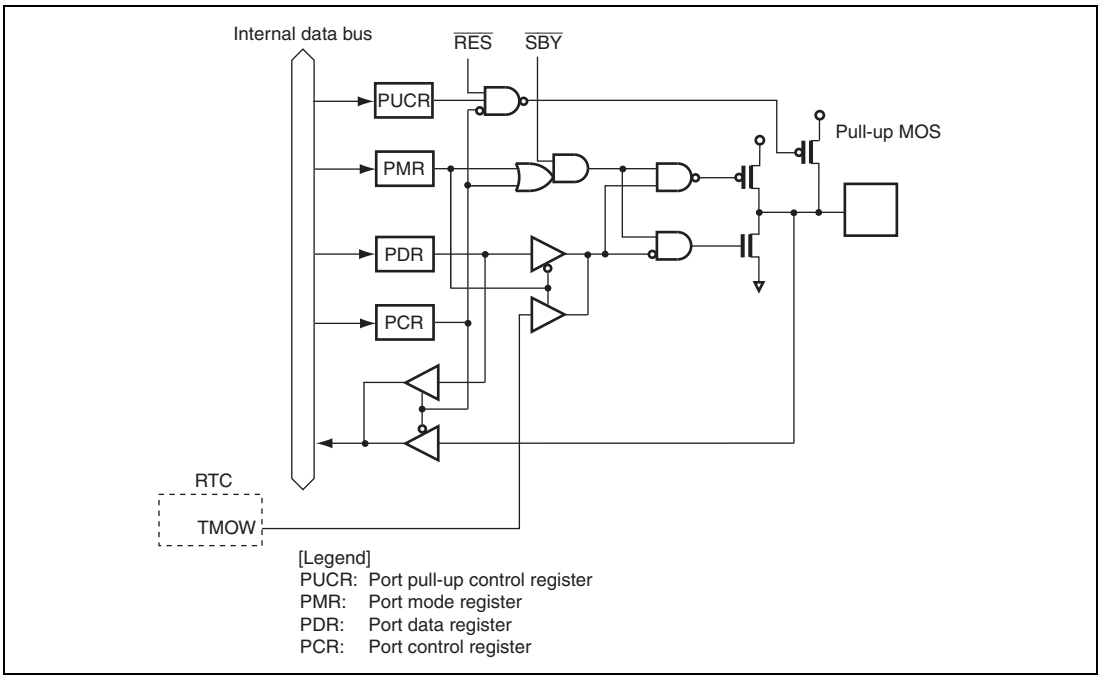



**Figure B.6 Port 1 Block Diagram (P10)**

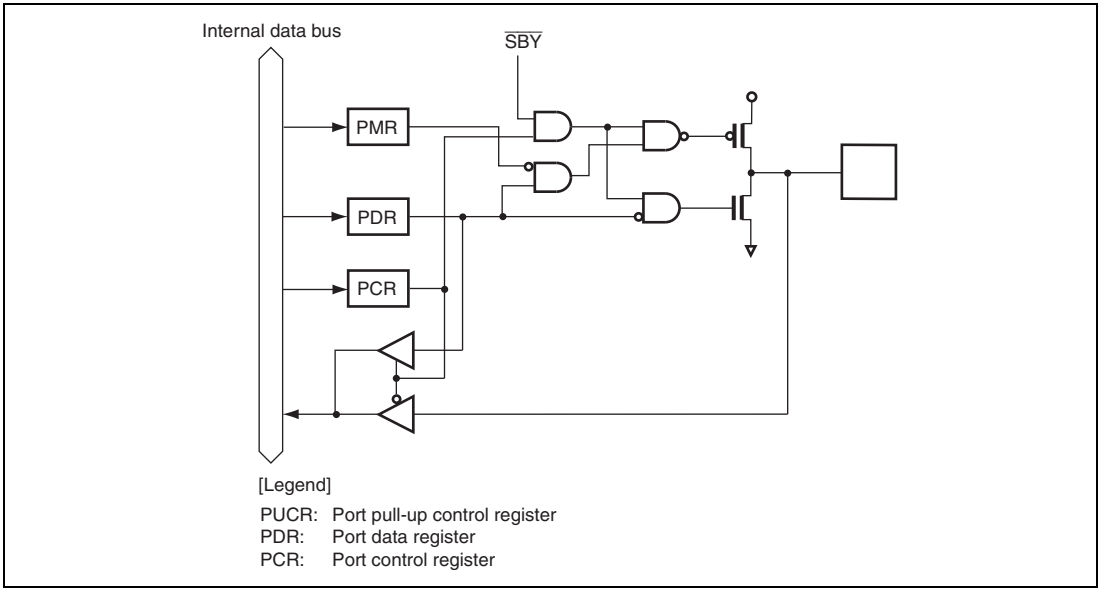



**Figure B.7 Port 2 Block Diagram (P27, P26, P25, P24, P23)**

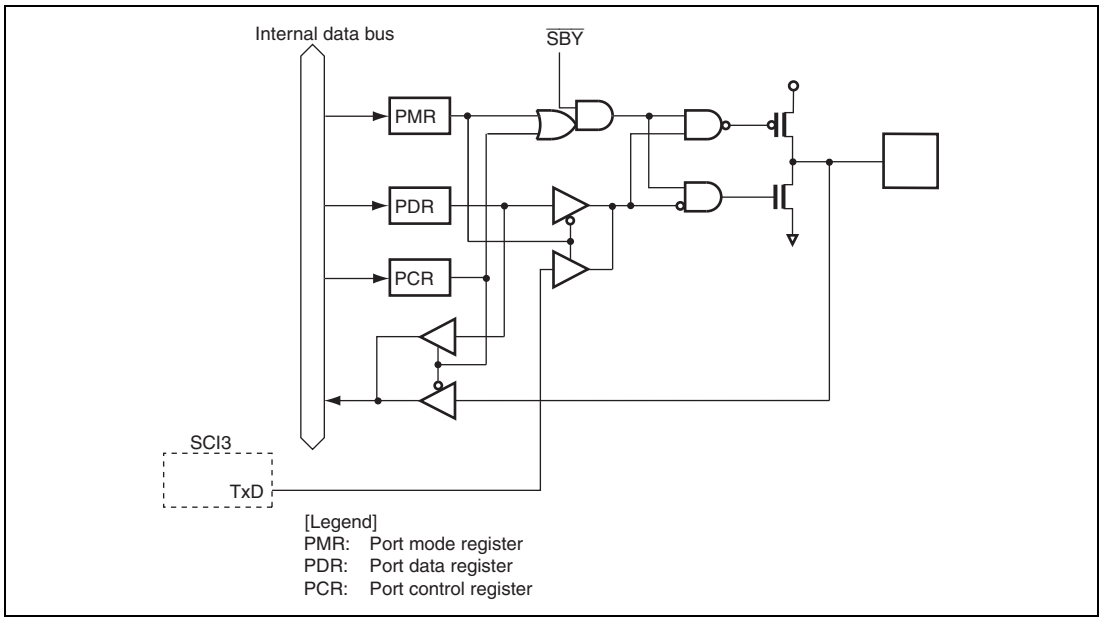



**Figure B.8 Port 2 Block Diagram (P22)**

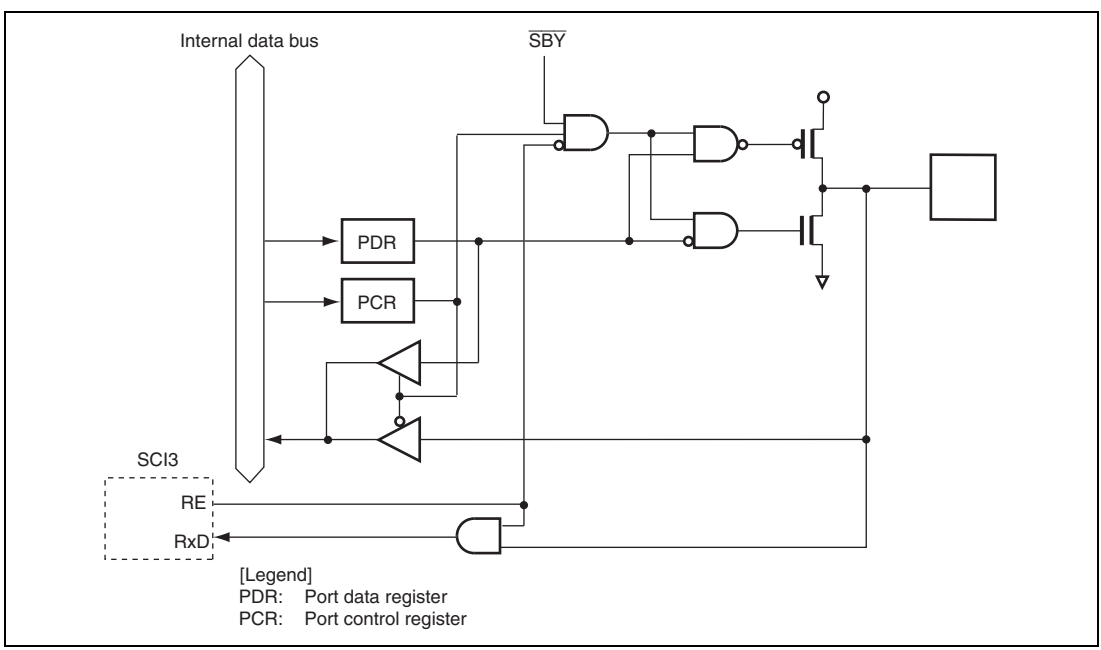



**Figure B.9 Port 2 Block Diagram (P21)**

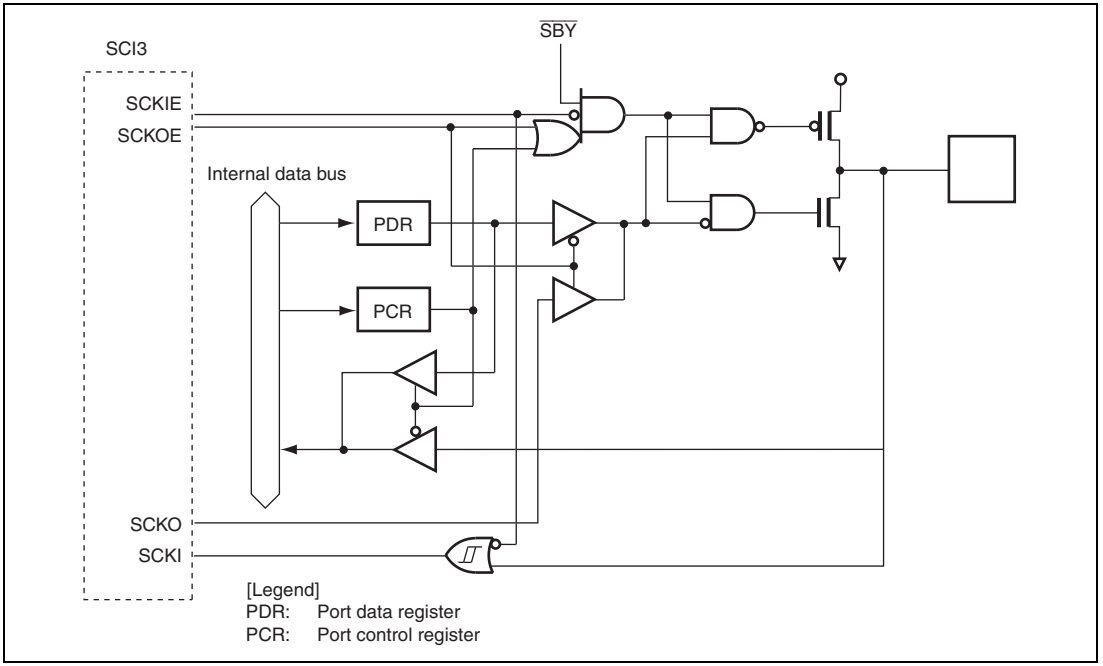



**Figure B.10 Port 2 Block Diagram (P20)**

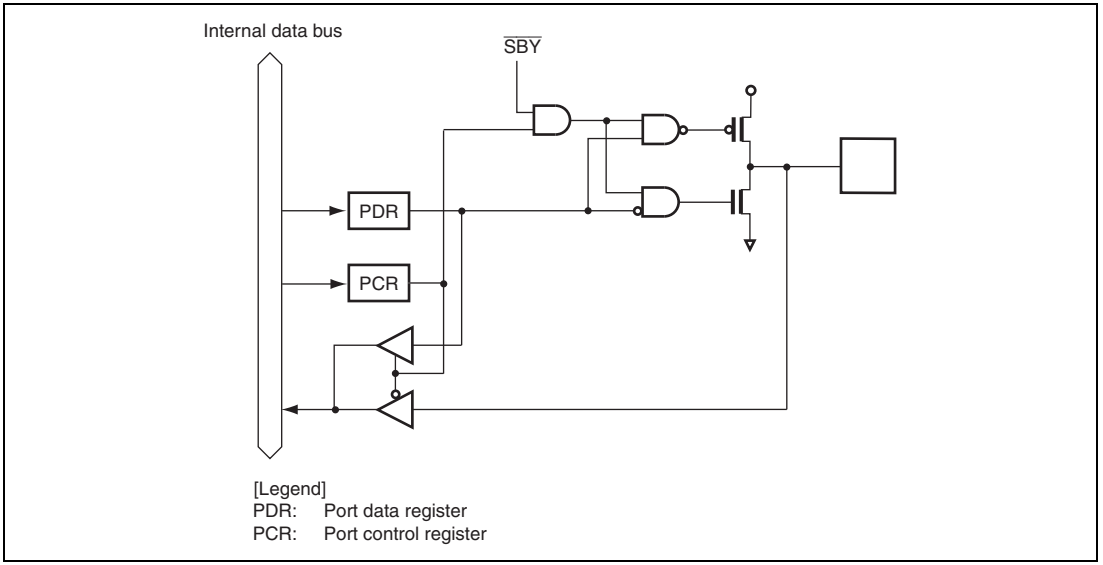



**Figure B.11 Port 3 Block Diagram (P37, P36, P35, P34, P33, P32, P31, P30)**

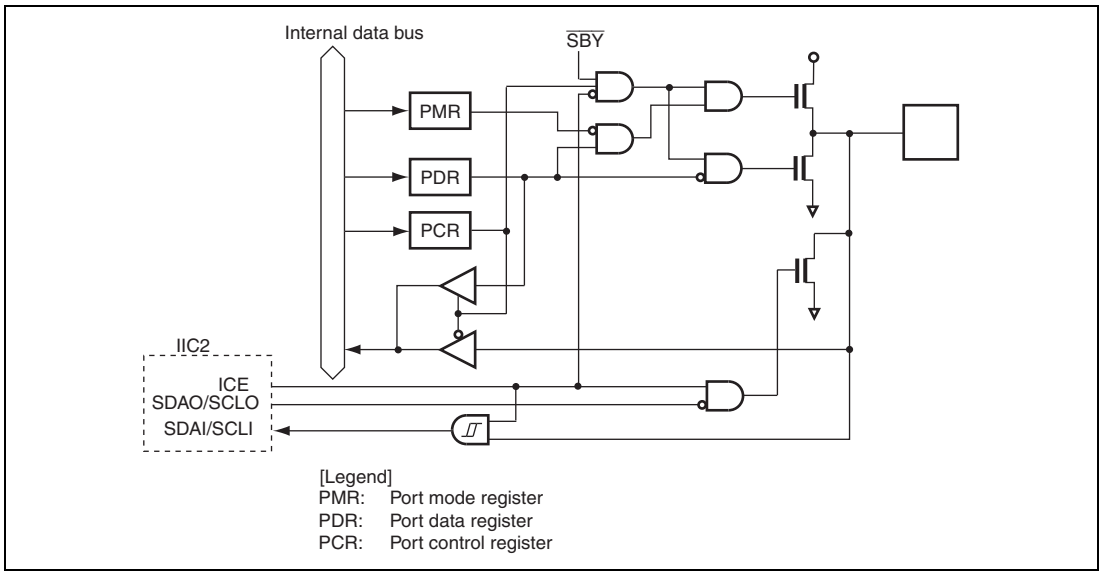



**Figure B.12 Port5 Block Diagram (P57, P56)**

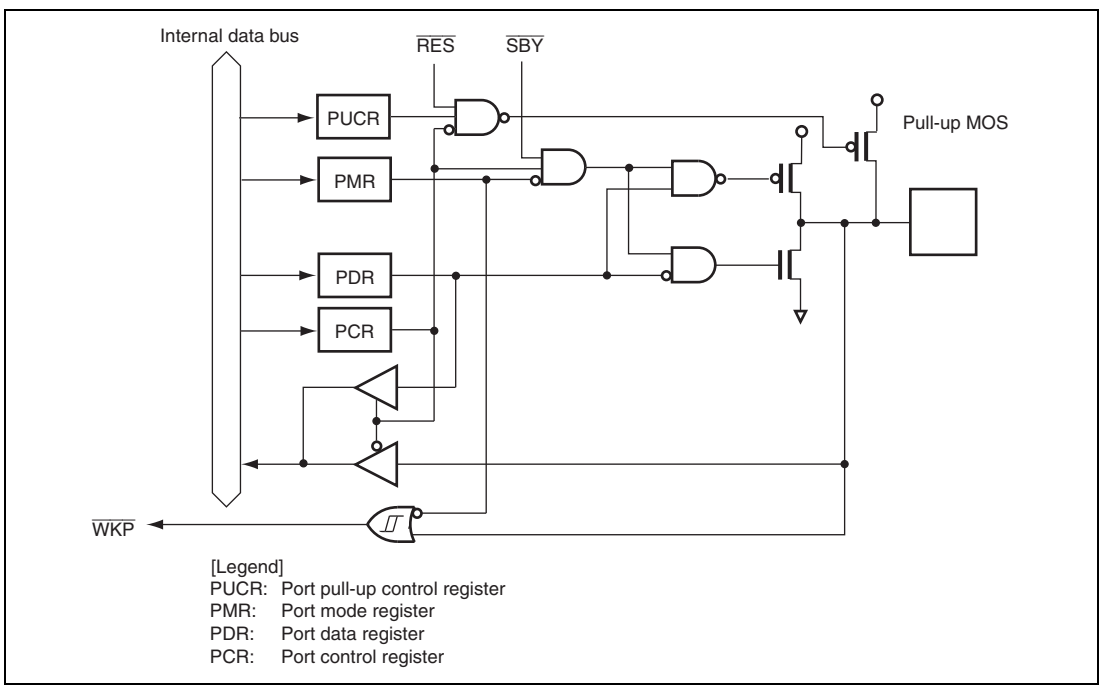



