User's Manual

16

# RL78/ F13, F14

## User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.2.10 Dec 2015

# Notice

1. 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.
2. 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.
3. 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.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “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 etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). 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 for which it is not intended. 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 which the product is not intended by Renesas Electronics.
6. 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.
7. 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 systems manufactured by you.
8. 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.
9. 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. You should not use Renesas Electronics products or 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. When exporting the Renesas Electronics 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.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
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.

(2012.4)

NOTES FOR CMOS DEVICES

(1) VOLTAGE APPLICATION WAVEFORM AT INPUT PIN: Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).

(2) HANDLING OF UNUSED INPUT PINS: Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

(3) PRECAUTION AGAINST ESD: A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

(4) STATUS BEFORE INITIALIZATION: Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

(5) POWER ON/OFF SEQUENCE: In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

(6) INPUT OF SIGNAL DURING POWER OFF STATE : Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

# How to Use This Manual

**Readers**

This manual is intended for user engineers who wish to understand the functions of the RL78/F13, F14 and design and develop application systems and programs for these devices.

**Purpose**

This manual is intended to give users an understanding of the functions described in the **Organization** below.

**Organization**

The RL78/F13, F14 manual is separated into two parts: this manual and the software edition (common to the RL78 family).

| **RL78/F13, F14 User's Manual Hardware** | **RL78 family User's Manual Software** |
|---|---|
| • Pin functions<br>• Internal block functions<br>• Interrupts<br>• Other on-chip peripheral functions<br>• Electrical specifications | • CPU functions<br>• Instruction set<br>• Explanation of each instruction |

**How to Read This Manual**

It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, and microcontrollers.

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler.
- To know details of the RL78/F13, F14 Microcontroller instructions:
  → Refer to the separate document **RL78 Family User's Manual: Software (R01US0015E)**.

| | | | |
|---|---|---|---|
| **Conventions** | Data significance: | Higher digits on the left and lower digits on the right | |
| | Active low representations: | $\overline{\times\times\times}$ (overscore over pin and signal name) | |
| | **Note**: | Footnote for item marked with **Note** in the text | |
| | **Caution**: | Information requiring particular attention | |
| | **Remark**: | Supplementary information | |
| | Numerical representations: | Binary | ···×××× or ××××B |
| | | Decimal | ···×××× |
| | | Hexadecimal | ···××××H |

**Related Documents** The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

**Documents Related to Devices**

| Document Name | Document No. |
|---|---|
| RL78/F13, F14 User's Manual: Hardware | R01UH0368E |
| RL78 Family User's Manual: Software | R01US0015E |

**Documents Related to Flash Memory Programming**

| Document Name | Document No. |
|---|---|
| PG-FP5 Flash Memory Programmer User's Manual | R20UT0008E |

**Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.**

**Other Documents**

| Document Name | Document No. |
|---|---|
| Renesas MPUs & MCUs RL78 Family | R01CP0003E |
| Semiconductor Package Mount Manual | **Note** |
| Quality Grades on NEC Semiconductor Devices | C11531E |
| Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD) | C11892E |
| Semiconductor Reliability Handbook | R51ZZ0001E |

**Note** See the “Semiconductor Package Mount Manual” website (http://www.renesas.com/products/package/manual/index.jsp).

**Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.**

All trademarks and registered trademarks are the property of their respective owners.
EEPROM is a trademark of Renesas Electronics Corporation.
Windows, Windows NT and Windows XP are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
PC/AT is a trademark of International Business Machines Corporation.
SuperFlash is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

**CONTENTS**


**CHAPTER 1 OVERVIEW ..... 1**

**1.1 Features ..... 1**

1.1.1 Applications ..... 2

**1.2 Product Lineup ..... 3**

**1.3 Function Overview ..... 4**

1.3.1 RL78/F14 Functions List ..... 4

1.3.2 RL78/F13 (CAN and LIN incorporated) Functions List ..... 6

1.3.3 RL78/F13 (LIN incorporated) Functions List ..... 8

**1.4 Block Diagram ..... 10**

1.4.1 RL78/F14: Block Diagram of R5F10PPn (n = E, F, G, H, J) 100-pin Products ..... 10

1.4.2 RL78/F14: Block Diagram of R5F10PMn (n = G, H, J) 80-pin Products ..... 11

1.4.3 RL78/F14: Block Diagram of R5F10PLn (n = G, H, J) 64-pin Products ..... 12

1.4.4 RL78/F14: Block Diagram of R5F10PGn (n = G, H, J) 48-pin Products ..... 13

1.4.5 RL78/F14: Block Diagram of R5F10PMn (n = E, F) 80-pin Products ..... 14

1.4.6 RL78/F14: Block Diagram of R5F10PLn (n = E, F) 64-pin Products ..... 15

1.4.7 RL78/F14: Block Diagram of R5F10PGn (n = D, E, F) 48-pin Products ..... 16

1.4.8 RL78/F14: Block Diagram of R5F10PBn (n = D, E) 32-pin Products ..... 17

1.4.9 RL78/F14: Block Diagram of R5F10PAn (n = D, E) 30-pin Products ..... 18

1.4.10 RL78/F13: Block Diagram of R5F10BMn (n = E, F, G) (CAN and LIN incorporated) 80-pin Products ..... 19

1.4.11 RL78/F13: Block Diagram of R5F10BLn (n = C, D, E, F, G) (CAN and LIN incorporated) 64-pin Products ..... 20

1.4.12 RL78/F13: Block Diagram of R5F10BGn (n = C, D, E, F, G) (CAN and LIN incorporated) 48-pin Products ..... 21

1.4.13 RL78/F13: Block Diagram of R5F10BBn (n = C, D, E, F, G) (CAN and LIN incorporated) 32-pin Products ..... 22

1.4.14 RL78/F13: Block Diagram of R5F10BAn (n = C, D, E, F, G) (CAN and LIN incorporated) 30-pin Products ..... 23

1.4.15 RL78/F13: Block Diagram of R5F10AMn (n = E, F, G) (LIN incorporated) 80-pin Products ..... 24

1.4.16 RL78/F13: Block Diagram of R5F10ALn (n = F, G) (LIN incorporated) 64-pin Products ..... 25

1.4.17 RL78/F13: Block Diagram of R5F10AGn (n = F, G) (LIN incorporated) 48-pin Products ..... 26

1.4.18 RL78/F13: Block Diagram of R5F10ALn (n = C, D, E) (LIN incorporated) 64-pin Products ..... 27

1.4.19 RL78/F13: Block Diagram of R5F10AGn (n = A, C, D, E) (LIN incorporated) 48-pin Products ..... 28

1.4.20 RL78/F13: Block Diagram of R5F10ABn (n = A, C, D, E) (LIN incorporated) 32-pin Products ..... 29

1.4.21 RL78/F13: Block Diagram of R5F10AAn (n = A, C, D, E) (LIN incorporated) 30-pin Products ..... 30

1.4.22 RL78/F13: Block Diagram of R5F10A6n (n = A, C, D, E) (LIN incorporated) 20-pin Products ..... 31

**1.5 Pin Configurations ..... 32**
1.5.1 RL78/F14 Pin Configuration for 100-pin Products ..... 32
1.5.2 RL78/F14 Pin Configuration for 80-pin Products ..... 33
1.5.3 RL78/F13 Pin Configuration for 80-pin Products ..... 34
1.5.4 RL78/F14 Pin Configuration for 64-pin Products ..... 35
1.5.5 RL78/F13 Pin Configuration for 64-pin Product ..... 36
1.5.6 RL78/F14 Pin Configuration for 48-pin Products ..... 37
1.5.7 RL78/F13 Pin Configuration for 48-pin Products ..... 38
1.5.8 RL78/F14 Pin Configuration for 32-pin Products ..... 39
1.5.9 RL78/F13 Pin Configuration for 32-pin Products ..... 40
1.5.10 RL78/F14 Pin Configuration for 30-pin Products ..... 41
1.5.11 RL78/F13 Pin Configuration for 30-pin Products ..... 42
1.5.12 RL78/F13 Pin Configuration for 20-pin Products ..... 43
**1.6 Order Information ..... 44**

**CHAPTER 2 PIN FUNCTIONS ..... 47**

**2.1 Pin Function List ..... 47**
2.1.1 RL78/F14 100-pin products ..... 49
2.1.2 RL78/F13 (CAN and LIN incorporated) 80-pin products ..... 51
2.1.3 RL78/F13 (LIN incorporated) 80-pin products ..... 53
2.1.4 Pins for each product (pins other than port pins) ..... 55
**2.2 Description of Pin Functions ..... 68**
2.2.1 P00 to P03 (Port 0) ..... 68
2.2.2 P10 to P17 (Port 1) ..... 68
2.2.3 P30 to P34 (Port 3) ..... 70
2.2.4 P40 to P47 (Port 4) ..... 71
2.2.5 P50 to P57 (Port 5) ..... 72
2.2.6 P60 to P67 (Port 6) ..... 73
2.2.7 P70 to P77 (Port 7) ..... 75
2.2.8 P80 to P87 (Port 8) ..... 76
2.2.9 P90 to P97 (Port 9) ..... 77
2.2.10 P100 to P107 (Port 10) ..... 77
2.2.11 P120 to P127 (Port 12) ..... 78
2.2.12 P130, P137 (Port 13) ..... 79
2.2.13 P140 (Port 14) ..... 79
2.2.14 P150 to P157 (Port 15) ..... 80
2.2.15 VDD, EVDD0, EVDD1, VSS, EVSS0, EVSS1 ..... 80
2.2.16 RESET ..... 81
2.2.17 REGC ..... 81
**2.3 Recommended Connection of Unused Pins ..... 82**

**CHAPTER 3 CPU ARCHITECTURE ..... 88**

**3.1 Memory Space ..... 88**

3.1.1 Internal program memory space ..... 114

3.1.2 Mirror area ..... 118

3.1.3 Internal data memory space ..... 120

3.1.4 Special function register (SFR) area ..... 121

3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area ..... 121

3.1.6 Data memory addressing ..... 122

**3.2 Processor Registers ..... 140**

3.2.1 Control registers ..... 140

3.2.2 General-purpose registers ..... 143

3.2.3 ES and CS registers ..... 145

3.2.4 Special function registers (SFRs) ..... 146

3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers) ..... 151

**3.3 Instruction Address Addressing ..... 184**

3.3.1 Relative addressing ..... 184

3.3.2 Immediate addressing ..... 184

3.3.3 Table indirect addressing ..... 185

3.3.4 Register direct addressing ..... 186

**3.4 Addressing for Processing Data Addresses ..... 187**

3.4.1 Implied addressing ..... 187

3.4.2 Register addressing ..... 187

3.4.3 Direct addressing ..... 188

3.4.4 Short direct addressing ..... 189

3.4.5 SFR addressing ..... 190

3.4.6 Register indirect addressing ..... 191

3.4.7 Based addressing ..... 192

3.4.8 Based indexed addressing ..... 196

3.4.9 Stack addressing ..... 197

**CHAPTER 4 PORT FUNCTIONS ..... 201**

**4.1 Port Functions ..... 201**

**4.2 Port Configuration ..... 203**

4.2.1 Port 0 ..... 204

4.2.2 Port 1 ..... 209

4.2.3 Port 3 ..... 220

4.2.4 Port 4 ..... 226

4.2.5 Port 5 ..... 236

4.2.6 Port 6 ..... 246

4.2.7 Port 7 ..... 256

4.2.8 Port 8 ... 270
4.2.9 Port 9 ... 280
4.2.10 Port 10 ... 283
4.2.11 Port 12 ... 288
4.2.12 Port 13 ... 295
4.2.13 Port 14 ... 298
4.2.14 Port 15 ... 300
**4.3 Registers Controlling Port Function ... 307**
4.3.1 Port mode registers (PMxx) ... 319
4.3.2 Port registers (Pxx) ... 320
4.3.3 Pull-up resistor option registers (PUxx) ... 322
4.3.4 Port input mode registers (PIM1, PIM3, PIM5 to PIM7, PIM12) ... 323
4.3.5 Port output mode registers (POM1, POM6, POM7, POM12) ... 324
4.3.6 Port mode control registers 7, 9, 12 (PMC7, PMC9, PMC12) ... 325
4.3.7 A/D port configuration register (ADPC) ... 326
4.3.8 Port input threshold control register
(PITHL1, PITHL3 to PITHL7, PITHL10, PITHL12, PITHL15) ... 328
4.3.9 Peripheral I/O redirection register 0 (PIOR0) ... 330
4.3.10 Peripheral I/O redirection register 1 (PIOR1) ... 331
4.3.11 Peripheral I/O redirection register 2 (PIOR2) ... 332
4.3.12 Peripheral I/O redirection register 3 (PIOR3) ... 333
4.3.13 Peripheral I/O redirection register 4 (PIOR4) ... 334
4.3.14 Peripheral I/O redirection register 5 (PIOR5) ... 336
4.3.15 Peripheral I/O redirection register 6 (PIOR6) ... 337
4.3.16 Peripheral I/O redirection register 7 (PIOR7) ... 338
4.3.17 Peripheral I/O redirection register 8 (PIOR8) ... 339
4.3.18 Port output slew rate register (PSRSEL) ... 340
4.3.19 SNOOZE status output control register 0 (PSNZCNT0) ... 341
4.3.20 SNOOZE status output control register 1 (PSNZCNT1) ... 342
4.3.21 SNOOZE status output control register 2 (PSNZCNT2) ... 343
4.3.22 SNOOZE status output control register 3 (PSNZCNT3) ... 344
4.3.23 Port mode select register (PMS) ... 345
**4.4 Port Function Operations ... 346**
4.4.1 Writing to I/O port ... 346
4.4.2 Reading from I/O port ... 346
4.4.3 Operations on I/O port ... 346
4.4.4 Connecting to external device with different potential (3 V) ... 347
**4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function ... 349**
**4.6 Cautions When Using Port Function ... 360**
4.6.1 Cautions on 1-bit manipulation instruction for port register n (Pn) ... 360
4.6.2 Notes on specifying the pin settings ... 361