**Figure B.13 Port 5 Block Diagram (P55, P54, P53, P52, P51, P50)**

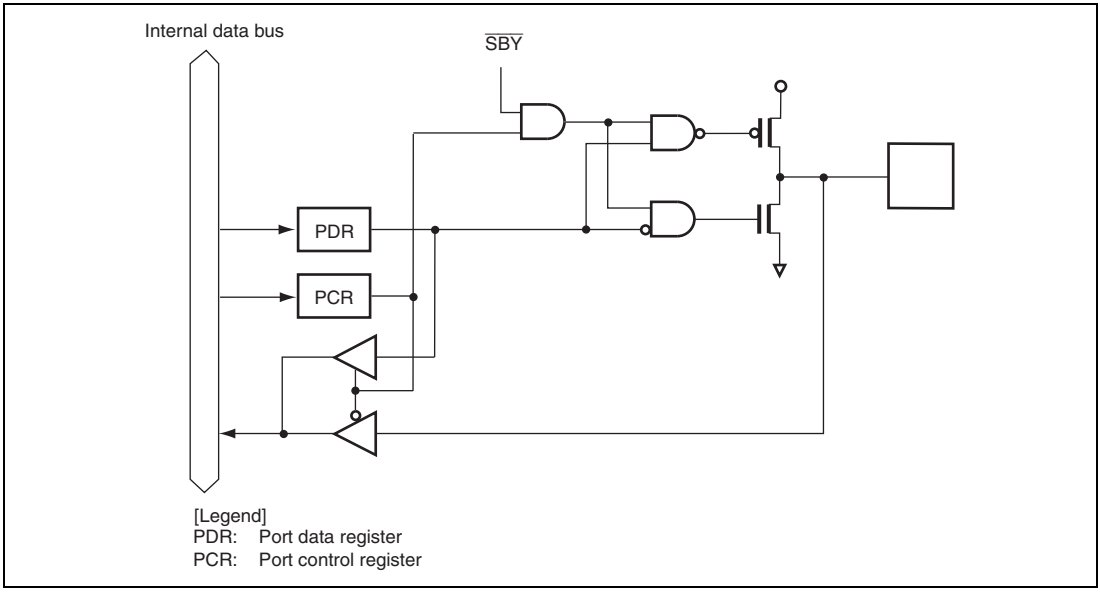



**Figure B.14 Port 7 Block Diagram (P77)**

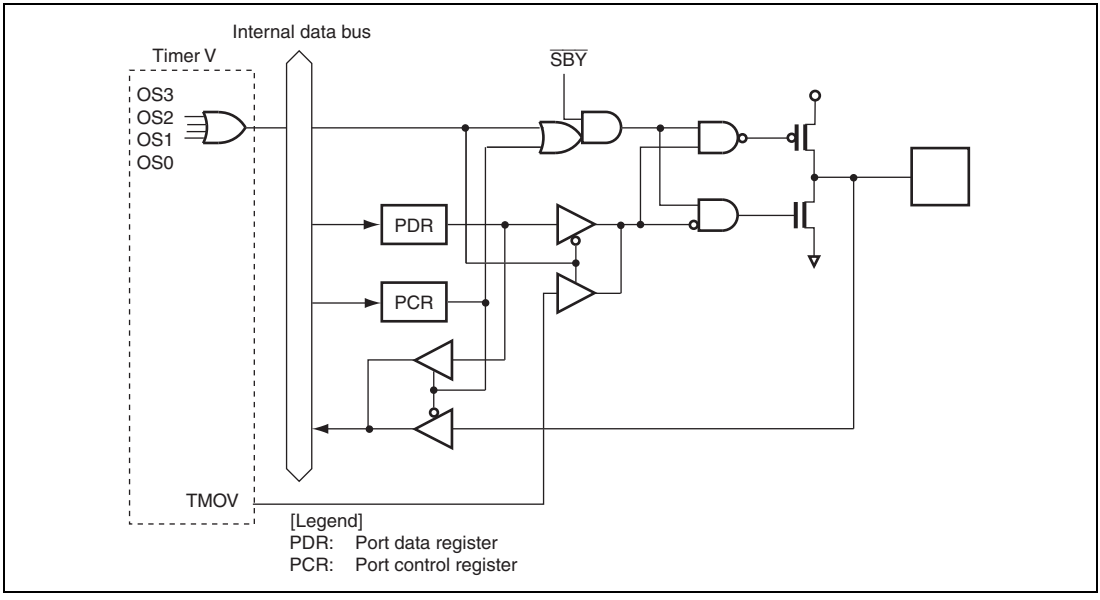



**Figure B.15 Port 7 Block Diagram (P76)**

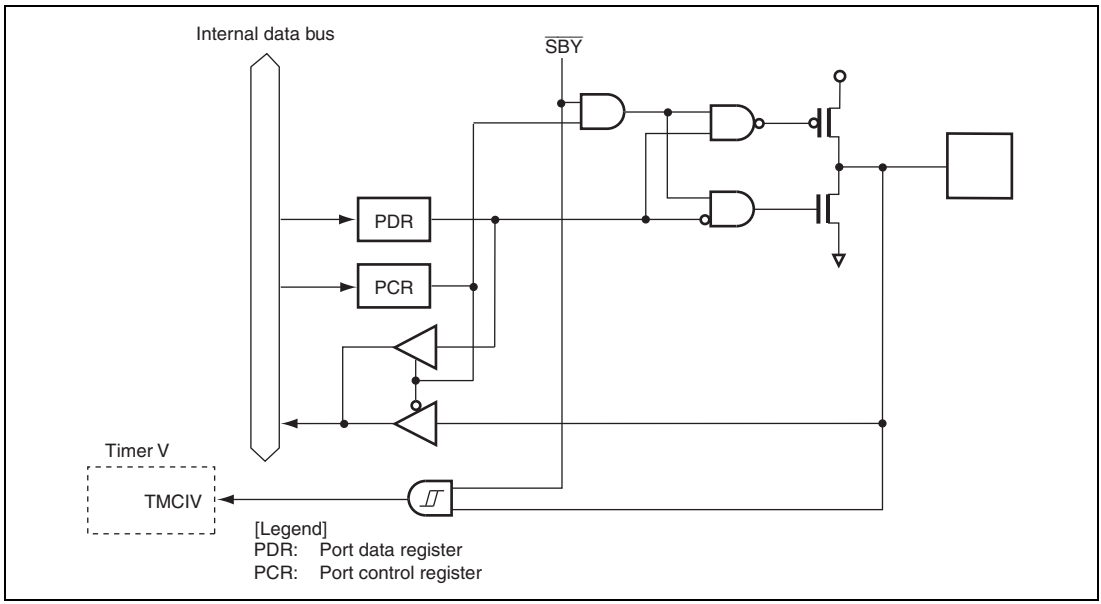



**Figure B.16 Port 7 Block Diagram (P75)**

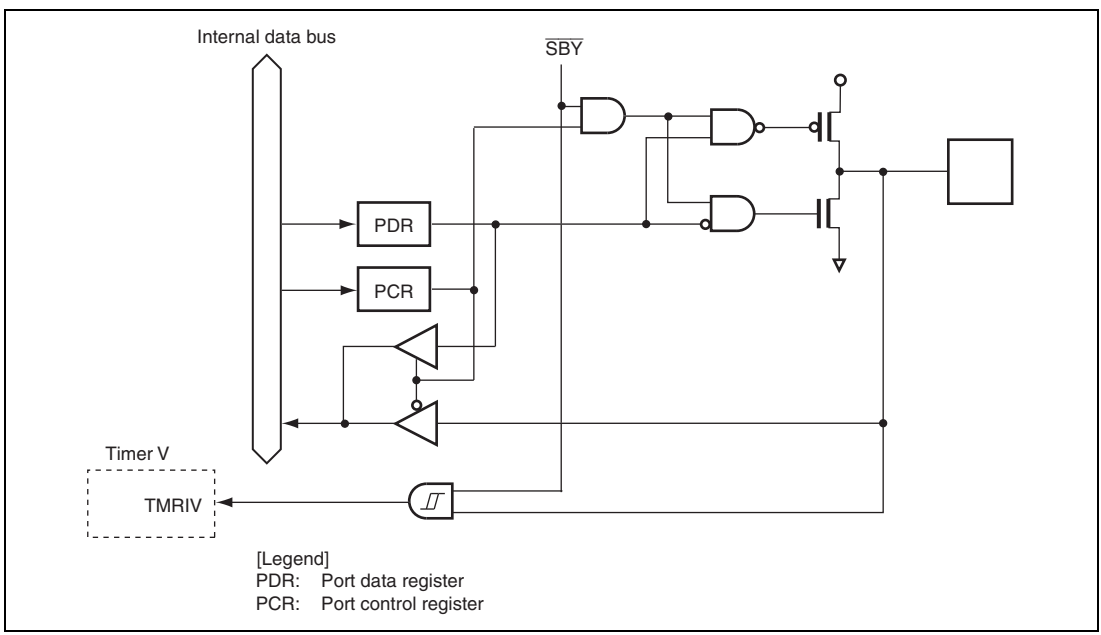



**Figure B.17 Port 7 Block Diagram (P74)**

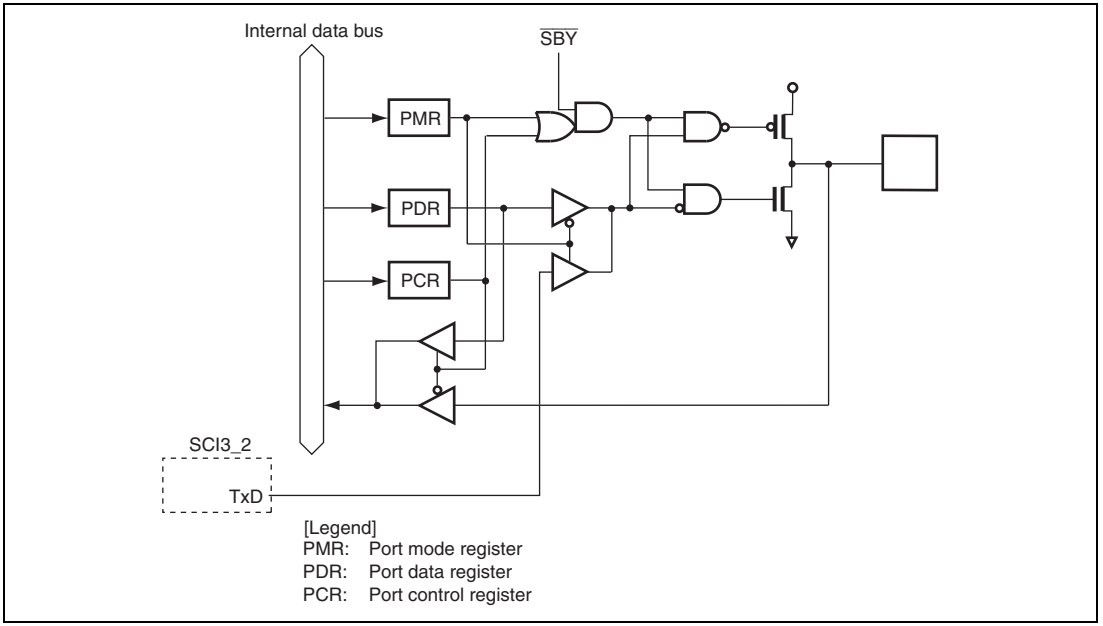



**Figure B.18 Port 7 Block Diagram (P72)**

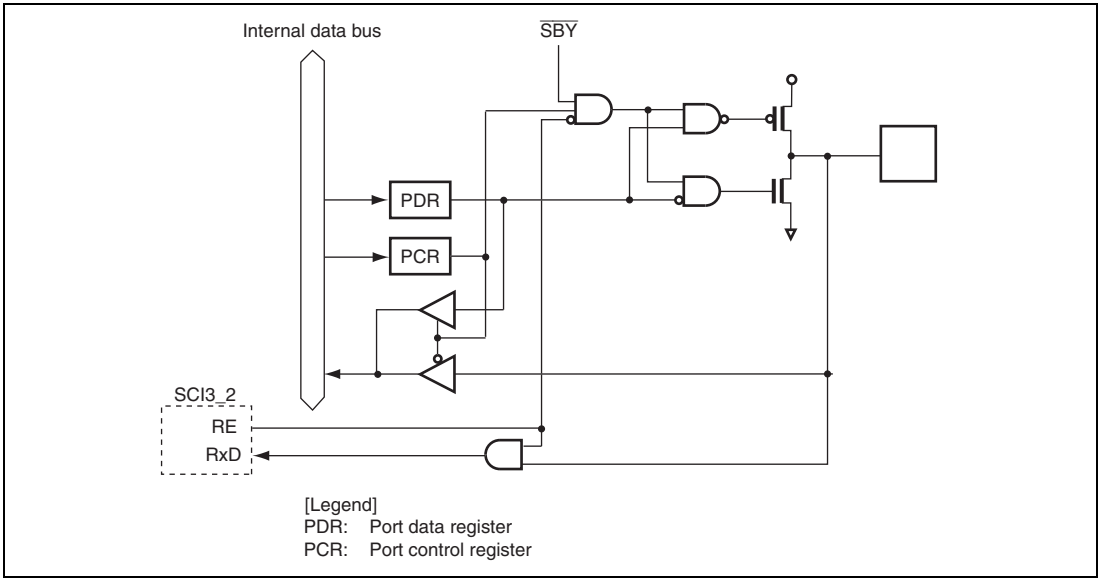



**Figure B.19 Port 7 Block Diagram (P71)**

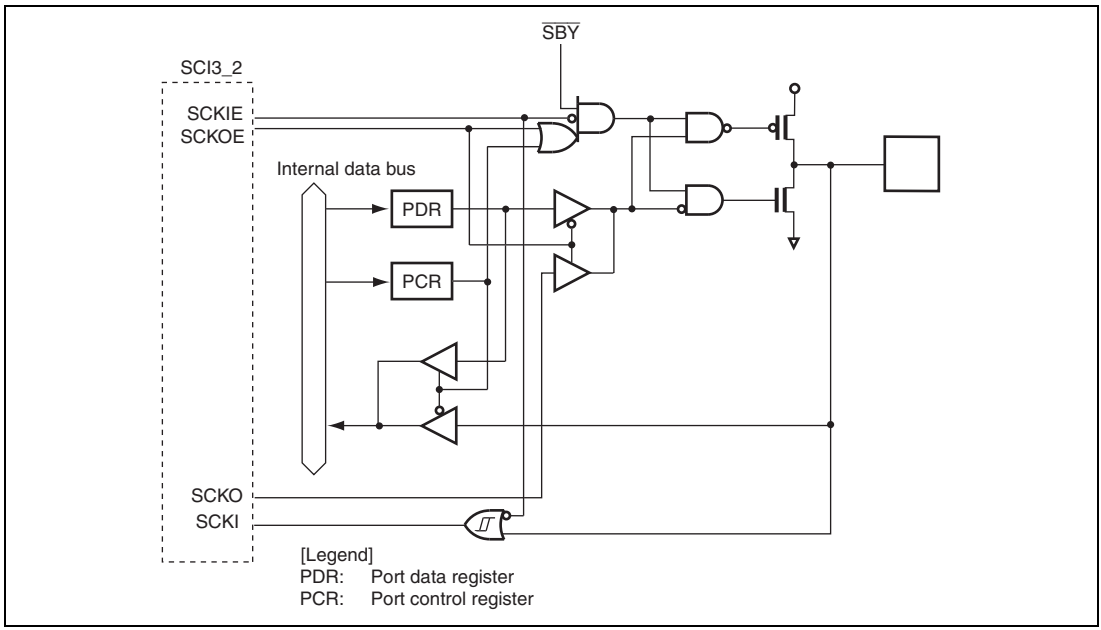



**Figure B.20 Port 7 Block Diagram (P70)**

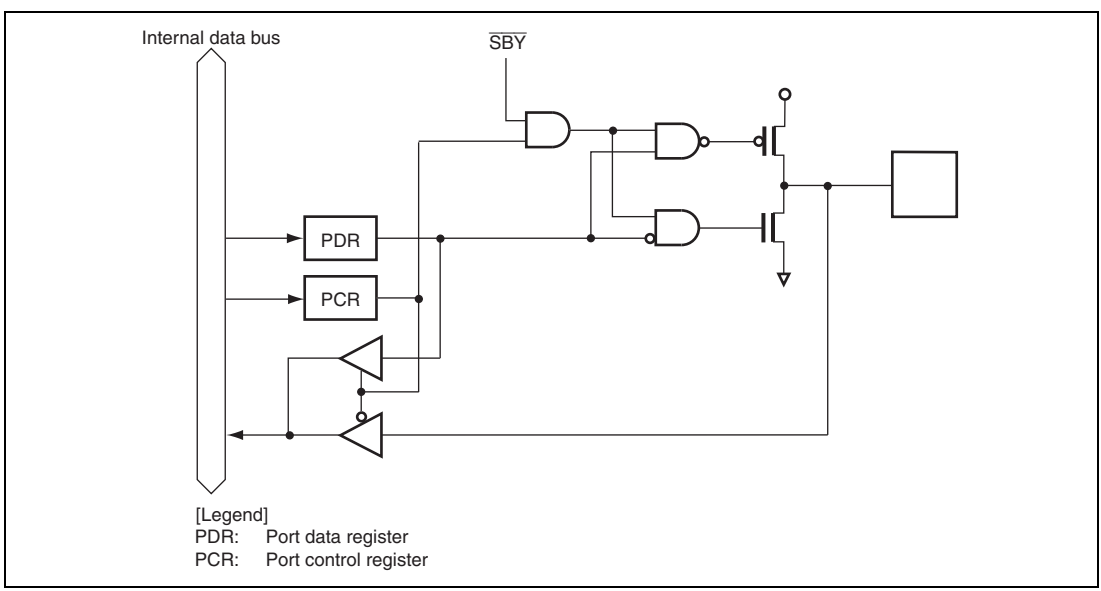



**Figure B.21 Port 8 Block Diagram (P87, P86, P85)**

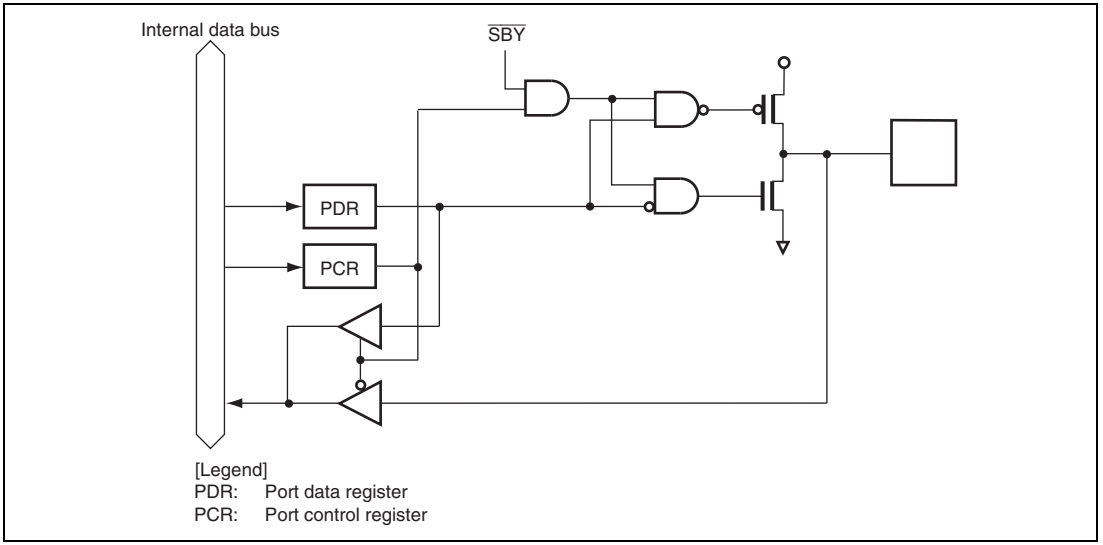



**Figure B.22 Port C Block Diagram (PC3, PC2, PC1, PC0)**

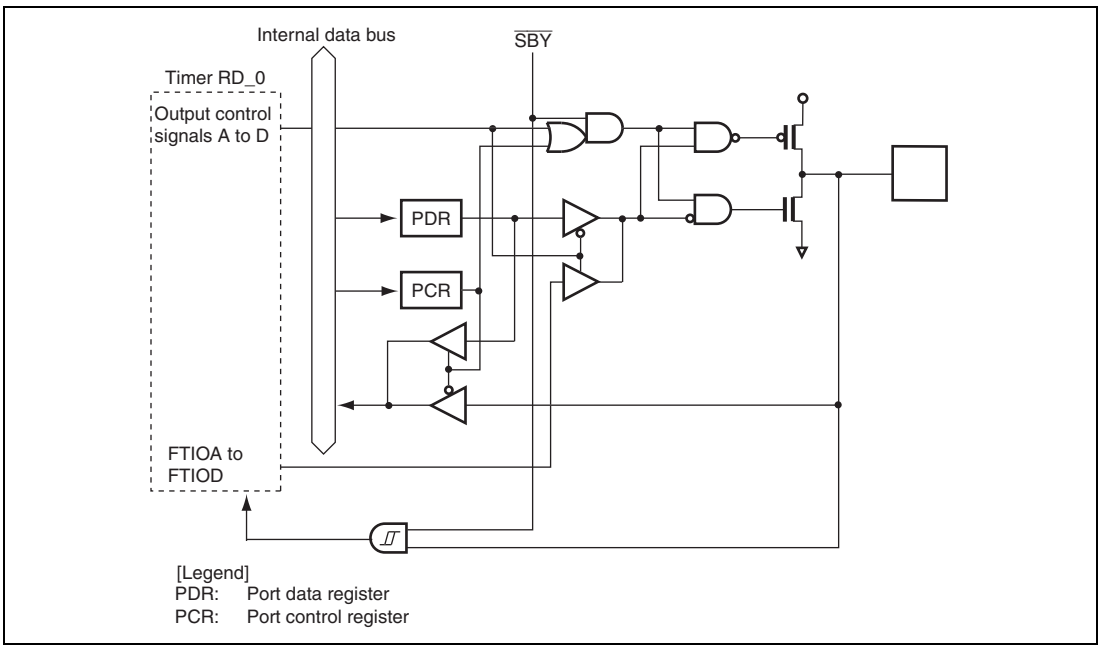



**Figure B.23 Port D Block Diagram (PD7, PD6, PD5, PD4, PD3, PD2, PD1, PD0)**

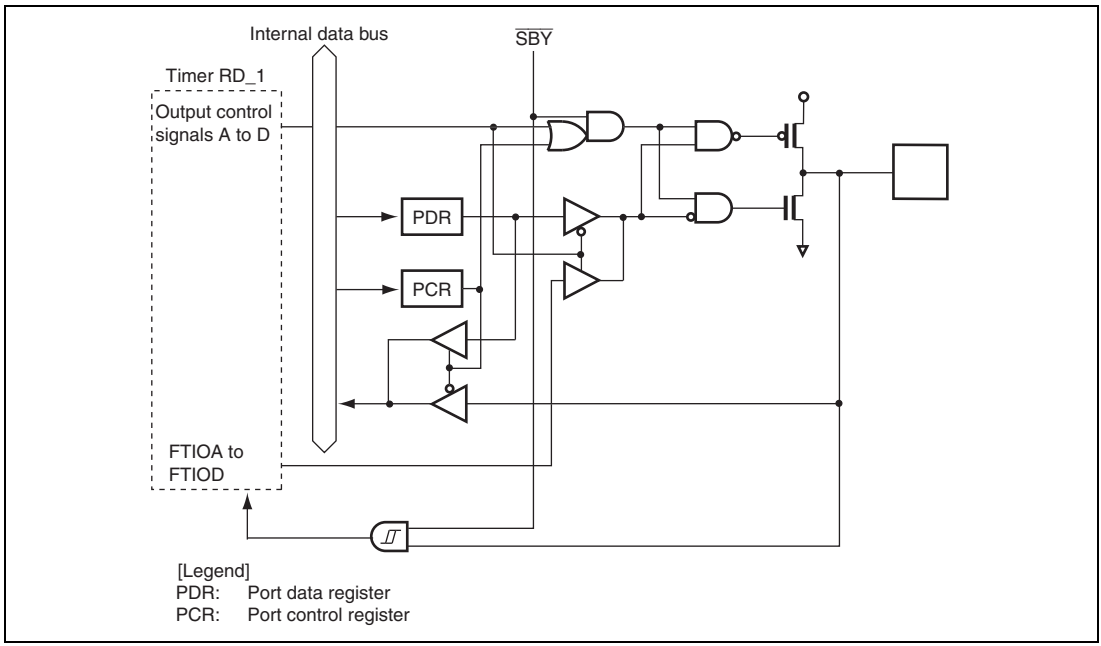



**Figure B.24 Port E Block Diagram (PE7, PE6, PE5, PE4, PE3, PE2, PE1, PE0)**

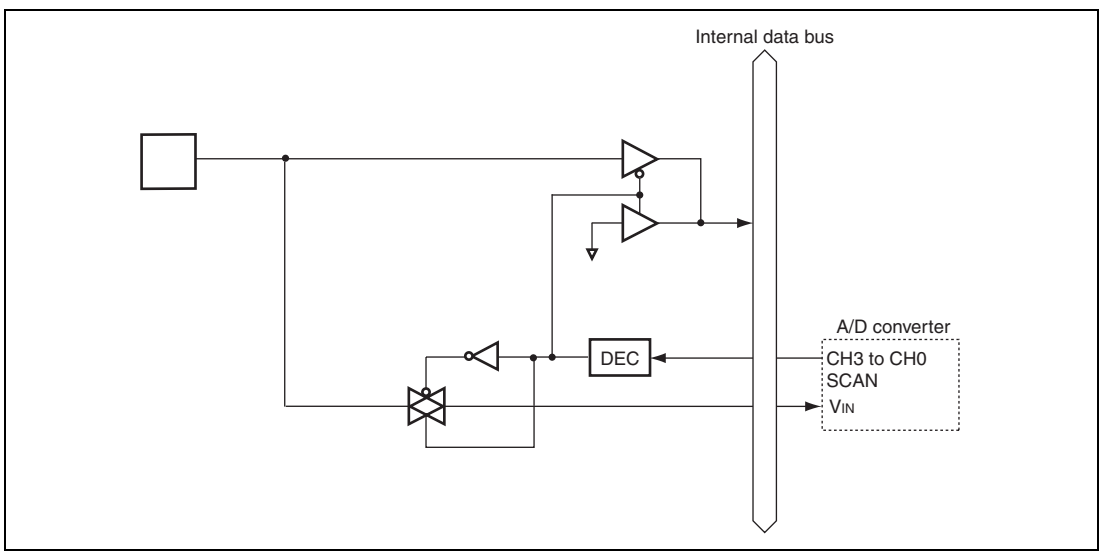



**Figure B.25 Port F Block Diagram (PF7, PF6, PF5, PF4, PF3, PF2, PF1, PF0)**

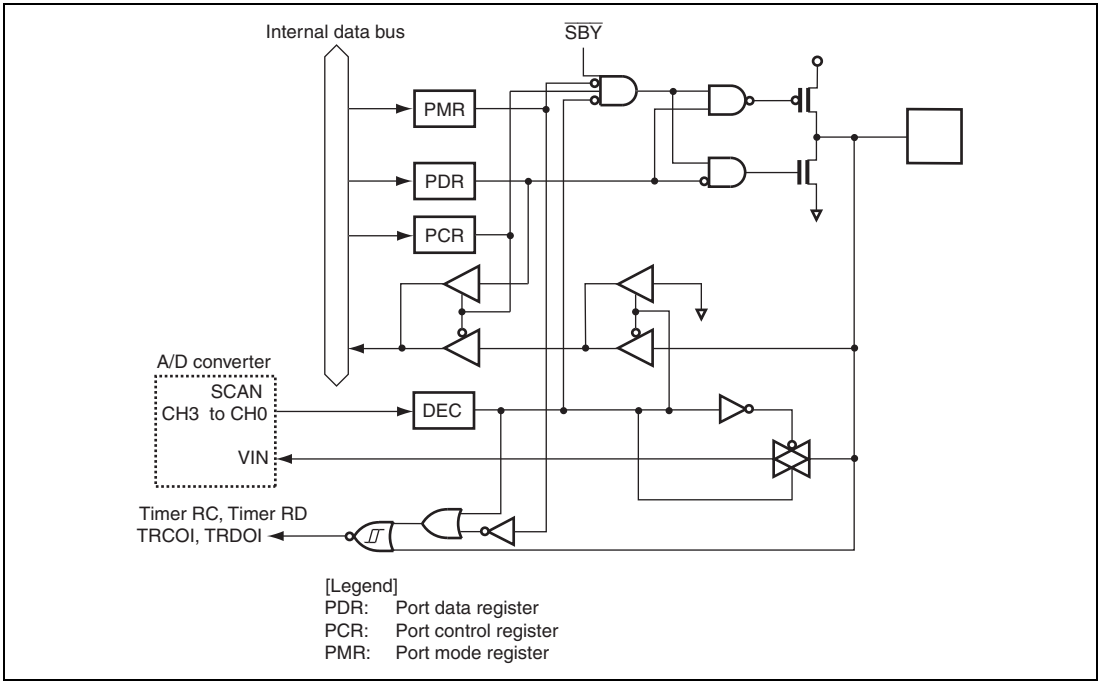



**Figure B.26 Port G Block Diagram (PG7, PG6, PG5)**

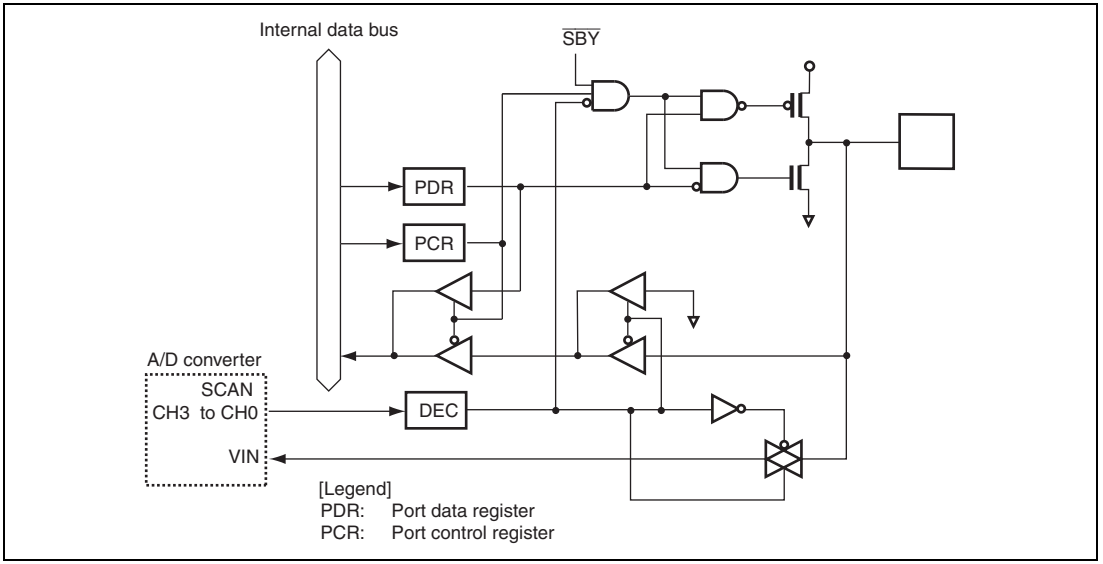