**CHAPTER 5 CLOCK GENERATOR ..... 362**

**5.1 Functions of Clock Generator ..... 362**

**5.2 Configuration of Clock Generator ..... 365**

**5.3 Registers Controlling Clock Generator ..... 369**

5.3.1 Clock Operation Mode Control Register (CMC) ..... 369

5.3.2 System Clock Control Register (CKC) ..... 372

5.3.3 Clock Operation Status Control Register (CSC) ..... 374

5.3.4 Oscillation Stabilization Time Counter Status Register (OSTC) ..... 375

5.3.5 Oscillation Stabilization Time Select Register (OSTS) ..... 377

5.3.6 Peripheral Enable Registers 0, 1, 2 (PER0, PER1, PER2) ..... 379

5.3.7 Operation Speed Mode Control Register (OSMC) ..... 385

5.3.8 High-Speed On-Chip Oscillator Frequency Select Register (HOCODIV) ..... 386

5.3.9 High-Speed On-Chip Oscillator Trimming Register (HIOTRM) ..... 387

5.3.10 CAN Clock Select Register (CANCKSEL) ..... 388

5.3.11 LIN Clock Select Register (LINCKSEL) ..... 389

5.3.12 Clock Select Register (CKSEL) ..... 390

5.3.13 PLL Control Register (PLLCTL) ..... 391

5.3.14 PLL Status Register (PLLSTS) ..... 393

5.3.15 fMP Clock Division Register (MDIV) ..... 394

**5.4 System Clock Oscillator ..... 395**

5.4.1 X1 Oscillator ..... 395

5.4.2 XT1 Oscillator ..... 395

5.4.3 High-Speed On-Chip Oscillator ..... 399

5.4.4 PLL Circuit ..... 399

5.4.5 Low-Speed On-Chip Oscillator ..... 399

5.4.6 WDT-Dedicated Low-Speed On-Chip Oscillator ..... 399

**5.5 Clock Generator Operation ..... 400**

**5.6 Controlling Clock ..... 402**

5.6.1 Example of Setting High-Speed On-Chip Oscillator ..... 402

5.6.2 Example of Setting X1 Oscillator ..... 404

5.6.3 Example of Setting XT1 Oscillator ..... 405

5.6.4 Examples of Setting PLL Circuit ..... 406

5.6.5 Example of Setting Low-Speed On-Chip Oscillator ..... 408

5.6.6 CPU Clock Status Transition Diagram ..... 409

5.6.7 Conditions before Changing CPU Clock and Processing after Changing CPU Clock ..... 413

5.6.8 Time Required for Switchover of CPU Clock, Main System/PLL Select Clock, and Main System Clock ..... 420

5.6.9 Conditions Before Clock Oscillation Is Stopped ..... 422

**5.7 Usage Notes ..... 423**

5.7.1 CPU/Peripheral Hardware Clock ..... 423

5.7.2 High-Speed On-Chip Oscillator ..... 423

**CHAPTER 6 TIMER ARRAY UNIT ..... 424**

**6.1 Functions of Timer Array Unit ..... 426**

6.1.1 Independent channel operation function ..... 426

6.1.2 Simultaneous channel operation function ..... 427

6.1.3 8-bit timer operation function (channels 1 and 3 only) ..... 428

6.1.4 LIN-bus supporting function (channel 7 of unit 0 only) ..... 429

**6.2 Configuration of Timer Array Unit ..... 430**

6.2.1 Timer count register mn (TCRmn) ..... 436

6.2.2 Timer data register mn (TDRmn) ..... 438

**6.3 Registers Controlling Timer Array Unit ..... 439**

6.3.1 Peripheral enable register 0 (PER0) ..... 440

6.3.2 Timer clock select register m (TPSm) ..... 441

6.3.3 Timer mode register mn (TMRmn) ..... 444

6.3.4 Timer status register mn (TSRmn) ..... 450

6.3.5 Timer channel enable status register m (TEm) ..... 451

6.3.6 Timer channel start register m (TSm) ..... 452

6.3.7 Timer channel stop register m (TTm) ..... 453

6.3.8 Timer input select register 0 (TIS0) ..... 454

6.3.9 Timer input select register 1 (TIS1) ..... 455

6.3.10 Timer input select register 2 (TIS2) ..... 456

6.3.11 Timer output enable register m (TOEm) ..... 457

6.3.12 Timer output register m (TOm) ..... 458

6.3.13 Timer output level register m (TOLm) ..... 459

6.3.14 Timer output mode register m (TOMm) ..... 460

6.3.15 Noise filter enable registers 1, 2 (NFEN1, NFEN2) ..... 461

6.3.16 Port mode registers 1, 3, 4, 7, 12 (PM1, PM3, PM4, PM7, PM12) ..... 464

6.3.17 PWM output delay control register 1 (PWMDLY1) ..... 466

6.3.18 PWM output delay control register 2 (PWMDLY2) ..... 467

**6.4 Basic Rules of Timer Array Unit ..... 468**

6.4.1 Basic rules of simultaneous channel operation function ..... 468

6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only) ..... 470

**6.5 Operation Timing of Counter ..... 471**

6.5.1 Count clock (fTCLK) ..... 471

6.5.2 Start timing of counter ..... 473

6.5.3 Operation of counter ..... 474

**6.6 Channel Output (TOmn pin) Control ..... 479**

6.6.1 TOmn pin output circuit configuration ..... 479

6.6.2 TOmn Pin Output Setting ..... 480

6.6.3 Cautions on Channel Output Operation ..... 481

6.6.4 Collective manipulation of TOmn bit ..... 486

6.6.5 Timer Interrupt and TOmn Pin Output at Operation Start ..... 487

**6.7 Independent Channel Operation Function of Timer Array Unit ..... 488**
6.7.1 Operation as interval timer/square wave output ..... 488
6.7.2 Operation as external event counter ..... 494
6.7.3 Operation as frequency divider ..... 499
6.7.4 Operation as input pulse interval measurement ..... 503
6.7.5 Operation as input signal high-/low-level width measurement ..... 508
6.7.6 Operation as delay counter ..... 513
**6.8 Simultaneous Channel Operation Function of Timer Array Unit ..... 518**
6.8.1 Operation as one-shot pulse output function ..... 518
6.8.2 Operation as PWM function ..... 526
6.8.3 Operation as multiple PWM output function ..... 533
**6.9 Cautions When Using Timer Array Unit ..... 542**
6.9.1 Cautions When Using Timer output ..... 542

**CHAPTER 7 TIMER RJ ..... 543**

**7.1 Overview ..... 543**
**7.2 I/O Pins ..... 544**
**7.3 Registers ..... 545**
7.3.1 Peripheral enable register 1 (PER1) ..... 546
7.3.2 Operation speed mode control register (OSMC) ..... 547
7.3.3 Clock Select Register (CKSEL) ..... 547
7.3.4 Timer RJ Counter Register 0 (TRJ0), Timer RJ Reload Register ..... 548
7.3.5 Timer RJ Control Register 0 (TRJCR0) ..... 549
7.3.6 Timer RJ I/O Control Register 0 (TRJIOC0) ..... 551
7.3.7 Timer RJ Mode Register 0 (TRJMR0) ..... 553
7.3.8 Timer RJ Event Pin Select Register 0 (TRJISR0) ..... 554
7.3.9 Port mode registers 1, 4 (PM1, PM4) ..... 555
**7.4 Operation ..... 556**
7.4.1 Reload Register and Counter Rewrite Operation ..... 556
7.4.2 Timer Mode ..... 557
7.4.3 Pulse Output Mode ..... 558
7.4.4 Event Counter Mode ..... 559
7.4.5 Pulse Width Measurement Mode ..... 561
7.4.6 Pulse Period Measurement Mode ..... 562
7.4.7 Coordination with Event Link Controller (ELC) ..... 563
7.4.8 Output Settings for Each Mode ..... 563
**7.5 Notes on Timer RJ ..... 564**
7.5.1 Count Operation Start and Stop Control ..... 564
7.5.2 Access to Flags (Bits TEDGF and TUNDF in TRJCR0 Register) ..... 564
7.5.3 Access to Counter Register ..... 564
7.5.4 When Changing Mode ..... 564

7.5.5 Procedure for Setting Pins TRJO0 and TRJIO0 ..... 565
7.5.6 When Timer RJ is not Used ..... 565
7.5.7 When Timer RJ Operating Clock is Stopped ..... 565
7.5.8 Procedure for Setting STOP Mode (Event Counter Mode) ..... 565
7.5.9 Functional Restriction in STOP Mode (Event Counter Mode Only) ..... 566
7.5.10 When Count is Forcibly Stopped by TSTOP Bit ..... 566
7.5.11 Digital Filter ..... 566
7.5.12 When Selecting fIL as Count Source ..... 566

**CHAPTER 8 TIMER RD ..... 567**

**8.1 Overview ..... 567**
**8.2 Registers ..... 569**
8.2.1 Peripheral enable register 1 (PER1) ..... 570
8.2.2 Clock Select Register (CKSEL) ..... 571
8.2.3 Timer RD ELC Register (TRDELC) ..... 572
8.2.4 Timer RD Start Register (TRDSTR) ..... 573
8.2.5 Timer RD Mode Register (TRDMR) ..... 574
8.2.6 Timer RD PWM Function Select Register (TRDPMR) ..... 575
8.2.7 Timer RD Function Control Register (TRDFCR) ..... 576
8.2.8 Timer RD Output Master Enable Register 1 (TRDOER1) ..... 578
8.2.9 Timer RD Output Master Enable Register 2 (TRDOER2) ..... 579
8.2.10 Timer RD Output Control Register (TRDOCR) ..... 580
8.2.11 Timer RD Digital Filter Function Select Register i (TRDDFi) (i = 0 or 1) ..... 583
8.2.12 Timer RD Control Register i (TRDCRi) (i = 0 or 1) ..... 585
8.2.13 Timer RD I/O Control Register Ai (TRDIORAi) (i = 0 or 1) ..... 590
8.2.14 Timer RD I/O Control Register Ci (TRDIORCi) (i = 0 or 1) ..... 592
8.2.15 Timer RD Status Register i (TRDSRi) (i = 0 or 1) ..... 594
8.2.16 Timer RD Interrupt Enable Register i (TRDIERi) (i = 0 or 1) ..... 598
8.2.17 Timer RD PWM Function Output Level Control Register i (TRDPOCRi) (i = 0 or 1) ..... 599
8.2.18 Timer RD Counter i (TRDi) (i = 0 or 1) ..... 600
8.2.19 Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi,TRDGRCi, TRDGRDi) (i = 0 or 1) ..... 602
8.2.20 PWM Output Delay Control Register 0 (PWMDLY0) ..... 611
8.2.21 Port mode registers (PM1, PM3, PM12) ..... 612
**8.3 Operation ..... 613**
8.3.1 Items Common to Multiple Modes ..... 613
8.3.2 Input Capture Function ..... 624
8.3.3 Output Compare Function ..... 628
8.3.4 PWM Function ..... 633
8.3.5 Reset Synchronous PWM Mode ..... 637
8.3.6 Complementary PWM Mode ..... 640

8.3.7 PWM3 Mode ... 644
**8.4 Timer RD Interrupt ... 647**
**8.5 Notes on Timer RD ... 649**
8.5.1 SFR Read/Write Access ... 649
8.5.2 Mode Switching ... 649
8.5.3 Count Source ... 650
8.5.4 Input Capture Function ... 650
8.5.5 Procedure for Setting Pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi (i = 0 or 1) ... 650
8.5.6 External clock TRDCLK0 ... 650
8.5.7 Reset Synchronous PWM Mode ... 651
8.5.8 Complementary PWM Mode ... 651

**CHAPTER 9 REAL-TIME CLOCK ... 656**

**9.1 Functions of Real-time Clock ... 656**
**9.2 Configuration of Real-time Clock ... 656**
**9.3 Registers Controlling Real-time Clock ... 658**
9.3.1 Peripheral enable register 0 (PER0) ... 659
9.3.2 Operation speed mode control register (OSMC) ... 660
9.3.3 Timer input select register 1 (TIS1) ... 661
9.3.4 Timer input select register 2 (TIS2) ... 662
9.3.5 RTC clock select register (RTCCL) ... 663
9.3.6 Real-time clock control register 0 (RTCC0) ... 664
9.3.7 Real-time clock control register 1 (RTCC1) ... 665
9.3.8 Second count register (SEC) ... 667
9.3.9 Minute count register (MIN) ... 667
9.3.10 Hour count register (HOUR) ... 668
9.3.11 Day count register (DAY) ... 670
9.3.12 Week count register (WEEK) ... 671
9.3.13 Month count register (MONTH) ... 672
9.3.14 Year count register (YEAR) ... 672
9.3.15 Watch error correction register (SUBCUD) ... 673
9.3.16 16-bit watch error correction register (SUBCUDW) ... 674
9.3.17 Alarm minute register (ALARMWM) ... 675
9.3.18 Alarm hour register (ALARMWH) ... 675
9.3.19 Alarm week register (ALARMWW) ... 675
**9.4 Real-time Clock Operation ... 677**
9.4.1 Starting operation of real-time clock ... 677
9.4.2 Shifting to HALT/STOP mode after starting operation ... 678
9.4.3 Reading/writing real-time clock ... 679
9.4.4 Setting alarm of real-time clock ... 681
9.4.5 1 Hz output of real-time clock ... 682