**Figure B.27 Port G Block Diagram (PG4, PG3, PG2, PG1, PG0)**

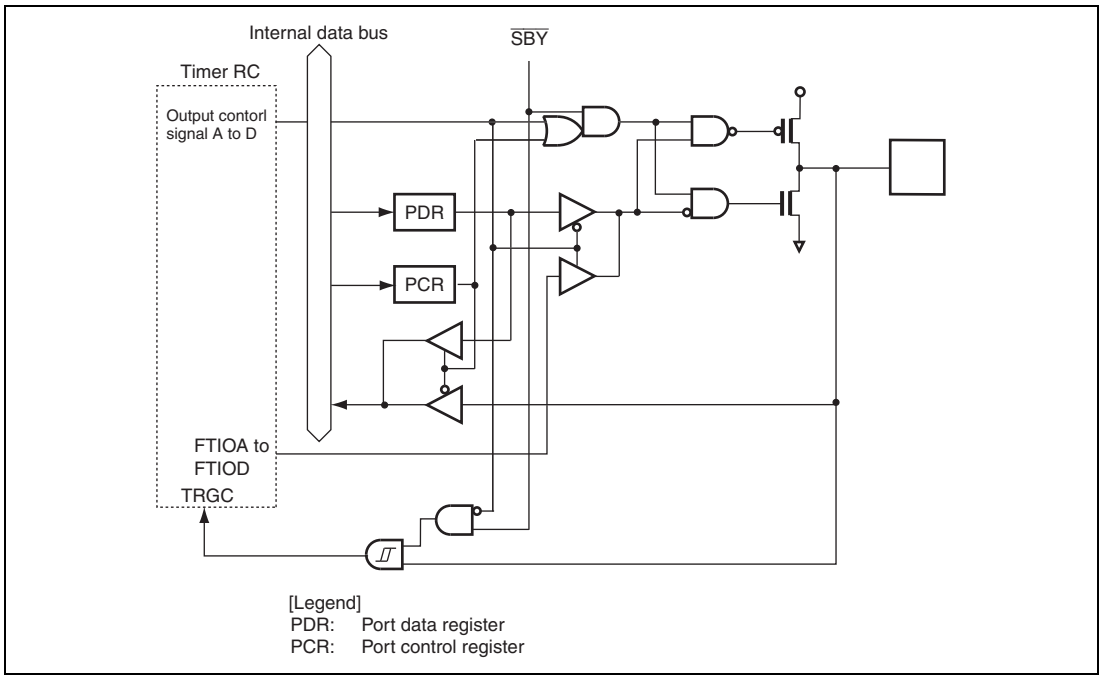



**Figure B.28 Port H Block Diagram (PH7, PH6, PH5, PH4)**

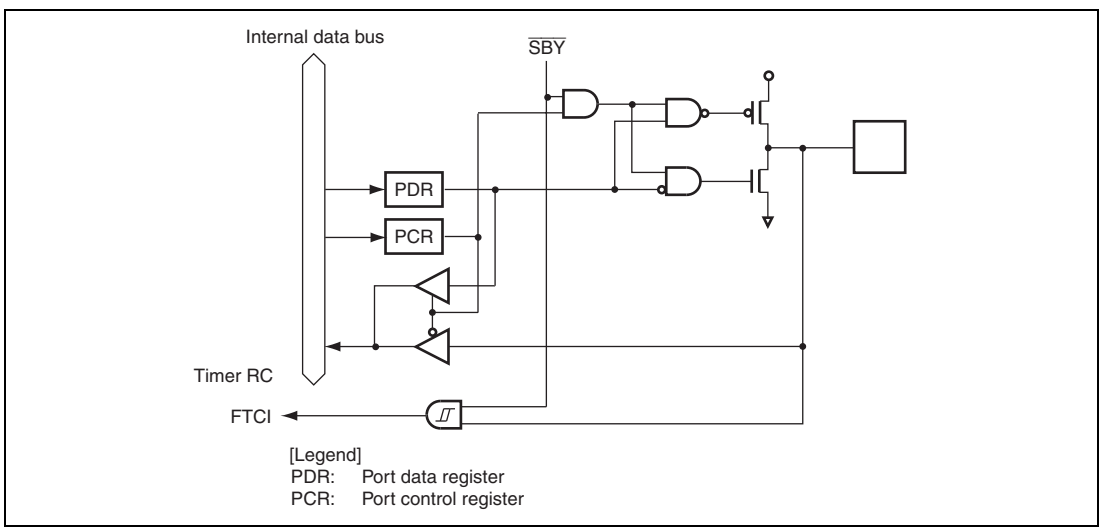



**Figure B.29 Port H Block Diagram (PH3)**

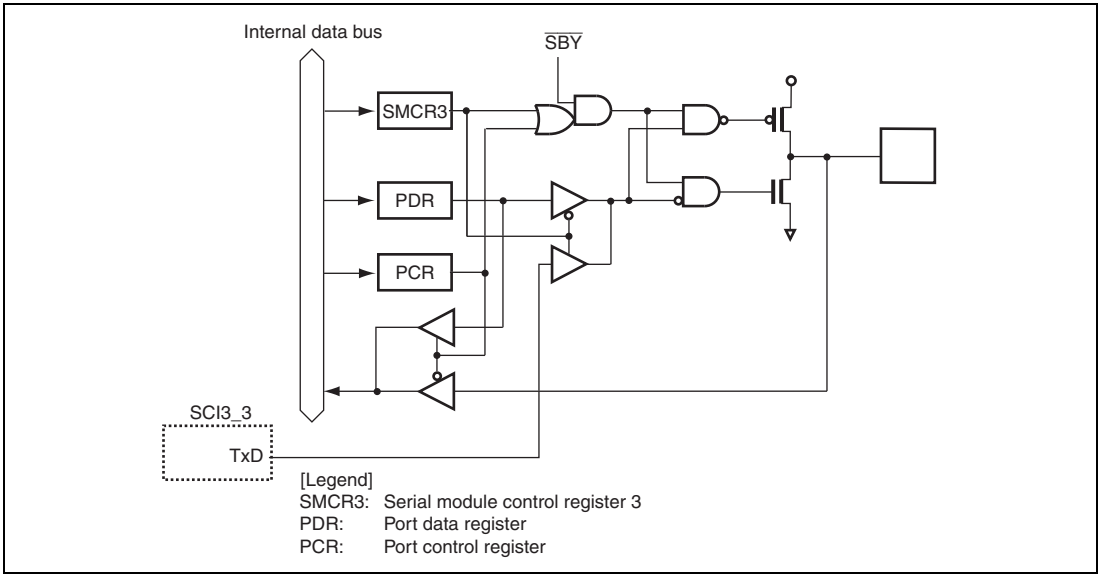



**Figure B.30 Port H Block Diagram (PH2)**

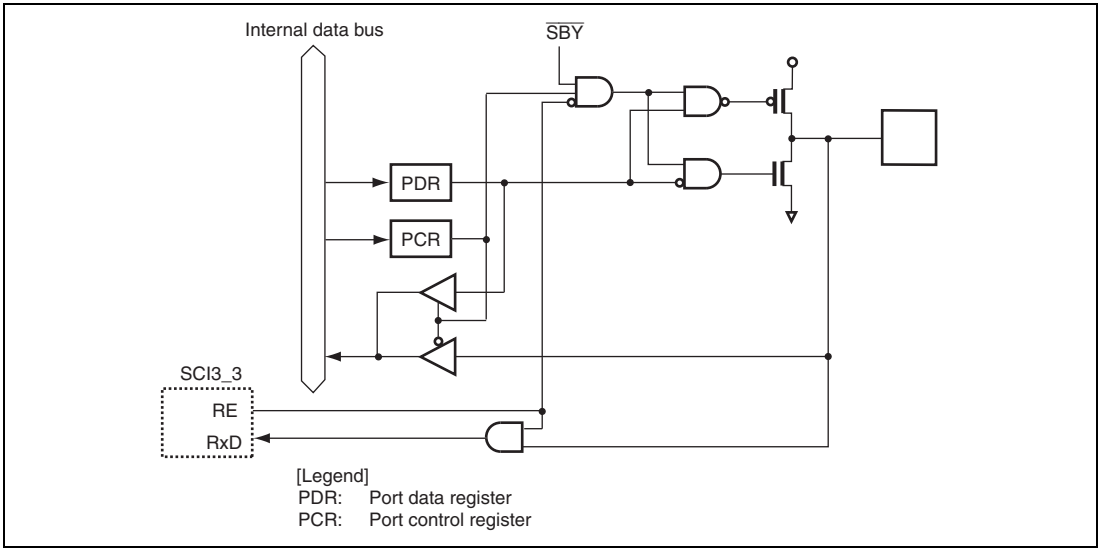



**Figure B.31 Port H Block Diagram (PH1)**

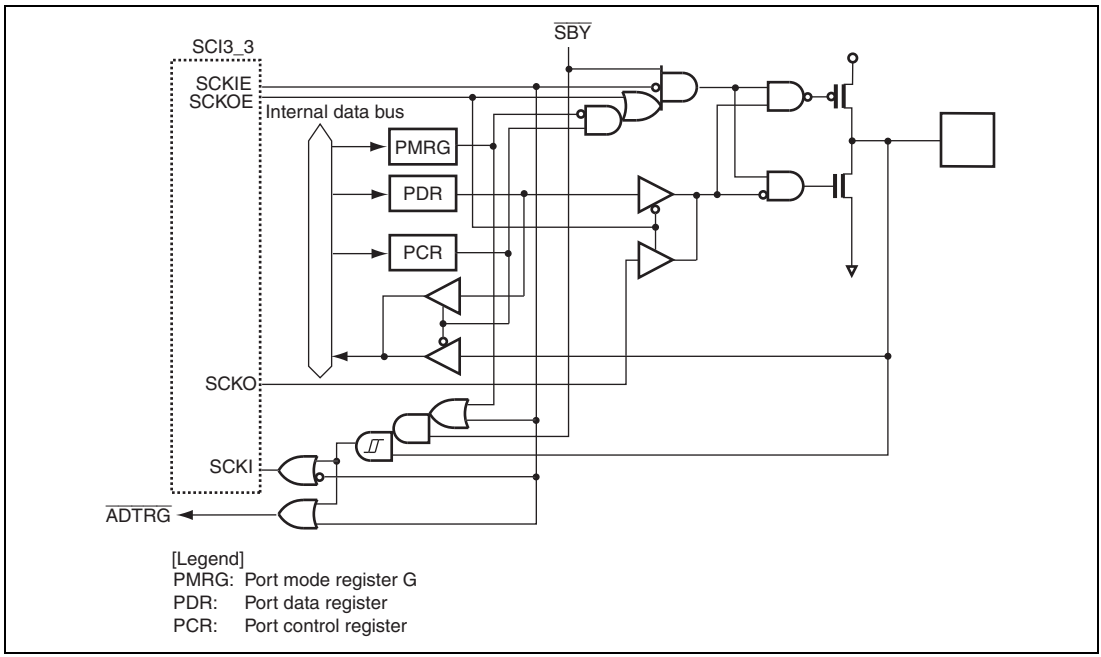



**Figure B.32 Port H Block Diagram (PH0)**

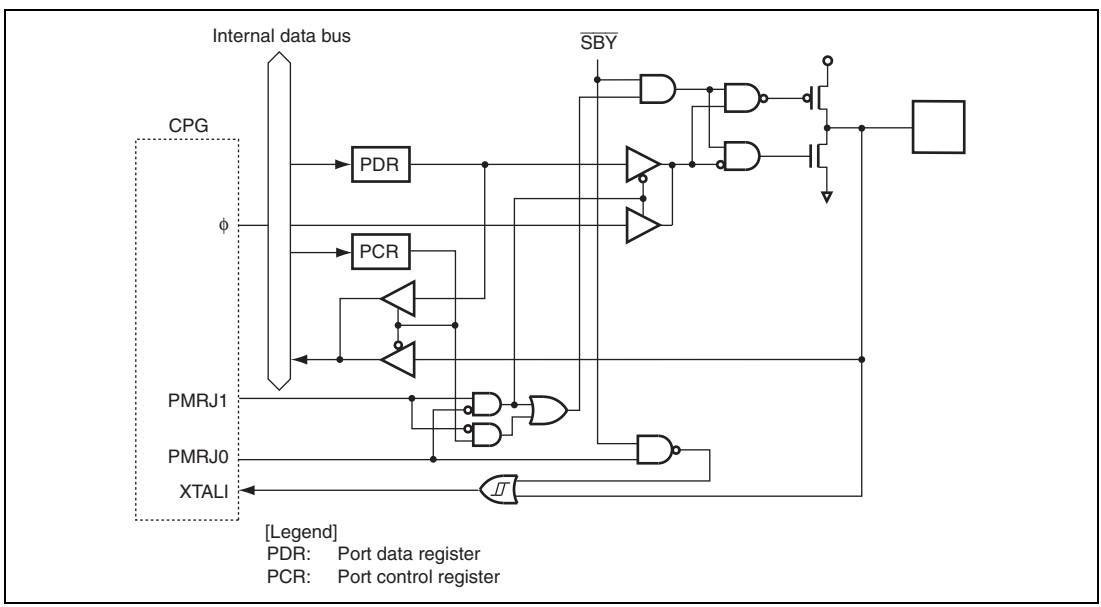



**Figure B.33 Port J Block Diagram (PJ1)**

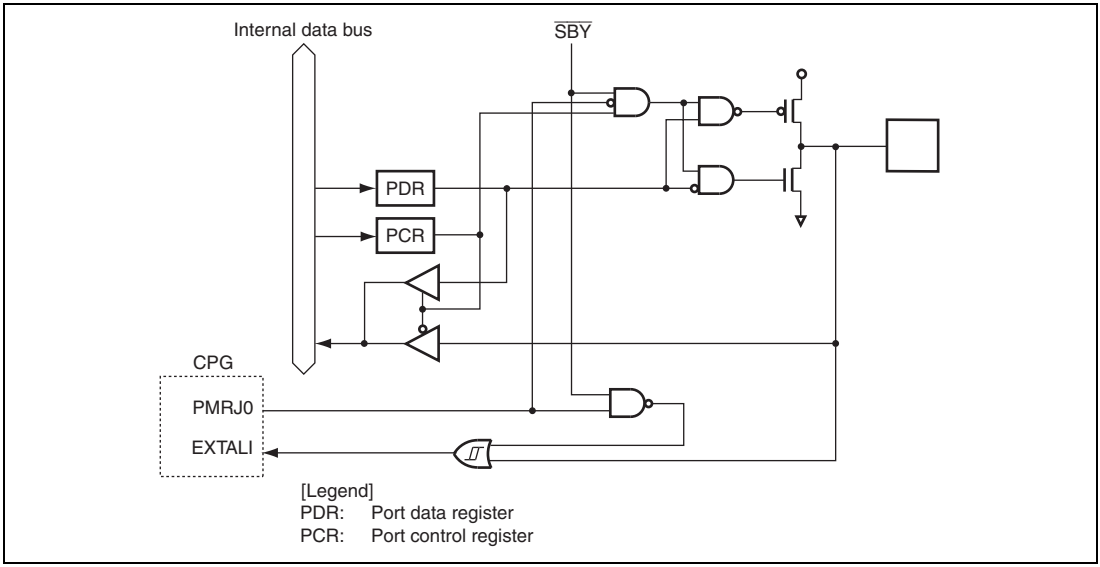



**Figure B.34 Port J Block Diagram (PJ0)**

## B.2 Port States in Each Operating Mode

| Port | Reset | Sleep | Subsleep | Standby | Subactive | Active |
|---|---|---|---|---|---|---|
| P17 to P14, P12 to P10 | High impedance | Retained | Retained | High impedance* | Functioning | Functioning |
| P27 to P20 | High impedance | Retained | Retained | High impedance | Functioning | Functioning |
| P37 to P30 | High impedance | Retained | Retained | High impedance | Functioning | Functioning |
| P57 to P50 | High impedance | Retained | Retained | High impedance* | Functioning | Functioning |
| P77 to P74, P72 to P70 | High impedance | Retained | Retained | High impedance | Functioning | Functioning |
| P87 to P85 | High impedance | Retained | Retained | High impedance | Functioning | Functioning |
| PC3 to PC0 | High impedance | Retained | Retained | High impedance | Functioning | Functioning |
| PD7 to PD0 | High impedance | Retained | Retained | High impedance | Functioning | Functioning |
| PE7 to PE0 | High impedance | Retained | Retained | High impedance | Functioning | Functioning |
| PF7 to PF0 | High impedance | High impedance | High impedance | High impedance | High impedance | High impedance |
| PG7 to PG0 | High impedance | Retained | Retained | High impedance | Functioning | Functioning |
| PH7 to PH0 | High impedance | Retained | Retained | High impedance | Functioning | Functioning |

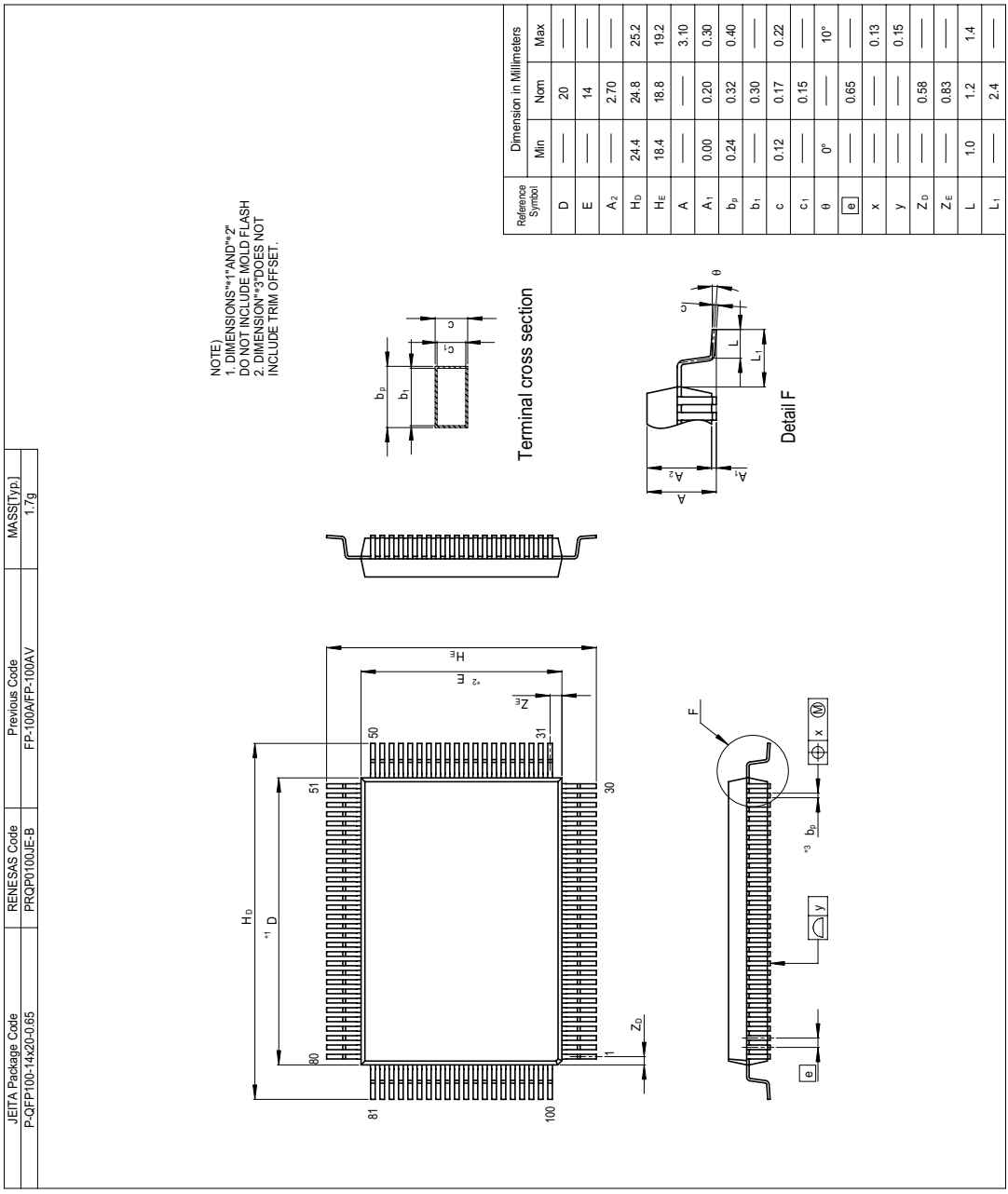
Note: * High level output when the pull-up MOS is in on state.