9.4.6 Example of watch error correction of real-time clock ..... 683

**CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER ..... 686**

**10.1 Functions of Clock Output/Buzzer Output Controller ..... 686**
**10.2 Configuration of Clock Output/Buzzer Output Controller ..... 688**
**10.3 Registers Controlling Clock Output/Buzzer Output Controller ..... 688**
10.3.1 Clock output select register 0 (CKS0) ..... 688
10.3.2 Clock Select Register (CKSEL) ..... 690
10.3.3 Port mode register 14 (PM14) ..... 691
**10.4 Operations of Clock Output/Buzzer Output Controller ..... 692**
10.4.1 Operation as output pin ..... 692
**10.5 Notes on Clock Output/Buzzer Output Controller ..... 692**

**CHAPTER 11 WATCHDOG TIMER ..... 693**

**11.1 Functions of Watchdog Timer ..... 693**
**11.2 Configuration of Watchdog Timer ..... 694**
**11.3 Register Controlling Watchdog Timer ..... 695**
11.3.1 Watchdog timer enable register (WDTE) ..... 695
**11.4 Operation of Watchdog Timer ..... 696**
11.4.1 Controlling operation of watchdog timer ..... 696
11.4.2 Setting overflow time of watchdog timer ..... 697
11.4.3 Setting window open period of watchdog timer ..... 698
11.4.4 Setting watchdog timer interval interrupt ..... 699

**CHAPTER 12 A/D CONVERTER ..... 700**

**12.1 Function of A/D Converter ..... 701**
**12.2 Configuration of A/D Converter ..... 703**
**12.3 Registers Used in A/D Converter ..... 705**
12.3.1 Peripheral enable register 0 (PER0) ..... 706
12.3.2 A/D converter mode register 0 (ADM0) ..... 707
12.3.3 A/D converter mode register 1 (ADM1) ..... 716
12.3.4 A/D converter mode register 2 (ADM2) ..... 717
12.3.5 10-bit A/D conversion result register (ADCR) ..... 720
12.3.6 8-bit A/D conversion result register (ADCRH) ..... 721
12.3.7 Analog input channel specification register (ADS) ..... 722
12.3.8 Conversion result comparison upper limit setting register (ADUL) ..... 725
12.3.9 Conversion result comparison lower limit setting register (ADLL) ..... 725
12.3.10 A/D test register (ADTES) ..... 726
12.3.11 A/D port configuration register (ADPC) ..... 727
12.3.12 A/D converter trigger select register 0 (ADTRGS0) (RL78/F13 only) ..... 728

12.3.13 A/D converter trigger select register 1 (ADTRGS1) (RL78/F13 only) ..... 729
12.3.14 Port mode control registers 7, 9, and 12 (PMC7, PMC9, PMC12) ..... 730
12.3.15 Port mode registers 3, 7 to 10, and 12 (PM3, PM7 to PM10, PM12) ..... 731
**12.4 A/D Converter Conversion Operations ..... 733**
**12.5 Input Voltage and Conversion Results ..... 735**
**12.6 A/D Converter Operation Modes ..... 736**
12.6.1 Software trigger mode (select mode, sequential conversion mode) ..... 736
12.6.2 Software trigger mode (select mode, one-shot conversion mode) ..... 737
12.6.3 Software trigger mode (scan mode, sequential conversion mode) ..... 738
12.6.4 Software trigger mode (scan mode, one-shot conversion mode) ..... 739
12.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode) ..... 740
12.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode) ..... 741
12.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode) ..... 742
12.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode) ..... 743
12.6.9 Hardware trigger wait mode (select mode, sequential conversion mode) ..... 744
12.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode) ..... 745
12.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode) ..... 746
12.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode) ..... 747
**12.7 A/D Converter Setup Flowchart ..... 748**
12.7.1 Setting up software trigger mode ..... 749
12.7.2 Setting up hardware trigger no-wait mode ..... 750
12.7.3 Setting up hardware trigger wait mode ..... 751
12.7.4 Setup when using temperature sensor
(example for software trigger mode and one-shot conversion mode) ..... 752
12.7.5 Setting up test mode ..... 753
**12.8 SNOOZE Mode Function ..... 754**
12.8.1 If an interrupt is generated after A/D conversion ends ..... 755
12.8.2 If no interrupt is generated after A/D conversion ends ..... 756
**12.9 How to Read A/D Converter Characteristics Table ..... 757**
**12.10 Cautions for A/D Converter ..... 759**

**CHAPTER 13 D/A CONVERTER (RL78/F14 Only) ..... 763**

**13.1 Function of D/A Converter ..... 763**
**13.2 Configuration of D/A Converter ..... 764**
**13.3 Registers of D/A Converter ..... 765**
13.3.1 A/D Port Configuration Register (ADPC) ..... 766
13.3.2 Peripheral Enable Register 1 (PER1) ..... 767
13.3.3 D/A Converter Mode Register (DAM) ..... 768
13.3.4 D/A Converter Mode Register 2 (DAM2) ..... 769
13.3.5 D/A Conversion Value Setting Register 0 (DACS0) ..... 770
13.3.6 Port Mode Register 8 (PM8) ..... 771

**13.4 Operations of D/A Converter ... 773**
13.4.1 Operation in Normal Mode ... 773
13.4.2 Operation in Real-Time Output Mode ... 774
**13.5 Cautions for D/A Converter ... 775**

**CHAPTER 14 COMPARATOR (RL78/F14 Only) ... 776**

**14.1 Overview ... 776**
**14.2 Registers to Control the Comparator ... 778**
14.2.1 Peripheral Enable Register 1 (PER1) ... 778
14.2.2 Comparator Control Register (CMPCTL) ... 779
14.2.3 Comparator I/O Select Register (CMPSEL) ... 781
14.2.4 Comparator Output Monitor Register (CMPMON) ... 782
14.2.5 A/D port configuration register (ADPC) ... 783
14.2.6 D/A converter mode register 2 (DAM2) ... 784
14.2.7 Port mode register 4 (PM4) ... 785
14.2.8 Port mode register (PM8) ... 786
**14.3 Operation ... 787**
14.3.1 Noise Filter ... 788
14.3.2 Comparator Interrupts ... 789
14.3.3 Comparator ELC Event Output ... 789
14.3.4 Comparator Pin Output ... 789
14.3.5 Stopping or Supplying Comparator Clock ... 789
14.3.6 Comparator Setting Flowchart ... 790

**CHAPTER 15 SERIAL ARRAY UNIT ... 792**

**15.1 Functions of Serial Array Unit ... 793**
15.1.1 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11) ... 793
15.1.2 UART (UART0, UART1) ... 794
15.1.3 Simplified $I^2C$ (IIC00, IIC01, IIC10, IIC11) ... 795
**15.2 Configuration of Serial Array Unit ... 796**
**15.3 Registers Controlling Serial Array Unit ... 802**
15.3.1 Peripheral enable register 0 (PER0) ... 803
15.3.2 Serial clock select register m (SPSm) ... 804
15.3.3 Serial mode register mn (SMRmn) ... 805
15.3.4 Serial communication operation setting register mn (SCRmn) ... 807
15.3.5 Higher 7 bits of the serial data register mn (SDRmn) ... 810
15.3.6 Serial flag clear trigger register mn (SIRmn) ... 812
15.3.7 Serial status register mn (SSRmn) ... 813
15.3.8 Serial channel start register m (SSm) ... 815
15.3.9 Serial channel stop register m (STm) ... 816
15.3.10 Serial channel enable status register m (SEm) ... 817

15.3.11 Serial output enable register m (SOEm) ... 818
15.3.12 Serial output register m (SOm) ... 819
15.3.13 Serial output level register m (SOLm) ... 820
15.3.14 Serial slave select enable register m (SSEm) ... 821
15.3.15 Input switch control register (ISC) ... 822
15.3.16 Noise filter enable register 0 (NFEN0) ... 823
15.3.17 Port input mode registers 1, 3, 5 to 7, 12 (PIM1, PIM3, PIM5 to PIM7, PIM12) ... 824
15.3.18 Port output mode registers 1, 6, 7, 12 (POM1, POM6, POM7, POM12) ... 825
15.3.19 Port mode registers 1, 3 to 7, 12 (PM1, PM3 to PM7, PM12) ... 826
**15.4 Operation stop mode ... 828**
15.4.1 Stopping the operation by units ... 829
15.4.2 Stopping the operation by channels ... 830
**15.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) Communication ... 832**
15.5.1 Master transmission ... 834
15.5.2 Master reception ... 844
15.5.3 Master transmission/reception ... 854
15.5.4 Slave transmission ... 864
15.5.5 Slave reception ... 874
15.5.6 Slave transmission/reception ... 881
15.5.7 Calculating transfer clock frequency ... 891
15.5.8 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11) communication ... 893
**15.6 Clock Synchronous Serial Communication with SPI Function ... 894**
15.6.1 Master transmission ... 898
15.6.2 Master reception ... 908
15.6.3 Master transmission/reception ... 918
15.6.4 Slave transmission ... 928
15.6.5 Slave reception ... 938
15.6.6 Slave transmission/reception ... 945
15.6.7 Calculating transfer clock frequency ... 955
15.6.8 Procedure for processing errors that occurred during clock synchronous serial communication with SPI function ... 957
**15.7 Operation of UART (UART0, UART1) Communication ... 958**
15.7.1 UART transmission ... 960
15.7.2 UART reception ... 970
15.7.3 Calculating baud rate ... 977
15.7.4 Procedure for processing errors that occurred during UART (UART0, UART1) communication ... 981
**15.8 LIN Communication Operation ... 982**
15.8.1 LIN transmission ... 982
15.8.2 LIN reception ... 985
**15.9 Operation of Simplified $I^2C$ (IIC00, IIC01, IIC10, IIC11) Communication ... 991**

15.9.1 Address field transmission ..... 994
15.9.2 Data transmission ..... 1000
15.9.3 Data reception ..... 1005
15.9.4 Stop condition generation ..... 1010
15.9.5 Calculating transfer rate ..... 1012
15.9.6 Procedure for processing errors that occurred during simplified I²C (IIC00, IIC01, IIC10, IIC11) communication ..... 1015

**CHAPTER 16 SERIAL INTERFACE IICA ..... 1016**

**16.1 Functions of Serial Interface IICA ..... 1016**
**16.2 Configuration of Serial Interface IICA ..... 1019**
**16.3 Registers Controlling Serial Interface IICA ..... 1022**
16.3.1 Peripheral enable register 0 (PER0) ..... 1022
16.3.2 IICA control register 00 (IICCTL00) ..... 1023
16.3.3 IICA status register 0 (IICS0) ..... 1028
16.3.4 IICA flag register 0 (IICF0) ..... 1030
16.3.5 IICA control register 01 (IICCTL01) ..... 1032
16.3.6 IICA low-level width setting register 0 (IICWL0) ..... 1034
16.3.7 IICA high-level width setting register 0 (IICWH0) ..... 1034
16.3.8 Port mode register 6 (PM6) ..... 1035
16.3.9 Port output mode register (POM6) ..... 1036
**16.4 I²C Bus Mode Functions ..... 1037**
16.4.1 Pin configuration ..... 1037
16.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers ..... 1038
**16.5 I²C Bus Definitions and Control Methods ..... 1040**
16.5.1 Start conditions ..... 1040
16.5.2 Addresses ..... 1041
16.5.3 Transfer direction specification ..... 1041
16.5.4 Acknowledge (ACK) ..... 1042
16.5.5 Stop condition ..... 1043
16.5.6 Wait ..... 1044
16.5.7 Canceling wait ..... 1046
16.5.8 Interrupt request (INTIICA0) generation timing and wait control ..... 1047
16.5.9 Address match detection method ..... 1048
16.5.10 Error detection ..... 1048
16.5.11 Extension code ..... 1048
16.5.12 Arbitration ..... 1049
16.5.13 Wakeup function ..... 1051
16.5.14 Communication reservation ..... 1054
16.5.15 Cautions ..... 1058
16.5.16 Communication operations ..... 1059

16.5.17 Timing of $I^2C$ interrupt request (INTIICA0) occurrence ... 1066
**16.6 Timing Charts ... 1087**

**CHAPTER 17 LIN/UART MODULE (RLIN3) ... 1102**

**17.1 Overview ... 1102**
**17.2 Register Descriptions ... 1107**
17.2.1 LIN Registers for Master Mode ... 1109
17.2.2 LIN Registers for Slave Mode ... 1137
17.2.3 Registers for UART ... 1165
**17.3 Modes ... 1195**
17.3.1 LIN Reset Mode ... 1197
17.3.2 LIN Mode ... 1198
17.3.3 UART Mode ... 1200
17.3.4 LIN Self-Test Mode ... 1200
**17.4 LIN Mode ... 1201**
17.4.1 Operation Overview ... 1201
17.4.2 Data Transmission/Reception ... 1209
17.4.3 Transmission/Reception Data Buffering ... 1211
17.4.4 Wake-up Transmission/Reception ... 1214
17.4.5 Status ... 1216
17.4.6 Error Status ... 1218
**17.5 UART Mode ... 1224**
17.5.1 Operation Overview ... 1224
17.5.2 Data Transmission/Reception ... 1239
17.5.3 Buffer Processing of Transmission Data ... 1241
17.5.4 Status ... 1242
17.5.5 Error Status ... 1243
**17.6 LIN Self-Test Mode ... 1244**
17.6.1 Change to LIN Self-Test Mode ... 1245
17.6.2 Transmission in LIN Master Self-Test Mode ... 1246
17.6.3 Reception in LIN Master Self-Test Mode ... 1247
17.6.4 Transmission in LIN Slave Self-Test Mode ... 1248
17.6.5 Reception in LIN Slave Self-Test Mode ... 1249
17.6.6 Terminating LIN Self-Test Mode ... 1250
**17.7 Baud Rate Generator ... 1251**
17.7.1 LIN Master Mode ... 1251
17.7.2 LIN Slave Mode ... 1253
17.7.3 UART Mode ... 1255
**17.8 Noise Filter ... 1256**
**17.9 Interrupts ... 1258**

**CHAPTER 18 CAN INTERFACE (RS-CAN LITE) ..... 1259**

**18.1 Overview ..... 1259**

**18.2 Input/Output Pins ..... 1262**

**18.3 Register Descriptions ..... 1263**

18.3.1 CANi Bit Configuration Register L (CiCFGL) (i = 0) ..... 1285

18.3.2 CANi Bit Configuration Register H (CiCFGH) (i = 0) ..... 1286

18.3.3 CANi Control Register L (CiCTRL) (i = 0) ..... 1288

18.3.4 CANi Control Register H (CiCTRH) (i = 0) ..... 1290

18.3.5 CANi Status Register L (CiSTSL) (i = 0) ..... 1292

18.3.6 CANi Status Register H (CiSTSH) (i = 0) ..... 1294

18.3.7 CANi Error Flag Register L (CiERFLL) (i = 0) ..... 1295

18.3.8 CANi Error Flag Register H (CiERFLH) (i = 0) ..... 1298

18.3.9 CAN Global Configuration Register L (GCFGL) ..... 1299

18.3.10 CAN Global Configuration Register H (GCFGH) ..... 1301

18.3.11 CAN Global Control Register L (GCTRL) ..... 1302

18.3.12 CAN Global Control Register H (GCTRH) ..... 1303

18.3.13 CAN Global Status Register (GSTS) ..... 1304

18.3.14 CAN Global Error Flag Register (GERFLL) ..... 1305

18.3.15 CAN Global Transmit Interrupt Status Register (GTINTSTS) ..... 1306

18.3.16 CAN Timestamp Register (GTSC) ..... 1307

18.3.17 CAN Receive Rule Number Configuration Register (GAFLCFG) ..... 1308

18.3.18 CAN Receive Rule Entry Register jAL (GAFLIDLj) (j = 0 to 15) ..... 1309

18.3.19 CAN Receive Rule Entry Register jAH (GAFLIDHj) (j = 0 to 15) ..... 1310

18.3.20 CAN Receive Rule Entry Register jBL (GAFLMLj) (j = 0 to 15) ..... 1311

18.3.21 CAN Receive Rule Entry Register jBH (GAFLMHj) (j = 0 to 15) ..... 1312

18.3.22 CAN Receive Rule Entry Register jCL (GAFLPLj) (j = 0 to 15) ..... 1313

18.3.23 CAN Receive Rule Entry Register jCH (GAFLPHj) (j = 0 to 15) ..... 1314

18.3.24 CAN Receive Buffer Number Configuration Register (RMNB) ..... 1315

18.3.25 CAN Receive Buffer Receive Complete Flag Register (RMND0) ..... 1316

18.3.26 CAN Receive Buffer Register nAL (RMIDLn) (n = 0 to 15) ..... 1317

18.3.27 CAN Receive Buffer Register nAH (RMIDHn) (n = 0 to 15) ..... 1318

18.3.28 CAN Receive Buffer Register nBL (RMTSn) (n = 0 to 15) ..... 1319

18.3.29 CAN Receive Buffer Register nBH (RMPTRn) (n = 0 to 15) ..... 1320

18.3.30 CAN Receive Buffer Register nCL (RMDF0n) (n = 0 to 15) ..... 1321

18.3.31 CAN Receive Buffer Register nCH (RMDF1n) (n = 0 to 15) ..... 1322

18.3.32 CAN Receive Buffer Register nDL (RMDF2n) (n = 0 to 15) ..... 1323

18.3.33 CAN Receive Buffer Register nDH (RMDF3n) (n = 0 to 15) ..... 1324

18.3.34 CAN Receive FIFO Control Register m (RFCCm) (m = 0, 1) ..... 1325

18.3.35 CAN Receive FIFO Status Register m (RFSTSm) (m = 0, 1) ..... 1327

18.3.36 CAN Receive FIFO Pointer Control Register m (RFPCTRm) (m = 0, 1) ..... 1329

18.3.37 CAN Receive FIFO Access Register mAL (RFIDLm) (m = 0, 1) ..... 1330

18.3.38 CAN Receive FIFO Access Register mAH (RFIDHm) (m = 0, 1) ..... 1331
18.3.39 CAN Receive FIFO Access Register mBL (RFTSm) (m = 0, 1)..... 1332
18.3.40 CAN Receive FIFO Access Register mBH (RFPTRm) (m = 0, 1) ..... 1333
18.3.41 CAN Receive FIFO Access Register mCL (RFDF0m) (m = 0, 1) ..... 1334
18.3.42 CAN Receive FIFO Access Register mCH (RFDF1m) (m = 0, 1)..... 1335
18.3.43 CAN Receive FIFO Access Register mDL (RFDF2m) (m = 0, 1) ..... 1336
18.3.44 CAN Receive FIFO Access Register mDH (RFDF3m) (m = 0, 1)..... 1337
18.3.45 CANi Transmit/Receive FIFO Control Register kL (CFCCLk) (i = 0) (k = 0) ..... 1338
18.3.46 CANi Transmit/Receive FIFO Control Register kH (CFCCHk) (i = 0) (k = 0)..... 1340
18.3.47 CANi Transmit/Receive FIFO Status Register k (CFSTSk) (i = 0) (k = 0)..... 1342
18.3.48 CANi Transmit/Receive FIFO Pointer Control Register k (CFPCTRk) (i = 0) (k = 0) ..... 1344
18.3.49 CANi Transmit/Receive FIFO Access Register kAL (CFIDLk) (i = 0) (k = 0) ..... 1345
18.3.50 CANi Transmit/Receive FIFO Access Register kAH (CFIDHk) (i = 0) (k = 0) ..... 1346
18.3.51 CANi Transmit/Receive FIFO Access Register kBL (CFTSk) (i = 0) (k = 0) ..... 1347
18.3.52 CANi Transmit/Receive FIFO Access Register kBH (CFPTRk) (i = 0) (k = 0) ..... 1348
18.3.53 CANi Transmit/Receive FIFO Access Register kCL (CFDF0k) (i = 0) (k = 0)..... 1349
18.3.54 CANi Transmit/Receive FIFO Access Register kCH (CFDF1k) (i = 0) (k = 0) ..... 1350
18.3.55 CANi Transmit/Receive FIFO Access Register kDL (CFDF2k) (i = 0) (k = 0)..... 1351
18.3.56 CANi Transmit/Receive FIFO Access Register kDH (CFDF3k) (i = 0) (k = 0) ..... 1352
18.3.57 Receive FIFO Message Lost Status Register (RFMSTS)..... 1353
18.3.58 CANi Transmit/Receive FIFO Message Lost Status Register (CFMSTS) (i = 0) ..... 1354
18.3.59 CAN Receive FIFO Interrupt Status Register (RFISTS)..... 1355
18.3.60 CAN Transmit/Receive FIFO Receive Interrupt Status Register (CFISTS) ..... 1356
18.3.61 CANi Transmit Buffer Control Register p (TMCp) (i = 0) (p = 0 to 3) ..... 1357
18.3.62 CANi Transmit Buffer Status Register p (TMSTSp) (i = 0) (p = 0 to 3) ..... 1359
18.3.63 CANi Transmit Buffer Transmit Request Status Register (TMTRSTS) (i = 0) ..... 1360
18.3.64 CANi Transmit Buffer Transmit Complete Status Register (TMTCSTS) (i = 0)..... 1361
18.3.65 CANi Transmit Buffer Transmit Abort Status Register (TMTASTS) (i = 0) ..... 1362
18.3.66 CANi Transmit Buffer Interrupt Enable Register (TMIEC) (i = 0) ..... 1363
18.3.67 CANi Transmit Buffer Register pAL (TMIDLp) (i = 0) (p = 0 to 3) ..... 1364
18.3.68 CANi Transmit Buffer Register pAH (TMIDHp) (i = 0) (p = 0 to 3) ..... 1365
18.3.69 CANi Transmit Buffer Register pBH (TMPTRp) (i = 0) (p = 0 to 3) ..... 1366
18.3.70 CANi Transmit Buffer Register pCL (TMDF0p) (i = 0) (p = 0 to 3)..... 1367
18.3.71 CANi Transmit Buffer Register pCH (TMDF1p) (i = 0) (p = 0 to 3) ..... 1368
18.3.72 CANi Transmit Buffer Register pDL (TMDF2p) (i = 0) (p = 0 to 3)..... 1369
18.3.73 CANi Transmit Buffer Register pDH (TMDF3p) (i = 0) (p = 0 to 3) ..... 1370
18.3.74 CANi Transmit History Buffer Control Register (THLCCi) (i = 0) ..... 1371
18.3.75 CANi Transmit History Buffer Status Register (THLSTSi) (i = 0) ..... 1372
18.3.76 CANi Transmit History Buffer Access Register (THLACCi) (i = 0) ..... 1373
18.3.77 CANi Transmit History Buffer Pointer Control Register (THLPCTRi) (i = 0) ..... 1374
18.3.78 CAN Global RAM Window Control Register (GRWCR)..... 1375

18.3.79 CAN Global Test Configuration Register (GTSTCFG) ..... 1376
18.3.80 CAN Global Test Control Register (GTSTCTRL) ..... 1377
18.3.81 CAN Global Test Protection Unlock Register (GLOCKK) ..... 1378
18.3.82 CAN RAM Test Register r (RPGACCr) (r = 0 to 127) ..... 1379
**18.4 CAN Modes ..... 1380**
18.4.1 Global Modes ..... 1380
18.4.2 Channel Modes ..... 1382
**18.5 Reception Function ..... 1387**
18.5.1 Data Processing Using the Receive Rule Table ..... 1387
18.5.2 Timestamp ..... 1389
**18.6 Transmission Functions ..... 1390**
18.6.1 Transmit Priority Determination ..... 1390
18.6.2 Transmission Using Transmit Buffers ..... 1391
18.6.3 Transmission Using FIFO Buffers ..... 1392
18.6.4 Transmit History Function ..... 1394
**18.7 Test Function ..... 1395**
18.7.1 Standard Test Mode ..... 1395
18.7.2 Listen-Only Mode ..... 1395
18.7.3 Self-Test Mode (Loopback Mode) ..... 1396
18.7.4 RAM Test ..... 1396
**18.8 Interrupt ..... 1397**
**18.9 RAM Window ..... 1401**
**18.10 Initial Settings ..... 1402**
18.10.1 Clock Setting ..... 1404
18.10.2 Bit Timing Setting ..... 1404
18.10.3 Communication Speed Setting ..... 1405
18.10.4 Receive Rule Setting ..... 1407
18.10.5 Buffer Setting ..... 1408
**18.11 Reception Procedure ..... 1410**
18.11.1 Receive Buffer Reading Procedure ..... 1410
18.11.2 FIFO Buffer Reading Procedure ..... 1412
**18.12 Transmission Procedure ..... 1415**
18.12.1 Procedure for Transmission from Transmit Buffers ..... 1415
18.12.2 Procedure for Transmission from Transmit/Receive FIFO Buffers ..... 1418
18.12.3 Transmit History Buffer Reading Procedure ..... 1421
**18.13 Test Settings ..... 1422**
18.13.1 Self-Test Mode Setting Procedure ..... 1422
18.13.2 Protection Unlock Procedure ..... 1423
18.13.3 RAM Test Setting Procedure ..... 1424
**18.14 Notes on the CAN Module ..... 1425**