# C. Product Code Lineup

| Product Classification | | | Product Code | Model Marking | Package (Code) |
|---|---|---|---|---|---|
| H8/36109 | Flash memory version | Standard product | HD64F36109F | HD64F36109F | QFP-100 (FP-100A) |
| | | | HD64F36109H | HD64F36109H | LQFP-100 (FP-100U) |
| | | Product with POR & LVDC | HD64F36109GF | HD64F36109GF | QFP-100 (FP-100A) |
| | | | HD64F36109GH | HD64F36109GH | LQFP-100 (FP-100U) |

# D. Package Dimensions

The package dimensions that are shown in the Renesas Semiconductor Packages Data Book have priority.




| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | — | 20 | — |
| E | — | 14 | — |
| $A_2$ | — | 2.70 | — |
| $H_D$ | 24.4 | 24.8 | 25.2 |
| $H_E$ | 18.4 | 18.8 | 19.2 |
| A | — | — | 3.10 |
| $A_1$ | 0.00 | 0.20 | 0.30 |
| $b_p$ | 0.24 | 0.32 | 0.40 |
| $b_1$ | — | 0.30 | — |
| c | 0.12 | 0.17 | 0.22 |
| $c_1$ | — | 0.15 | — |
| θ | 0° | — | 10° |
| e | — | 0.65 | — |
| x | — | — | 0.13 |
| y | — | — | 0.15 |
| $Z_D$ | — | 0.58 | — |
| $Z_E$ | — | 0.83 | — |
| L | 1.0 | 1.2 | 1.4 |
| $L_1$ | — | 2.4 | — |

**Figure D.1 FP-100A Package Dimensions**

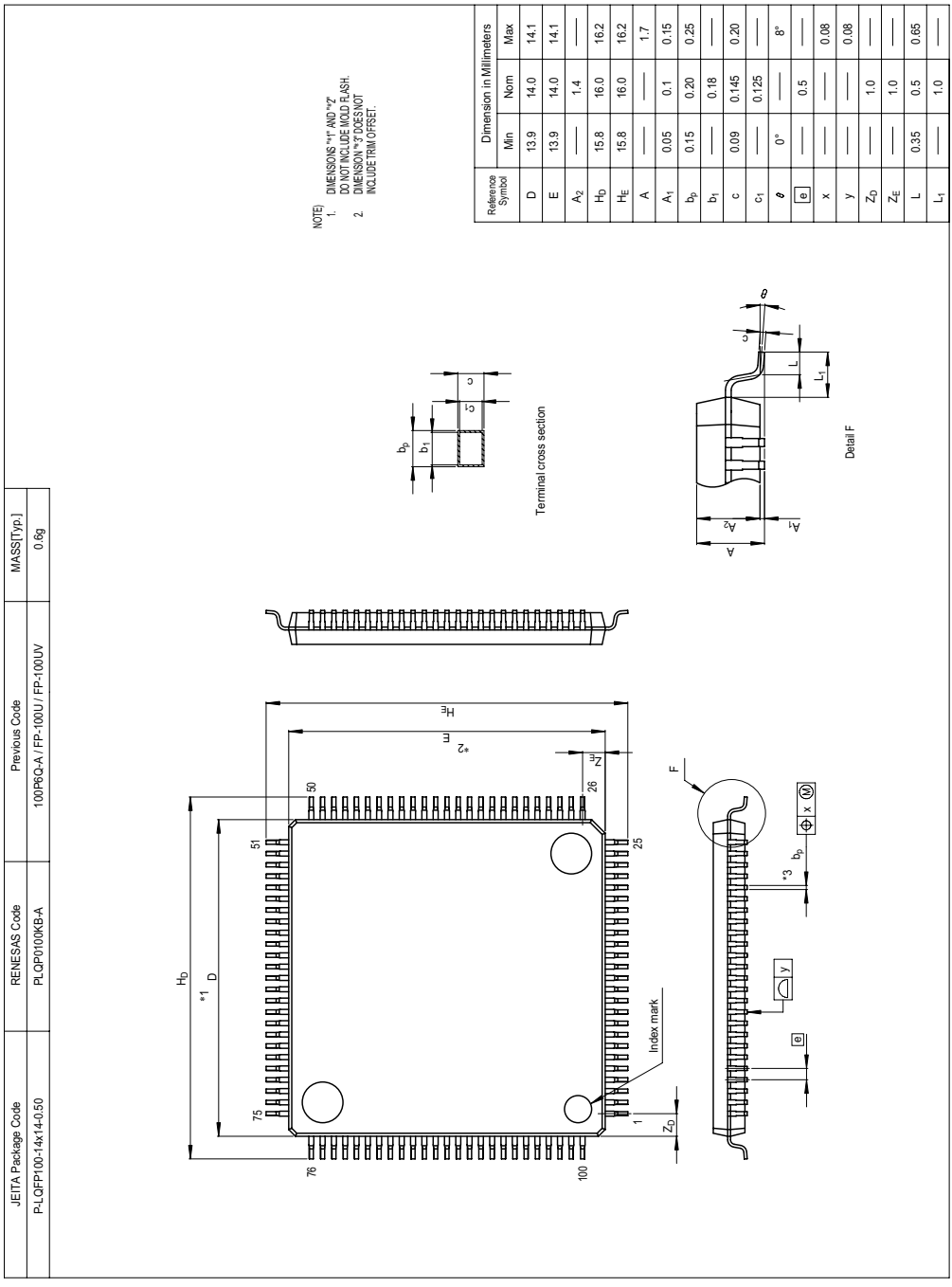



**Figure D.2 FP-100U Package Dimensions**

# Main Revisions for this Edition

**Item** | **Page** | **Revision (See Manual for Details)**

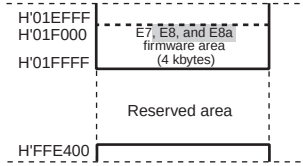
---

2.1 Address Space and Memory Map

Figure 2.1 Memory Map

12

Figure amended




---

5.2.4 Clock Control/Status Register (CKCSR)

76

Table amended

| Bit | Bit Name | Initial Value | R/W | Description | | | |
|---|---|---|---|---|---|---|---|
| 7 | PMRJ1 | 0 | R/W | OSC Pin Function Select 1 and 0 | | | |
| 6 | PMRJ0 | 0 | R/W | PMRJ1 | PMRJ0 | OSC2 | OSC1 |
| | | | | 0 | 0 | I/O | I/O |
| | | | | 1 | 0 | CLKOUT | I/O |
| | | | | 0 | 1 | (open) | OSC1 (external clock input) |
| | | | | 1 | 1 | OSC2 | OSC1 |

---

6.1.3 System Control Register 3 (SYSCR3)

94

Table amended

| Bit | Bit Name | Initial Value | R/W | Description | | | |
|---|---|---|---|---|---|---|---|
| 7 | PMRJ1 | 0 | R/W | OSC Pin Function Select 1 and 0 | | | |
| 6 | PMRJ0 | 0 | R/W | PMRJ1 | PMRJ0 | OSC2 | OSC1 |
| | | | | 0 | 0 | I/O | I/O |
| | | | | 1 | 0 | CLKOUT | I/O |
| | | | | 0 | 1 | (open) | OSC1 (external clock input) |
| | | | | 1 | 1 | OSC2 | OSC1 |

---

9.4.1 Port Mode Register 5 (PMR5)

139

Table amended

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | POF57 | 0 | R/W | When the bit is set to 1, the corresponding pin functions as the NMOS open-drain output. When cleared to 0, the pin functions as the NMOS push-pull output. |
| 6 | POF56 | 0 | R/W | |

---

9.13.3 Pin Functions

- PJ1/OSC2/CLKOUT pin

185

Table amended

| Register | CKCSR | | PCRJ | |
|---|---|---|---|---|
| Bit Name | PMRJ1 | PMRJ0 | PCRJ1 | Pin Function |
| Setting Value | 0 | 0 | 0 | PJ1 input pin |
| | | | 1 | PJ1 output pin |
| | | 1 | X | Open |

---

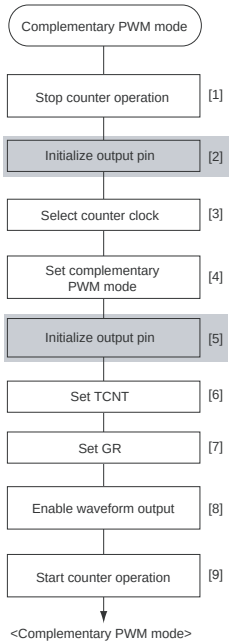
- PJ0/OSC1 pin

185

Table amended

| Register | CKCSR | | PCRJ | |
|---|---|---|---|---|
| Bit Name | PMRJ1 | PMRJ0 | PCRJ0 | Pin Function |
| Setting Value | X | 0 | 0 | PJ0 input pin |
| | | | 1 | PJ0 output pin |
| | 0 | 1 | X | OSC1 (clock input) |
| | 1 | 1 | X | OSC1 oscillation pin |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 9.13.3 Pin Functions<br>• PJ0/OSC1 pin | 185 | Table Note added<br>Note: For details on external clock input, also see 5.2.4, Clock Control/Status Register. |
| 14.4.7 Complementary PWM Mode<br>Figure 14.32 Example of Complementary PWM Mode Setting Procedure | 327 | Figure amended |




[1] Clear bits STR0 and STR1 in TSTR to 0, and stop the counter operation of TCNT_0. Stop TCNT_0 and TCNT_1 and set complementary PWM mode.
[2] Write H'00 to TRDOCR.
[3] Use bits TPSC2 to TPSC0 in TCR to select the same counter clock for channels 0 and 1. When an external clock is selected, select the edge of the external clock by bits CKEG1 and CKEG0 in TCR. Do not use bits CCLR1 and CCLR0 in TCR to clear the counter.
[4] Use bits CMD1 and CMD0 in TFCR to set complementary PWM mode. FTIOB0 to FTIOD0 and FTIOA1 to FTIOD1 automatically become PWM output pins.
[5] Set H'00 to TRDOCR.
[6] TCNT_1 must be H'0000. Set a non-overlapped period to TCNT_0.
[7] GRA_0 is a cycle register. Set the cycle to GRA_0. Set the timing to change the PWM output waveform to GRB_0, GRA_1, and GRB_1. Note that the timing must be set within the range of compare match carried out for TCNT_0 and TCNT_1. For GR settings, see 3. Setting GR Value in Complementary PWM Mode in section 13.4.7.
[8] Use TOER to enable or disable the timer output.
[9] Set the STR0 and STR1 bits in TSTR to 1 to start the count operation.