**CHAPTER 19 DTC ..... 1426**

**19.1 Overview ..... 1426**

**19.2 Registers ..... 1428**

19.2.1 Allocation of DTC Control Data Area and DTC Vector Table Area ..... 1430

19.2.2 DTC Control Data Allocation ..... 1431

19.2.3 DTC Vector Table ..... 1432

19.2.4 Peripheral enable register 1 (PER1) ..... 1436

19.2.5 DTC Activation Enable Register i (DTCENi) (i = 0 to 5) ..... 1437

19.2.6 DTC Base Address Register (DTCBAR) ..... 1440

19.2.7 DTC Control Register j (DTCCRj) (j = 0 to 23) ..... 1441

19.2.8 DTC Block Size Register j (DTBLSj) (j = 0 to 23) ..... 1442

19.2.9 DTC Transfer Count Register j (DTCCTj) (j = 0 to 23) ..... 1442

19.2.10 DTC Transfer Count Reload Register j (DTRLDj) (j = 0 to 23) ..... 1443

19.2.11 DTC Source Address Register j (DTSARj) (j = 0 to 23) ..... 1443

19.2.12 DTC Destination Address Register j (DTDARj) (j = 0 to 23) ..... 1443

19.2.13 High-speed DTC Channel Select Register 0 (SELHS0) ..... 1444

19.2.14 High-speed DTC Channel Select Register 1 (SELHS1) ..... 1445

19.2.15 High-speed DTC Control Register m (HDTCCR0/1) (m = 0, 1) ..... 1446

19.2.16 High-speed DTC Transfer Count Register m (HDTCCT0/1) (m = 0, 1) ..... 1447

19.2.17 DTC Transfer Count Reload Register m (HDTRLD0/1) (m = 0, 1) ..... 1448

19.2.18 High-speed DTC Source Address Register m (HDTSAR0/1) (m = 0, 1) ..... 1448

19.2.19 High-speed DTC Destination Address Register m (HDTDAR0/1) (m = 0, 1) ..... 1448

**19.3 Operation ..... 1449**

19.3.1 Activation Sources ..... 1449

19.3.2 Normal Mode ..... 1451

19.3.3 Repeat Mode ..... 1453

19.3.4 Chain Transfers ..... 1457

19.3.5 High-Speed Transfer Operation ..... 1459

**19.4 Notes on DTC ..... 1460**

19.4.1 Setting DTC Registers and Vector Table ..... 1460

19.4.2 Allocation of DTC Control Data Area and DTC Vector Table Area ..... 1460

19.4.3 DTC Pending Instruction ..... 1461

19.4.4 Operations when an Instruction which Accesses an SFR Register that Requires a Wait is Executed ..... 1461

19.4.5 Operation when Accessing Data Flash Memory Space ..... 1461

19.4.6 Number of DTC Execution Clock Cycles ..... 1462

19.4.7 Number of High-speed DTC Execution Clock Cycles ..... 1463

19.4.8 DTC Response Time ..... 1464

19.4.9 DTC Activation Sources ..... 1464

19.4.10 Operation in Standby Mode Status ..... 1465

19.4.11 Notes When the RAM Area Is the Source of the Data for Transfer ..... 1465

19.4.12 Vector Address for High-Speed Transfer ... 1465

**CHAPTER 20 EVENT LINK CONTROLLER (ELC) (RL78/F14 Only) ... 1466**

**20.1 Overview ... 1466**

**20.2 Registers ... 1467**

20.2.1 Event Output Destination Select Register n (ELSELRn) (n = 00 to 25) ... 1468

20.2.2 Timer input select register 0 ... 1471

20.2.3 A/D converter mode register 1 (ADM1) ... 1471

20.2.4 D/A converter mode register (DAM) ... 1471

**20.3 Operation ... 1472**

**CHAPTER 21 INTERRUPT FUNCTIONS ... 1473**

**21.1 Interrupt Function Types ... 1473**

**21.2 Interrupt Sources and Configuration ... 1474**

**21.3 Registers Controlling Interrupt Functions ... 1484**

21.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L) ... 1490

21.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L) ... 1492

21.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) ... 1494

21.3.4 External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1) ... 1497

21.3.5 Interrupt source determination flag register 0 (INTFLG0) ... 1499

21.3.6 Interrupt mask register (INTMSK) ... 1502

21.3.7 Input switch control register (ISC) ... 1503

21.3.8 Program status word (PSW) ... 1504

**21.4 Interrupt Servicing Operations ... 1505**

21.4.1 Maskable interrupt request acknowledgment ... 1505

21.4.2 Software interrupt request acknowledgment ... 1508

21.4.3 Multiple interrupt servicing ... 1508

21.4.4 Interrupt servicing during division instruction ... 1512

21.4.5 Interrupt request hold ... 1514

**CHAPTER 22 KEY INTERRUPT FUNCTION ... 1515**

**22.1 Functions of Key Interrupt ... 1515**

**22.2 Configuration of Key Interrupt ... 1516**

**22.3 Register Controlling Key Interrupt ... 1517**

22.3.1 Key return mode register (KRM) ... 1517

**CHAPTER 23 STANDBY FUNCTION ... 1518**

**23.1 Standby Function and Configuration ... 1518**

23.1.1 Standby function ... 1518
**23.2 Registers controlling standby function ... 1519**
23.2.1 Oscillation stabilization time counter status register (OSTC) ... 1520
23.2.2 Oscillation stabilization time select register (OSTS) ... 1521
23.2.3 STOP status output control register (STPSTC) ... 1522
**23.3 Standby Function Operation ... 1523**
23.3.1 HALT mode ... 1523
23.3.2 STOP mode ... 1530
23.3.3 SNOOZE mode ... 1536

**CHAPTER 24 RESET FUNCTION ... 1541**

**24.1 Register for Confirming Reset Source ... 1552**
24.1.1 Reset control flag register (RESF) ... 1552
24.1.2 POR/CLM reset confirmation register (POCRES) ... 1553

**CHAPTER 25 POWER-ON-RESET CIRCUIT ... 1555**

**25.1 Functions of Power-on-reset Circuit ... 1555**
**25.2 Configuration of Power-on-reset Circuit ... 1556**
**25.3 Operation of Power-on-reset Circuit ... 1556**
**25.4 Cautions for Power-on-reset Circuit ... 1559**

**CHAPTER 26 VOLTAGE DETECTOR ... 1561**

**26.1 Functions of Voltage Detector ... 1561**
**26.2 Configuration of Voltage Detector ... 1562**
**26.3 Registers Controlling Voltage Detector ... 1562**
26.3.1 Voltage detection register (LVIM) ... 1563
26.3.2 Voltage detection level register (LVIS) ... 1564
**26.4 Operation of Voltage Detector ... 1567**
26.4.1 When used as reset mode ... 1567
26.4.2 When used as interrupt mode ... 1569
26.4.3 When used as interrupt and reset mode ... 1571
**26.5 Cautions for Voltage Detector ... 1577**
26.5.1 Checking reset source ... 1577
26.5.2 Delay from the time LVD reset source is generated until the time LVD reset has been generated or released ... 1578

**CHAPTER 27 SAFETY FUNCTIONS ... 1579**

**27.1 Overview of Safety Functions ... 1579**
**27.2 Registers Used by Safety Functions ... 1581**
**27.3 Operation of Safety Functions ... 1582**

27.3.1 Flash memory CRC operation function (high-speed CRC) ... 1582
27.3.2 CRC operation function (general-purpose CRC) ... 1586
27.3.3 RAM-ECC function ... 1590
27.3.4 CPU stack pointer monitor function ... 1596
27.3.5 Clock monitor ... 1599
27.3.6 RAM guard function ... 1600
27.3.7 SFR guard function ... 1601
27.3.8 Invalid memory access detection function ... 1602
27.3.9 Frequency detection function ... 1605
27.3.10 A/D test function ... 1608
27.3.11 Digital output signal level detection function for I/O ports ... 1613

**CHAPTER 28 REGULATOR ... 1614**

**28.1 Regulator Overview ... 1614**

**CHAPTER 29 OPTION BYTE ... 1615**

**29.1 Functions of Option Bytes ... 1615**
29.1.1 User option byte (000C0H to 000C2H/020C0H to 020C2H) ... 1615
29.1.2 On-chip debug option byte (000C3H/ 020C3H) ... 1616
**29.2 Format of User Option Byte ... 1617**
**29.3 Format of On-chip Debug Option Byte ... 1621**
**29.4 Setting of Option Byte ... 1622**

**CHAPTER 30 FLASH MEMORY ... 1623**

**30.1 Serial Programming Using Flash Memory Programmer ... 1625**
30.1.1 Programming Environment ... 1627
30.1.2 Communication Mode ... 1627
**30.2 Serial Programming Using External Device (that Incorporates UART) ... 1629**
30.2.1 Programming Environment ... 1629
30.2.2 Communication Mode ... 1630
**30.3 Connection of Pins on Board ... 1631**
30.3.1 P40/TOOL0 pin ... 1631
30.3.2 RESET pin ... 1631
30.3.3 Port pins ... 1632
30.3.4 REGC pin ... 1632
30.3.5 X1 and X2 pins ... 1632
30.3.6 Power supply ... 1632
**30.4 Serial Programming Method ... 1633**
30.4.1 Serial programming procedure ... 1633
30.4.2 Flash memory programming mode ... 1634

30.4.3 Selecting communication mode ... 1635
30.4.4 Communication commands ... 1636
**30.5 Processing Time for Each Command when PG-FP5 Is in Use (Reference Value) ... 1637**
**30.6 Self-Programming ... 1638**
30.6.1 Self-programming procedure ... 1639
30.6.2 Boot swap function ... 1640
30.6.3 Flash shield window function ... 1642
**30.7 Security Settings ... 1643**
**30.8 Data Flash ... 1645**
30.8.1 Data flash overview ... 1645
30.8.2 Register controlling data flash memory ... 1646
30.8.3 Procedure for accessing data flash memory ... 1647

**CHAPTER 31 ON-CHIP DEBUG FUNCTION ... 1648**

**31.1 Overview of On-chip Debug Function ... 1648**
31.1.1 Hot Plug-in ... 1648
31.1.2 Real-time RAM Monitor (RRM) and Dynamic Memory Modification (DMM) by DTC ... 1648
**31.2 Connecting E1 On-chip Debugging Emulator to RL78/F13 or RL78/F14 ... 1650**
**31.3 On-Chip Debug Security ID ... 1651**
**31.4 Securing of User Resources ... 1651**
31.4.1 Securement of memory space ... 1651

**CHAPTER 32 BCD CORRECTION CIRCUIT ... 1654**

**32.1 BCD Correction Circuit Function ... 1654**
**32.2 Registers Used by BCD Correction Circuit ... 1654**
**32.3 BCD Correction Circuit Operation ... 1655**

**CHAPTER 33 INSTRUCTION SET ... 1657**

**33.1 Conventions Used in Operation List ... 1658**
33.1.1 Operand identifiers and specification methods ... 1658
33.1.2 Description of operation column ... 1659
33.1.3 Description of flag operation column ... 1660
33.1.4 PREFIX instruction ... 1660
**33.2 Operation List ... 1661**

**CHAPTER 34 ELECTRICAL SPECIFICATIONS (GRADE L) ... 1679**

**34.1 Absolute Maximum Ratings ... 1680**
**34.2 Oscillator Characteristics ... 1682**
34.2.1 Main System Clock Oscillator Characteristics ... 1682
34.2.2 On-chip Oscillator Characteristics ... 1683

34.2.3 Subsystem Clock Oscillator Characteristics ..... 1684
34.2.4 PLL Circuit Characteristics ..... 1685
**34.3 DC Characteristics ..... 1686**
34.3.1 Pin Characteristics ..... 1686
34.3.2 Supply Current Characteristics ..... 1692
**34.4 AC Characteristics ..... 1697**
34.4.1 Basic Operation ..... 1697
**34.5 Peripheral Functions Characteristics ..... 1700**
34.5.1 Serial Array Unit ..... 1700
34.5.2 Serial Interface IICA ..... 1719
34.5.3 On-chip Debug (UART) ..... 1720
34.5.4 LIN/UART Module (RLIN3) UART Mode ..... 1720
**34.6 Analog Characteristics ..... 1721**
34.6.1 A/D Converter Characteristics ..... 1721
34.6.2 Temperatures Sensor Characteristics ..... 1725
34.6.3 D/A Converter Characteristics ..... 1725
34.6.4 Comparator Characteristics ..... 1725
34.6.5 POR Circuit Characteristics ..... 1726
34.6.6 LVD Circuit Characteristics ..... 1727
**34.7 Power Supply Voltage Rising Time ..... 1728**
**34.8 STOP Mode Memory Retention Characteristics ..... 1728**
**34.9 Flash Memory Programming Characteristics ..... 1729**

**CHAPTER 35 ELECTRICAL SPECIFICATIONS (GRADE K) ..... 1730**

**35.1 Absolute Maximum Ratings ..... 1731**
**35.2 Oscillator Characteristics ..... 1733**
35.2.1 Main System Clock Oscillator Characteristics ..... 1733
35.2.2 On-chip Oscillator Characteristics ..... 1734
35.2.3 Subsystem Clock Oscillator Characteristics ..... 1735
35.2.4 PLL Circuit Characteristics ..... 1736
**35.3 DC Characteristics ..... 1737**
35.3.1 Pin Characteristics ..... 1737
35.3.2 Supply Current Characteristics ..... 1743
**35.4 AC Characteristics ..... 1748**
35.4.1 Basic Operation ..... 1748
**35.5 Peripheral Functions Characteristics ..... 1751**
35.5.1 Serial Array Unit ..... 1751
35.5.2 Serial Interface IICA ..... 1770
35.5.3 On-chip Debug (UART) ..... 1771
35.5.4 LIN/UART Module (RLIN3) UART Mode ..... 1771
**35.6 Analog Characteristics ..... 1772**