Note: To re-enter complementary PWM mode, first, enter a mode other than the complementary PWM mode. After that, repeat the setting procedures from step [1].
For settings of waveform outputs with a duty cycle of 0% and 100%, see the settings shown in 2. Examples of Complementary PWM Mode Operation and 3. Setting GR Value in Complementary PWM Mode in section 14.4.7.

| 17.3.8 Bit Rate Register (BRR)<br>Table 17.3 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2) | 381 | Table amended |
|---|---|---|

| | Operating Frequency ϕ (MHz) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | | | 12 | | | 12.288 | | | 14 | | |
| Bit Rate (bit/s) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) | n | N | Error (%) |

| Item | Page | Revision (See Manual for Details) |
|---|---|---|
| 17.8.2 Mark State and Break Sending | 404 | Description amended.<br>When the TXD or TXD2 bit in PMR1 or the TXD_3 bit in SMCR is 0, the TXD pin is used as an I/O port whose direction (input or output) and level are determined by PCR and PDR. This can be used to set the TXD pin to mark state (high level) or send a break during serial data transmission. To maintain the communication line at mark state until TE is set to 1, set both PCR and PDR to 1 and also cleared the TXD bit to 0. Then, the TXD pin becomes an I/O port, and 1 is output from the TXD pin. |
| 18.7 Usage Notes | 442 | Newly added. |
| 22.2 Register Bits | 490 | Table amended |

| Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Module Name |
|---|---|---|---|---|---|---|---|---|---|
| SYSCR3 | — | — | — | — | — | — | — | STS3 | Power-down modes |

# Index


## Numerics

14-bit PWM ..... 365

## A

A/D converter ..... 445
Absolute address ..... 32
Acknowledge ..... 423
Address break ..... 65
Addressing modes ..... 31
Arithmetic operations instructions ..... 22
Asynchronous mode ..... 383
Auto-reload timer operation ..... 202

## B

Band-gap regulator ..... 461
Bit manipulation instructions ..... 25
Bit rate ..... 379
Bit synchronous circuit ..... 441
Block data transfer instructions ..... 29
Boot mode ..... 111
Boot program ..... 111
Branch instructions ..... 27
Break ..... 404
Buffer operation ..... 336

## C

Clock pulse generators ..... 71
Clock synchronous mode ..... 391
Clocked synchronous serial format ..... 432
Combinations of instructions and addressing modes ..... 552
Complementary PWM mode ..... 326
Condition field ..... 30
Condition-code register (CCR) ..... 15
CPU ..... 11

## D

Data reading procedure ..... 196
Data transfer instructions ..... 21
Digital filter includes ..... 253

## E

Effective address ..... 34
Effective address extension ..... 30
Erase/erase-verify ..... 118
Erasing units ..... 106
Error protection ..... 120
Event counter operation ..... 202
Exception handling ..... 45
External oscillators ..... 84

## F

Flash memory ..... 105
Framing error ..... 387

## G

General registers ..... 14

## H

Hardware protection ..... 120

## I

I/O port block diagrams ..... 553
I/O ports ..... 125

I²C Bus format ..... 423
I²C bus interface 2 (IIC2) ..... 407
Immediate ..... 33
Initial setting procedure ..... 195
Input capture function ..... 312
Instruction list ..... 523
Instruction set ..... 20
Internal interrupts ..... 58
Internal power supply step-down circuit ..... 471
Interrupt mask bit (I) ..... 15
Interrupt response time ..... 62
Interval timer operation ..... 201
IRQ3 to IRQ0 interrupts ..... 56

## L

Large current ports ..... 2
Logic operations instructions ..... 24
Low-voltage detection circuit ..... 461
LVDI (interrupt by low voltage detect) circuit ..... 469
LVDR (reset by low voltage detect) circuit ..... 468

## M

Mark state ..... 404
Memory indirect ..... 33
Memory map ..... 12
Module standby function ..... 103
Multiprocessor communication function 397

## N

NMI interrupt ..... 56
Noise canceler ..... 435
Noise canceller ..... 371
Number of execution states ..... 541

## O

On-board programming modes ..... 111
On-chip oscillator ..... 77
Operation field ..... 30
Overrun error ..... 387

## P

Package ..... 2
Package dimensions ..... 572
Parity error ..... 387
Pin arrangement ..... 4
Power-down modes ..... 89
Power-down states ..... 121
Power-on reset ..... 461
Power-on reset circuit ..... 467
Prescaler S ..... 87
Prescaler W ..... 87
Product code lineup ..... 572
Program counter (PC) ..... 15
Program/program-verify ..... 115
Program-counter relative ..... 33
Programmer mode ..... 121
Programming units ..... 106
Programming/erasing in user programming mode ..... 114
PWM mode ..... 243, 316
PWM2 mode ..... 247

## R

Realtime clock (RTC) ..... 187
Register addresses ..... 474
Register bits ..... 483
Register direct ..... 31
Register field ..... 30
Register indirect ..... 31
Register indirect with displacement ..... 32
Register indirect with post-increment ..... 32
Register indirect with pre-decrement ..... 32

Register states in each operating mode ... 491
Registers
ABRKCR ... 66, 481, 489, 496
ABRKSR ... 68, 481, 489, 496
ADCR ... 451, 475, 485, 492
ADCSR ... 449, 475, 485, 492
ADDR ... 448, 475, 485, 492
BARE ... 68, 481, 489, 496
BARH ... 68, 481, 489, 496
BARL ... 68, 481, 489, 496
BDRH ... 68, 481, 489, 496
BDRL ... 68, 481, 489, 496
BRR ... 379, 481, 488, 496
CKCSR ... 76, 479, 487, 495
EBR1 ... 109, 480, 488, 496
FENR ... 110, 480, 488, 496
FLMCR1 ... 107, 480, 488, 496
FLMCR2 ... 108, 480, 488, 496
FLPWCR ... 110, 480, 488, 496
GRA ... 235, 286, 474, 475, 483, 484, 491
GRB ... 235, 286, 474, 475, 483, 484, 491
GRC ... 235, 286, 474, 475, 483, 484, 491
GRD ... 235, 286, 474, 475, 483, 484, 491
ICCR1 ... 410, 480, 488, 495
ICCR2 ... 413, 480, 488, 495
ICDRR ... 422, 480, 488, 495
ICDRS ... 422
ICDRT ... 422, 480, 488, 495
ICIER ... 417, 480, 488, 495
ICMR ... 415, 480, 488, 495
ICR ... 55, 479, 487, 495
ICSR ... 419, 480, 488, 495
IEGR1 ... 48, 482, 490, 497
IEGR2 ... 49, 482, 490, 497
IENR1 ... 50, 482, 490, 497
IENR2 ... 51, 482, 490, 497
IRR1 ... 51, 482, 490, 497
IRR2 ... 53, 482, 490, 497
IWPR ... 53, 482, 490, 497
LVDCR ... 464, 479, 487, 495
LVDSR ... 466, 479, 487, 495
MSTCR1 ... 94, 482, 490, 498
MSTCR2 ... 95, 482, 490, 498
MSTCR4 ... 96, 476, 485, 492
PCR1 ... 127, 482, 489, 497
PCR2 ... 131, 482, 489, 497
PCR3 ... 135, 482, 489, 497
PCR5 ... 140, 482, 490, 497
PCR7 ... 144, 482, 490, 497
PCR8 ... 147, 482, 490, 497
PCRC ... 149, 482, 490, 497
PCRD ... 152, 476, 485, 492
PCRE ... 161, 476, 485, 492
PCRG ... 174, 476, 485, 492
PCRH ... 179, 476, 485, 492
PCRJ ... 184, 476, 485, 492
PDR1 ... 127, 481, 489, 497
PDR2 ... 132, 481, 489, 497
PDR3 ... 136, 481, 489, 497
PDR5 ... 140, 481, 489, 497
PDR7 ... 145, 481, 489, 497
PDR8 ... 148, 481, 489, 497
PDRC ... 150, 481, 489, 497
PDRD ... 152, 475, 485, 492
PDRE ... 161, 475, 485, 492
PDRF ... 170, 475, 485, 492
PDRG ... 174, 476, 485, 492
PDRH ... 180, 476, 485, 492
PDRJ ... 185, 476, 485, 492
PMR1 ... 126, 481, 489, 497
PMR3 ... 132, 482, 489, 497
PMR5 ... 139, 481, 489, 497
PMRF ... 171, 476, 485, 492
PMRG ... 175, 476, 485, 492
POCR ... 297, 476, 485, 493
PUCR1 ... 128, 481, 489, 497
PUCR5 ... 141, 481, 489, 497
PWCR ... 366, 481, 489, 496
PWDRL ... 367, 481, 489, 496
PWDRU ... 367, 481, 489, 496

RCCR .....73, 479, 487, 495
RCTRMDPR .....74, 479, 487, 495
RCTRMDR.....75, 479, 487, 495
RDR.....373, 481, 489, 496
RHRDR .....190, 479, 487, 494
RMINDR .....189, 479, 487, 494
RSECDR.....188, 479, 487, 494
RSR ..... 373
RTCCR1 .....192, 479, 487, 494
RTCCR2 .....193, 479, 487, 494
RTCCSR.....194, 479, 487, 494
RWKDR .....191, 479, 487, 494
SAR .....421, 480, 488, 495
SCR3.....375, 481, 488, 496
SMCR.....371, 475, 484, 492
SMR.....374, 481, 488, 496
SSR.....377, 481, 489, 496
SYSCR1 .....90, 482, 490, 497
SYSCR2 .....93, 482, 490, 497
SYSCR3 .....94, 482, 490, 497
TCB1 .....201, 480, 488, 495
TCNTV.....205, 480, 488, 496
TCORA.....205, 480, 488, 496
TCORB.....205, 480, 488, 496
TCRV0 .....206, 480, 488, 496
TCRV1 .....209, 481, 488, 496
TCSRV .....207, 480, 488, 496
TCSRWD .....360, 481, 489, 496
TCWD .....362, 481, 489, 496
TDR.....374, 481, 489, 496
TLB1.....201, 480, 488, 495
TMB1 .....200, 480, 488, 495
TMWD .....362, 481, 489, 496
TRCCNT .....235, 475, 484, 491
TRCCR1.....224, 479, 487, 494
TRCCR2.....225, 479, 487, 494
TRCDF .....234, 479, 487, 494
TRCIER.....226, 479, 487, 494
TRCIOR0 .....229, 479, 487, 494
TRCIOR1 .....231, 479, 487, 494
TRCMR ..... 222, 479, 487, 494
TRCOER..... 233, 479, 487, 494
TRCSR..... 227, 479, 487, 494
TRDCNT ..... 286, 474, 483, 491
TRDCR ..... 287, 476, 485, 493
TRDDF ..... 298, 476, 485, 493
TRDFCR..... 280, 477, 486, 493
TRDIER..... 296, 476, 485, 493
TRDIORA..... 289, 476, 485, 493
TRDIORC..... 289, 476, 485, 493
TRDMDR ..... 278, 477, 486, 493
TRDOCR ..... 284, 477, 486, 493
TRDOER1 ..... 282, 477, 486, 493
TRDOER2 ..... 284, 477, 486, 493
TRDPMR..... 279, 477, 486, 493
TRDSR ..... 293, 476, 485, 493
TRDSTR..... 274, 477, 486, 493
TSR.....373
Reset exception handling..... 56
Reset synchronous PWM mode..... 322

## S

Sample-and-hold circuit..... 453
Scan mode..... 452
Serial communication interface 3 (SCI3) ..... 369
Shift instructions..... 24
Single mode ..... 452
Slave address..... 423
Sleep mode..... 100
Software protection..... 120
Stack pointer (SP) ..... 14
Stack status ..... 58
Standby mode ..... 100
Start condition..... 423
Stop condition..... 423
Subactive mode..... 101
Subclock generator ..... 86
Subsleep mode ..... 101

Synchronous operation ..... 315
System control instructions..... 28

## T

Timer B1..... 199
Timer RC..... 217
Timer RD..... 265
Timer V ..... 203
Transfer rate..... 412
Trap instruction..... 45
Trimming ..... 82

## W

Watchdog timer..... 359
Waveform output ..... 368
Waveform output by compare match..... 309
WKP5 to WKP0 interrupts ..... 57

**Renesas 16-Bit Single-Chip Microcomputer**
**H8/36109 Group**
**User's Manual: Hardware**

Publication Date: Rev.1.00, January 25, 2006
Rev.2.00, September 12, 2011
Published by: Renesas Electronics Corporation

**SALES OFFICES** Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.


**Renesas Electronics America Inc.**
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

**Renesas Electronics Canada Limited**
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

**Renesas Electronics Europe Limited**
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

**Renesas Electronics Europe GmbH**
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

**Renesas Electronics (China) Co., Ltd.**
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

**Renesas Electronics (Shanghai) Co., Ltd.**
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

**Renesas Electronics Hong Kong Limited**
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

**Renesas Electronics Taiwan Co., Ltd.**
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

**Renesas Electronics Singapore Pte. Ltd.**
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

**Renesas Electronics Malaysia Sdn.Bhd.**
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

**Renesas Electronics Korea Co., Ltd.**
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141



© 2011 Renesas Electronics Corporation and Renesas Solutions Corp. All rights reserved.

Colophon 1.1

# H8/36109 Group
# User's Manual: Hardware

Renesas Electronics Corporation

R01UH0294EJ0200
(Previous Number: REJ09B0240-0150)

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Renesas Electronics:

DF36109HV