35.6.1 A/D Converter Characteristics ... 1772
35.6.2 Temperatures Sensor Characteristics ... 1776
35.6.3 D/A Converter Characteristics ... 1776
35.6.4 Comparator Characteristics ... 1776
35.6.5 POR Circuit Characteristics ... 1777
35.6.6 LVD Circuit Characteristics ... 1778
**35.7 Power Supply Voltage Rising Time ... 1779**
**35.8 STOP Mode Memory Retention Characteristics ... 1779**
**35.9 Flash Memory Programming Characteristics ... 1780**

**CHAPTER 36 ELECTRICAL SPECIFICATIONS (GRADE Y) ... 1781**

**36.1 Absolute Maximum Ratings ... 1782**
**36.2 Oscillator Characteristics ... 1784**
36.2.1 Main System Clock Oscillator Characteristics ... 1784
36.2.2 On-chip Oscillator Characteristics ... 1785
36.2.3 Subsystem Clock Oscillator Characteristics ... 1786
36.2.4 PLL Circuit Characteristics ... 1787
**36.3 DC Characteristics ... 1788**
36.3.1 Pin Characteristics ... 1788
36.3.2 Supply Current Characteristics ... 1794
**36.4 AC Characteristics ... 1799**
36.4.1 Basic Operation ... 1799
**36.5 Peripheral Functions Characteristics ... 1802**
36.5.1 Serial Array Unit ... 1802
36.5.2 Serial Interface IICA ... 1821
36.5.3 On-chip Debug (UART) ... 1822
36.5.4 LIN/UART Module (RLIN3) UART Mode ... 1822
**36.6 Analog Characteristics ... 1823**
36.6.1 A/D Converter Characteristics ... 1823
36.6.2 Temperatures Sensor Characteristics ... 1827
36.6.3 D/A Converter Characteristics ... 1827
36.6.4 Comparator Characteristics ... 1827
36.6.5 POR Circuit Characteristics ... 1828
36.6.6 LVD Circuit Characteristics ... 1829
**36.7 Power Supply Voltage Rising Time ... 1830**
**36.8 STOP Mode Memory Retention Characteristics ... 1830**
**36.9 Flash Memory Programming Characteristics ... 1831**

**APPENDIX A RELATED PRODUCTS ... 1840**

**A.1 List of Analog and Power Devices ... 1840**

**APPENDIX B REVISION HISTORY ..... 1842**

**Major Revisions in This Edition ..... 1842**

# CHAPTER 1 OVERVIEW

## 1.1 Features

- Minimum instruction execution time can be changed from high speed (0.03125 μs: @ 32 MHz operation with high-speed on-chip oscillator clock or PLL clock) to ultra low-speed (66.6 μs: @ 15 kHz operation with low-speed on-chip oscillator clock)
- General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)
- ROM: 16 to 256 KB
- RAM: 1 to 20 KB
- Data flash memory: 4 KB/8 KB
- High-speed on-chip oscillator clock
  - Selectable from 32 MHz (Typ.), 24 MHz (Typ.), 16 MHz (Typ.), 12 MHz (Typ.), 8 MHz (Typ.), 4 MHz (Typ.), and 1 MHz (Typ.) (Selectable from 64 MHz (Typ.) and 48 MHz (Typ.) when using Timer RD)
- Low-speed on-chip oscillator clock: 15 kHz × 2 channels (one for WWDT and one for CPU and peripherals other than WWDT)
- On-chip PLL (×3, ×4, ×6, ×8)
- On-chip single-power-supply flash memory (with prohibition of block erase/writing function)
- Self-programming (with boot swap function/flash shield window function)
- On-chip debug function
- On-chip power-on-reset (POR) circuit and voltage detector (LVD)
- On-chip watchdog timer (operable with the dedicated low-speed on-chip oscillator clock)
- Multiply/divide/multiply & accumulate instructions are supported
  - 16 bits × 16 bits = 32 bits (Unsigned or signed)
  - 32 bits ÷ 32 bits = 32 bits (Unsigned)
  - 16 bits × 16 bits + 32 bits = 32 bits (Unsigned or signed)
- On-chip key interrupt function
- On-chip clock output/buzzer output controller
- On-chip BCD adjustment
- I/O ports: 16 to 92 (including one input-only pin)
- Timer
  - 16-bit timer array unit: 8 to 16 channels
  - 16-bit timer RD: 2 channels (six triangle-wave outputs; sawtooth wave/triangle-wave modulation)
  - 16-bit timer RJ: 1 channel
  - Watchdog timer: 1 channel
  - Real-time clock: 1 channel
- Serial interface
  - CSI
  - UART/UART (LIN-bus supported)
  - LIN module (master/slave supported)
  - $I^2C$/simplified $I^2C$
  - CAN interface (RS-CAN lite)
- 8/10-bit resolution A/D converter ($V_{DD}$ = 2.7 to 5.5 V): 4 to 31 channels
- DTC (Max. 52 sources)
- ELC (Max. 26 channels for event link source, Max. 9 channels for event link destination) **Note**
- Safety functions (CRC calculation, PLL lock detection, AD test, SFR guard, etc.)

- 8-bit D/A converter Note
- On-chip comparator: 1 channel (input pin: 4 channels) Note
- Power supply voltage: $V_{DD}$ = 2.7 to 5.5 V
- Operating ambient temperature: $T_A$ = -40 to +105ºC (grade L)/$T_A$ = -40 to +125°C (grade K)/$T_A$ = -40 to +150ºC (grade Y)

**Note** Only available in the RL78/F14.

### 1.1.1 Applications

General automotive electrical applications (motor control, door control, headlight control, etc.), motorcycle engine control

## 1.2 Product Lineup

**Table 1-1. RL78/F14 Lineup**

| Code Flash | Data Flash | RAM | Pin Count | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | 100 pins | 80 pins | 64 pins | 48 pins (QFN) | 48 pins (QFP) | 32 pins | 30 pins |
| 48 KB | 4 KB | 4 KB | — | — | — | R5F10PGD | R5F10PGD | R5F10PBD | R5F10PAD |
| 64 KB | | 6 KB | R5F10PPE | R5F10PME | R5F10PLE | R5F10PGE | R5F10PGE | R5F10PBE | R5F10PAE |
| 96 KB | | 8 KB | R5F10PPF | R5F10PMF | R5F10PLF | R5F10PGF | R5F10PGF | — | — |
| 128 KB | 8 KB | 10 KB | R5F10PPG | R5F10PMG | R5F10PLG | R5F10PGG | R5F10PGG | — | — |
| 192 KB | | 16 KB | R5F10PPH | R5F10PMH | R5F10PLH | R5F10PGH | R5F10PGH | — | — |
| 256 KB | | 20 KB | R5F10PPJ | R5F10PMJ | R5F10PLJ | R5F10PGJ | R5F10PGJ | — | — |

**Table 1-2. RL78/F13 (CAN and LIN incorporated) Lineup**

| Code Flash | Data Flash | RAM | Pin Count | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 80 pins | 64 pins | 48 pins (QFN) | 48 pins (QFP) | 32 pins | 30 pins |
| 32 KB | 4 KB | 2 KB | — | R5F10BLC | R5F10BGC | R5F10BGC | R5F10BBC | R5F10BAC |
| 48 KB | | 3 KB | — | R5F10BLD | R5F10BGD | R5F10BGD | R5F10BBD | R5F10BAD |
| 64 KB | | 4 KB | R5F10BME | R5F10BLE | R5F10BGE | R5F10BGE | R5F10BBE | R5F10BAE |
| 96 KB | | 6 KB | R5F10BMF | R5F10BLF | R5F10BGF | R5F10BGF | R5F10BBF | R5F10BAF |
| 128 KB | | 8 KB | R5F10BMG | R5F10BLG | R5F10BGG | R5F10BGG | R5F10BBG | R5F10BAG |

**Table 1-3. RL78/F13 (LIN incorporated) Lineup**

| Code Flash | Data Flash | RAM | Pin Count | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | 80 pins | 64 pins | 48 pins (QFN) | 48 pins (QFP) | 32 pins | 30 pins | 20 pins |
| 16 KB | 4 KB | 1 KB | — | — | R5F10AGA | R5F10AGA | R5F10ABA | R5F10AAA | R5F10A6A |
| 32 KB | | 2 KB | — | R5F10ALC | R5F10AGC | R5F10AGC | R5F10ABC | R5F10AAC | R5F10A6C |
| 48 KB | | 3 KB | — | R5F10ALD | R5F10AGD | R5F10AGD | R5F10ABD | R5F10AAD | R5F10A6D |
| 64 KB | | 4 KB | R5F10AME | R5F10ALE | R5F10AGE | R5F10AGE | R5F10ABE | R5F10AAE | R5F10A6E |
| 96 KB | | 6 KB | R5F10AMF | R5F10ALF | R5F10AGF | R5F10AGF | — | — | — |
| 128 KB | | 8 KB | R5F10AMG | R5F10ALG | R5F10AGG | R5F10AGG | — | — | — |

## 1.3 Function Overview

### 1.3.1 RL78/F14 Functions List

**Table 1-4. RL78/F14 Functions List (1/2)**

| Series Name | | | R5F10PP | R5F10PM | R5F10PL | R5F10PG | R5F10PB | R5F10PA |
|---|---|---|---|---|---|---|---|---|
| | | Pin Count | 100 pins | 80 pins | 64 pins | 48 pins | 32 pins | 30 pins |
| Code flash | | | 64 to 256KB | | | 48 to 256KB | 48KB, 64KB | |
| Data flash | | | 8KB/4KB | | | | 4KB | |
| RAM | | | 6 to 20KB | | | 4 to 20KB | 48KB, 64KB | |
| Supply voltage range | | | 2.7 V to 5.5 V | | | | | |
| Maximum operation frequency | | | 32 MHz (grade L), 24 MHz (grade K, grade Y) | | | | | |
| System clock | Main system clock oscillator | Crystal/ceramic/ square wave | 1 to 20 MHz (operating at 2.7 V to 5.5 V) | | | | | |
| | High-speed on-chip oscillator | Normal high accuracy | 32 MHz (typ.) | | | | | |
| | Low-speed on-chip oscillator | For low-speed operation | 15 kHz (typ.) | | | | | |
| | Subsystem clock oscillator | | 32.768 kHz Note 7 | | | | None | |
| | PLL | | PLL multiplication factor: ×3/×4/×6/×8 | | | | | |
| Clock for peripherals | Low-speed on-chip oscillator | For peripherals other than WDT | 15 kHz (typ.) | | | | | |
| | | For WDT | 15 kHz (typ.) | | | | | |
| POR | | When power supply is rising | 1.56 V (typ.) | | | | | |
| | | When power supply is falling | 1.55 V (typ.) | | | | | |
| LVD | $V_{DD}$ voltage detection | When power supply is rising | 2.81 V (typ.) to 4.74 V (typ.) (in 6 steps) | | | | | |
| | | When power supply is falling | 2.75 V (typ.) to 4.64 V (typ.) (in 6 steps) | | | | | |
| Safety functions | WWDT (window watchdog timer) | | Yes | | | | | |
| | Illegal instruction execution detection function | | Yes | | | | | |
| | Flash memory CRC operation function | | Yes | | | | | |
| | RAM1 bit error correction function | | Yes | | | | | |
| | RAM2 bit error detection function | | Yes | | | | | |
| | Invalid memory access detection function | | Yes | | | | | |
| | Frequency detection function | | Yes | | | | | |
| | Clock monitor function | | Yes | | | | | |
| | Stack pointer monitor function | | Yes | | | | | |
| | I/O port output signal level detection function | | Yes | | | | | |
| | A/D test function | | Yes | | | | | |
| I/O ports | Input/Output | CMOS | 86ch | 68ch | 52ch | 38ch | 25ch | 23ch |
| | Output | CMOS | 1ch | | | | None | |
| | Input | Shared with oscillator pins | 4ch Note 7 | | | | 2ch | |
| | | Input only | 1ch | | | | | |
| Power supply pins | For internal circuits | | $V_{DD}$, $V_{SS}$, REGC | | | | | |
| | For I/O ports | | $EV_{DD0}$, $EV_{SS0}$ $EV_{DD1}$, $EV_{SS1}$ | $EV_{DD0}$, $EV_{SS0}$ | | None | | |
| | For analog circuits (AD, DA, COMP) | | $V_{DD}$, $V_{SS}$ ($AV_{REFP}$, $AV_{REFM}$: For AD) | | | | | |
| Multiply/divide and multiply-accumulate functions | Multiply | | 16 bits × 16 bits (signed) | | | | | |
| | | | 16 bits × 16 bits (unsigned) | | | | | |
| | Divide | | 32 bits ÷ 32 bits (unsigned) | | | | | |
| | Multiply-accumulate | | 16 bits × 16 bits + 32 bits (signed) | | | | | |
| | | | 16 bits × 16 bits + 32 bits (unsigned) | | | | | |
| | Arithmetic instructions (extended instruction set) | | Yes | | | | | |
| Vectored interrupt sources | External | Products with at least 128 Kbytes of code flash memory | 16ch Notes 4, 6 | 16ch Notes 4, 6 | 15ch Notes 3, 6 | 14ch Note 2 | 9ch Note 1 | |
| | | Products with up to 96 Kbytes of code flash memory | | 14ch Notes 3, 5 | 14ch Notes 3, 5 | 13ch Note 2 | | |
| | Internal | Products with at least 128 Kbytes of code flash memory | 48ch Note 4 | 48ch Note 4 | 48ch Note 3 | 48ch Note 2 | 41ch Note 1 | 40ch Note 1 |
| | | Products with up to 96 Kbytes of code flash memory | | 41ch Note 3 | 41ch Note 3 | 41ch Note 2 | | |
| Key return detection | | | 8ch | | | | 6ch | 8ch |
| DTC | | | 44 sources | 44 sources/38 sources | | | 37 sources | |
| Timer | TAU | | 16 bits (8ch × 2) | 16 bits (8ch × 2/8ch + 4ch) | | | 16 bits (8ch + 4ch) | |
| | RTC | | 1ch | | | | | |
| | Timer RJ | | 16 bits × 1 | | | | | |
| | Timer RD | | 16 bits × 2 | | | | | |
| Serial I/F | CSI/simplified $I^2C$ /UART | | 4ch/4ch/2ch | | | | 3ch/3ch/2ch | |
| | | SPI | Yes | | | | | |
| | Multimaster $I^2C$ | | 1ch | | | | | None |
| | LIN/UART module (RLIN3) | | 2ch | 2ch/1ch | | | 1ch | |
| | CAN interface (RS-CAN lite) | | 1ch | | | | | |

(**Notes** and **Caution** are listed on the next page.)

**Table 1-4. RL78/F14 Functions List (2/2)**

| Series Name | | R5F10PP | R5F10PM | R5F10PL | R5F10PG | R5F10PB | R5F10PA |
|---|---|---|---|---|---|---|---|
| | Pin Count | 100 pins | 80 pins | 64 pins | 48 pins | 32 pins | 30 pins |
| A/D converter 10-bit SAR | $V_{DD}$ | 24ch | 18ch/16ch | 17ch/16ch | 13ch | 8ch | 10ch |
| | $EV_{DD}$ | 7ch | 7ch/4ch | 3ch | 5ch/2ch | 2ch | |
| | Internal | 2ch | | | | | |
| D/A converter | 8-bit | 1ch | | | | | |
| Comparator | | 1ch | | | | | |
| ELC | | Link source: 26ch<br>Link destination: 9ch | Link source: 26ch/20ch<br>Link destination: 9ch/7ch | | | Link source: 20ch<br>Link destination: 7ch | |
| PCLBUZ | | 1ch | | | | None | |
| Self-programming | | Yes | | | | | |
| On-chip debug | Trace | Yes | | | | | |
| | Hot plug-in | Yes | | | | | |
| Option byte | | Yes | | | | | |

**Notes 1.** The following pairs of internal and external sources are each counted as a single source in this number: INTP4 and INTSPM, INTP5 and INTCMP0.

**2.** The following pairs of internal and external sources are each counted as a single source in this number: INTP4 and INTSPM, INTP5 and INTCMP0, INTP6 and INTTM11H, INTP7 and INTTM13H, INTP8 and INTRTC, INTP9 and INTTM01H.

**3.** The following pairs of internal and external sources are each counted as a single source in this number: INTP4 and INTSPM, INTP5 and INTCMP0, INTP6 and INTTM11H, INTP7 and INTTM13H, INTP8 and INTRTC, INTP9 and INTTM01H, INTP10 and INTTM03H.

**4.** The following pairs of internal and external sources are each counted as a single source in this number: INTP4 and INTSPM, INTP5 and INTCMP0, INTP6 and INTTM11H, INTP7 and INTTM13H, INTP8 and INTRTC, INTP9 and INTTM01H, INTP10 and INTTM03H, INTP13 and INTCLM.

**5.** INTP11 and INTLIN0WUP are counted as a single source because using them at the same time is not possible.

**6.** Both sources in the following pairs are counted as a single source in this number: INTP11 and INTLIN0WUP, INTP12 and INTLIN1WUP.

**7.** Do not use the XT1 and XT2 pin functions in grade-Y products.

**Caution For details, see 1.5 Pin Configurations.**

### 1.3.2 RL78/F13 (CAN and LIN incorporated) Functions List

**Table 1-5. RL78/F13 (CAN and LIN incorporated) Functions List**

| Series Name | | | R5F10BM | R5F10BL | R5F10BG | R5F10BB | R5F10BA |
|---|---|---|---|---|---|---|---|
| | | Pin Count | 80 pins | 64 pins | 48 pins | 32 pins | 30 pins |
| Code flash | | | 64 to 128KB | 32 to 128KB | | | |
| Data flash | | | 4KB | | | | |
| RAM | | | 4 to 8KB | 2 to 8KB | | | |
| Supply voltage range | | | 2.7 V to 5.5 V | | | | |
| Maximum operation frequency | | | 32 MHz (grade L), 24 MHz (grade K, grade Y) | | | | |
| System clock | Main system clock oscillator | Crystal/ceramic/ square wave | 1 to 20 MHz (operating at 2.7 V to 5.5 V) | | | | |
| | High-speed on-chip oscillator | Normal high accuracy | 32 MHz (typ.) | | | | |
| | Low-speed on-chip oscillator | For low-speed operation | 15 kHz (typ.) | | | | |
| | Subsystem clock oscillator | | 32.768 kHz[Note 5] | | | None | |
| | PLL | | PLL multiplication factor: ×3/×4/×6/×8 | | | | |
| Clock for peripherals | Low-speed on-chip oscillator | For peripherals other than WDT | 15 kHz (typ.) | | | | |
| | | For WDT | 15 kHz (typ.) | | | | |
| POR | | When power supply is rising | 1.56 V (typ.) | | | | |
| | | When power supply is falling | 1.55 V (typ.) | | | | |
| LVD | $V_{DD}$ voltage detection | When power supply is rising | 2.81 V (typ.) to 4.74 V (typ.) (in 6 steps) | | | | |
| | | When power supply is falling | 2.75 V (typ.) to 4.64 V (typ.) (in 6 steps) | | | | |
| Safety functions | WWDT (window watchdog timer) | | Yes | | | | |
| | Illegal instruction execution detection function | | Yes | | | | |
| | Flash memory CRC operation function | | Yes | | | | |
| | RAM1 bit error correction function | | Yes | | | | |
| | RAM2 bit error detection function | | Yes | | | | |
| | Invalid memory access detection function | | Yes | | | | |
| | Frequency detection function | | Yes | | | | |
| | Clock monitor function | | Yes | | | | |
| | Stack pointer monitor function | | Yes | | | | |
| | I/O port output signal level detection function | | Yes | | | | |
| | A/D test function | | Yes | | | | |
| I/O ports | Input/Output | CMOS | 68ch | 52ch | 38ch | 25ch | 23ch |
| | Output | CMOS | 1ch | | | None | |
| | Input | Shared with oscillator pins | 4ch[Note 5] | | | 2ch | |
| | | Input only | 1ch | | | | |
| Power supply pins | For internal circuits | | $V_{DD}$, $V_{SS}$, REGC | | | | |
| | For I/O ports | | $EV_{DD0}$, $EV_{SS0}$ | | None | | |
| | For analog circuits (AD, DA, COMP) | | $V_{DD}$, $V_{SS}$ ($AV_{REFP}$, $AV_{REFM}$: For AD) | | | | |
| Multiply/divide and multiply-accumulate functions | Multiply | | 16 bits × 16 bits (signed) | | | | |
| | | | 16 bits × 16 bits (unsigned) | | | | |
| | Divide | | 32 bits ÷ 32 bits (unsigned) | | | | |
| | Multiply-accumulate | | 16 bits × 16 bits + 32 bits (signed) | | | | |
| | | | 16 bits × 16 bits + 32 bits (unsigned) | | | | |
| | Arithmetic instructions (extended instruction set) | | Yes | | | | |
| Vectored interrupt sources | External | | 14ch [Notes 3, 4] | | 13ch [Note 2] | 9ch [Note 1] | |
| | Internal | | 40ch [Note 3] | | 40ch [Note 2] | 40ch [Note 1] | 39ch [Note 1] |
| Key return detection | | | 8ch | | | 6ch | 8ch |
| DTC | | | 37 sources | | | 36 sources | |
| Timer | TAU | | 16 bits (8ch + 4ch) | | | | |
| | RTC | | 1ch | | | | |
| | Timer RJ | | 16 bits × 1 | | | | |
| | Timer RD | | 16 bits × 2 | | | | |
| Serial I/F | CSI/simplified $I^2C$ /UART | | 4ch/4ch/2ch | | | 3ch/3ch/2ch | |
| | | SPI | Yes | | | | |
| | Multimaster $I^2C$ | | 1ch | | | | None |
| | LIN/UART module (RLIN3) | | 1ch | | | | |
| | CAN interface (RS-CAN lite) | | 1ch | | | | |
| A/D converter 10-bit SAR | $V_{DD}$ | | 16ch | 16ch | 13ch | 8ch | 10ch |
| | $EV_{DD}$ | | 4ch | 3ch | 2ch | | |
| | Internal | | 2ch | | | | |
| D/A converter | 8-bit | | None | | | | |
| Comparator | | | None | | | | |
| ELC | | | None | | | | |
| PCLBUZ | | | 1ch | | | None | |
| Self-programming | | | Yes | | | | |
| On-chip debug | Trace | | Yes | | | | |
| | Hot plug-in | | Yes | | | | |
| Option byte | | | Yes | | | | |

(**Notes** and **Caution** are listed on the next page.)

**Notes 1.** The following pairs of internal and external sources are each counted as a single source in this number: INTP4 and INTSPM.

**2.** The following pairs of internal and external sources are each counted as a single source in this number: INTP4 and INTSPM, INTP6 and INTTM11H, INTP7 and INTTM13H, INTP8 and INTRTC, INTP9 and INTTM01H.

**3.** The following pairs of internal and external sources are each counted as a single source in this number: INTP4 and INTSPM, INTP6 and INTTM11H, INTP7 and INTTM13H, INTP8 and INTRTC, INTP9 and INTTM01H, INTP10 and INTTM03H.

**4.** INTP11 and INTLIN0WUP are counted as a single source because using them at the same time is not possible.

**5.** Do not use the XT1 and XT2 pin functions in grade-Y products.

**Caution For details, see 1.5 Pin Configurations.**

### 1.3.3 RL78/F13 (LIN incorporated) Functions List

**Table 1-6. RL78/F13 (LIN incorporated) Functions List (1/2)**

| Series Name | | | R5F10AM | R5F10AL | R5F10AG | R5F10AB | R5F10AA | R5F10A6 |
|---|---|---|---|---|---|---|---|---|
| | | Pin Count | 80 pins | 64 pins | 48 pins | 32 pins | 30 pins | 20 pins |
| Code flash | | | 64 to 128KB | 32 to 128KB | 16 to 128KB | 16 to 64KB | | |
| Data flash | | | 4KB | | | | | |
| RAM | | | 4 to 8KB | 2 to 8KB | 1 to 8KB | 1 to 4KB | | |
| Supply voltage range | | | 2.7 V to 5.5 V | | | | | |
| Maximum operation frequency | | | 32 MHz (grade L), 24 MHz (grade K, grade Y) | | | | | |
| System clock | Main system clock oscillator | Crystal/ceramic/ square wave | 1 to 20 MHz (operating at 2.7 V to 5.5 V) | | | | | |
| | High-speed on-chip oscillator | Normal high accuracy | 32 MHz (typ.) | | | | | |
| | Low-speed on-chip oscillator | For low-speed operation | 15 kHz (typ.) | | | | | |
| | Subsystem clock oscillator | | 32.768 kHz Note 6 | | | None | | |
| | PLL | | PLL multiplication factor: ×3/×4/×6/×8 | | | | | |
| Clock for peripherals | Low-speed on-chip oscillator | For peripherals other than WDT | 15 kHz (typ.) | | | | | |
| | | For WDT | 15 kHz (typ.) | | | | | |
| POR | | When power supply is rising | 1.56 V (typ.) | | | | | |
| | | When power supply is falling | 1.55 V (typ.) | | | | | |
| LVD | $V_{DD}$ voltage detection | When power supply is rising | 2.81 V (typ.) to 4.74 V (typ.) (in 6 steps) | | | | | |
| | | When power supply is falling | 2.75 V (typ.) to 4.64 V (typ.) (in 6 steps) | | | | | |
| Safety functions | WWDT (window watchdog timer) | | Yes | | | | | |
| | Illegal instruction execution detection function | | Yes | | | | | |
| | Flash memory CRC operation function | | Yes | | | | | |
| | RAM1 bit error correction function | | Yes | | | | | |
| | RAM2 bit error detection function | | Yes | | | | | |
| | Invalid memory access detection function | | Yes | | | | | |
| | Frequency detection function | | Yes | | | | | |
| | Clock monitor function | | Yes | | | | | |
| | Stack pointer monitor function | | Yes | | | | | |
| | I/O port output signal level detection function | | Yes | | | | | |
| | A/D test function | | Yes | | | | | |
| I/O ports | Input/Output | CMOS | 68ch | 52ch | 38ch | 25ch | 23ch | 13ch |
| | Output | CMOS | 1ch | | | None | | |
| | Input | Shared with oscillator pins | 4ch Note 6 | | | 2ch | | |
| | | Input only | 1ch | | | | | |
| Power supply pins | For internal circuits | | $V_{DD}$, $V_{SS}$, REGC | | | | | |
| | For I/O ports | | $EV_{DD0}$, $EV_{SS0}$ | | None | | | |
| | For analog circuits (AD, DA, COMP) | | $V_{DD}$, $V_{SS}$ ($AV_{REFP}$, $AV_{REFM}$: For AD) | | | | | |
| Multiply/divide and multiply-accumulate functions | Multiply | | 16 bits × 16 bits (signed) | | | | | |
| | | | 16 bits × 16 bits (unsigned) | | | | | |
| | Divide | | 32 bits ÷ 32 bits (unsigned) | | | | | |
| | Multiply-accumulate | | 16 bits × 16 bits + 32 bits (signed) | | | | | |
| | | | 16 bits × 16 bits + 32 bits (unsigned) | | | | | |
| | Arithmetic instructions (extended instruction set) | | Yes | | | | | |
| Vectored interrupt sources | External | Products with at least 96 Kbytes of code flash memory | 13ch Note 4, 5 | | 12ch Note 3 | — | | |
| | | Products with up to 64 Kbytes of code flash memory | 13ch Note 4, 5 | 10ch Note 2 | | 8ch Note 2 | | 7ch Note 2 |
| | Internal | Products with at least 96 Kbytes of code flash memory | 35ch Note 4 | | 35ch Note 3 | — | | |
| | | Products with up to 64 Kbytes of code flash memory | 35ch Note 4 | 26ch Note 2 | | | | |
| Key return detection | | | 8ch | | | 6ch | 8ch | 2ch |
| DTC | Products with at least 96 Kbytes of code flash memory | | 36 sources | | | — | | |
| | Products with up to 64 Kbytes of code flash memory | | 36 sources | 30 sources | | 29 sources | | 28 sources |

(**Notes** and **Caution** are listed on the next page.)

**Table 1-6. RL78/F13 (LIN incorporated) Functions List (2/2)**

| Series Name | | | R5F10AM | R5F10AL | R5F10AG | R5F10AB | R5F10AA | R5F10A6 |
|---|---|---|---|---|---|---|---|---|
| | | Pin Count | 80 pins | 64 pins | 48 pins | 32 pins | 30 pins | 20 pins |
| TAU | Products with at least 96 Kbytes of code flash memory | | 16 bits (8ch + 4ch) | | | — | | |
| | Products with up to 64 Kbytes of code flash memory | | 16 bits (8ch + 4ch) | 16 bits (8ch) | | | | |
| Timer | RTC | | 1ch | | | | | |
| | Timer RJ | | 16 bits × 1 Note 1 | | | | | |
| | Timer RD | | 16 bits × 2 | | | | | |
| Serial I/F | CSI/simplified $I^2C$ /UART | Products with at least 96 Kbytes of code flash memory | 4ch/4ch/2ch | | | — | | |
| | | Products with up to 64 Kbytes of code flash memory | 4ch/4ch/2ch | 2ch/2ch/1ch | | | | |
| | SPI | | Yes | | | | | |
| | Multimaster $I^2C$ | Products with at least 96 Kbytes of code flash memory | 1ch | | | — | | |
| | | Products with up to 64 Kbytes of code flash memory | 1ch | None | | | | |
| | LIN/UART module (RLIN3) | | 1ch | | | | | |
| | CAN interface (RS-CAN lite) | | None | | | | | |
| A/D converter 10-bit SAR | $V_{DD}$ | Products with at least 96 Kbytes of code flash memory | 16ch | 16ch | 13ch | — | | |
| | | Products with up to 64 Kbytes of code flash memory | 16ch | 12ch | 12ch | 8ch | 10ch | 4ch |
| | $EV_{DD}$ | Products with at least 96 Kbytes of code flash memory | 4ch | 3ch | 2ch | — | | |
| | | Products with up to 64 Kbytes of code flash memory | 4ch | None | | | | |
| | Internal | | 2ch | | | | | |
| D/A converter | 8-bit | | None | | | | | |
| Comparator | | | None | | | | | |
| ELC | | | None | | | | | |
| PCLBUZ | | | 1ch | | | None | | |
| Self-programming | | | Yes | | | | | |
| On-chip debug | Trace | | Yes | | | | | |
| | Hot plug-in | | Yes | | | | | |
| Option byte | | | Yes | | | | | |

**Notes 1.** The 20-pin products do not have TRJIO0 and TRJO0 pins.

**2.** The following pairs of internal and external sources are each counted as a single source in this number: INTP4 and INTSPM.

**3.** The following pairs of internal and external sources are each counted as a single source in this number: INTP4 and INTSPM, INTP6 and INTTM11H, INTP7 and INTTM13H, INTP8 and INTRTC, INTP9 and INTTM01H.

**4.** The following pairs of internal and external sources are each counted as a single source in this number: INTP4 and INTSPM, INTP6 and INTTM11H, INTP7 and INTTM13H, INTP8 and INTRTC, INTP9 and INTTM01H, INTP10 and INTTM03H.

**5.** NTP11 and INTLIN0WUP are counted as a single source because using them at the same time is not possible.

**6.** Do not use the XT1 and XT2 pin functions in grade-Y products.

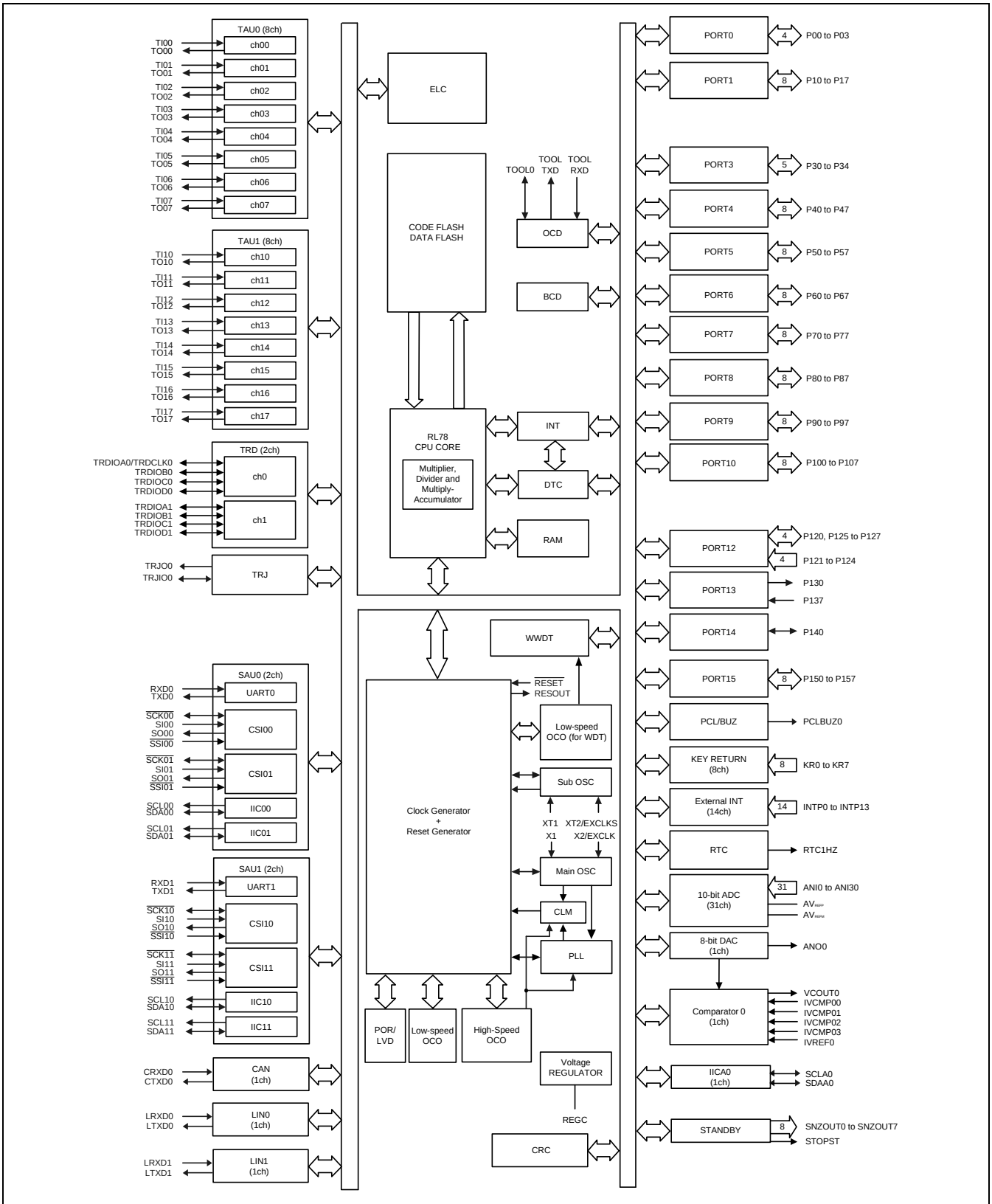
**Caution For details, see 1.5 Pin Configurations.**

## 1.4 Block Diagram

### 1.4.1 RL78/F14: Block Diagram of R5F10PPn (n = E, F, G, H, J) 100-pin Products

<R>

**Figure 1-1. Block Diagram**

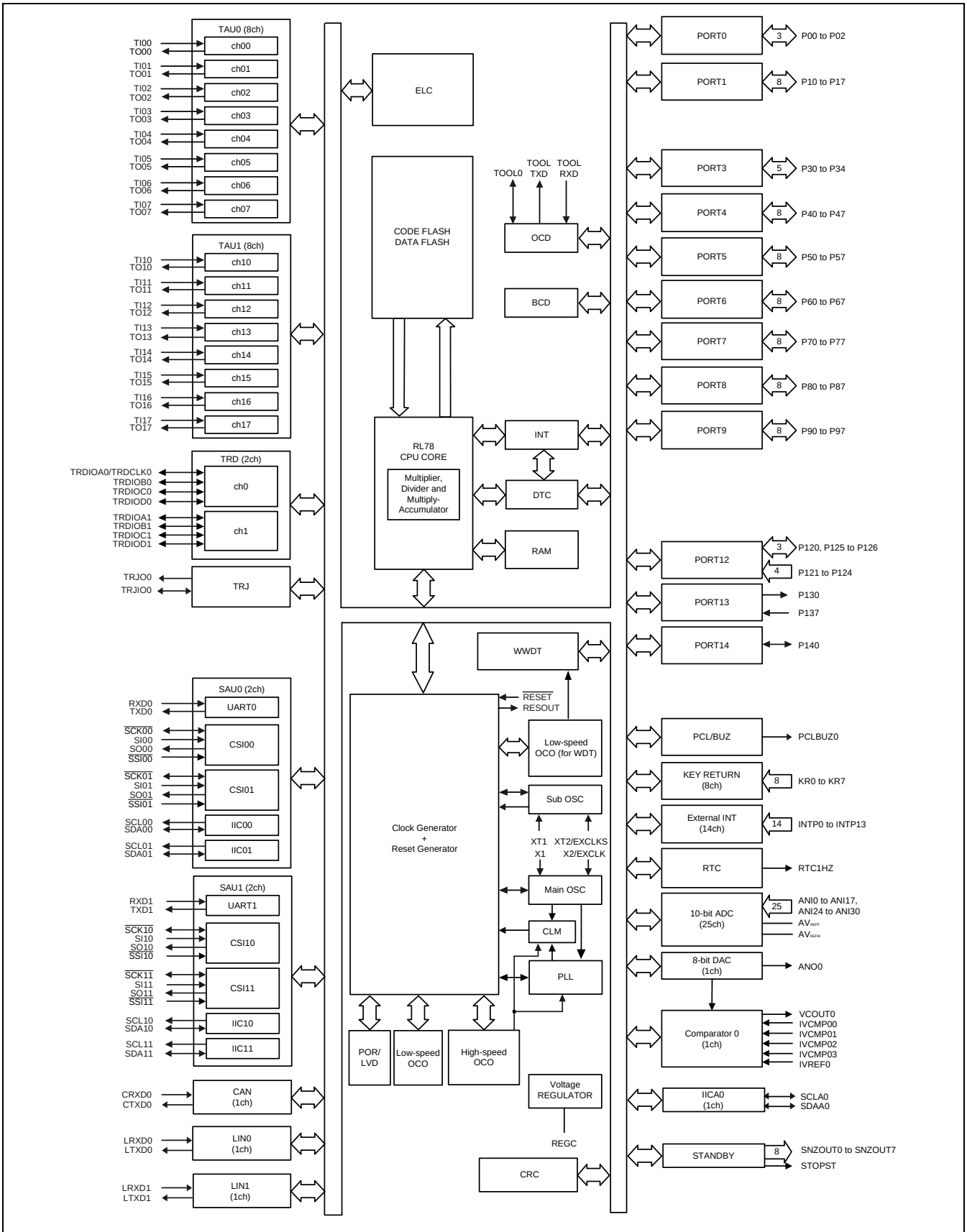



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.2 RL78/F14: Block Diagram of R5F10PMn (n = G, H, J) 80-pin Products

<R>

**Figure 1-2. Block Diagram**

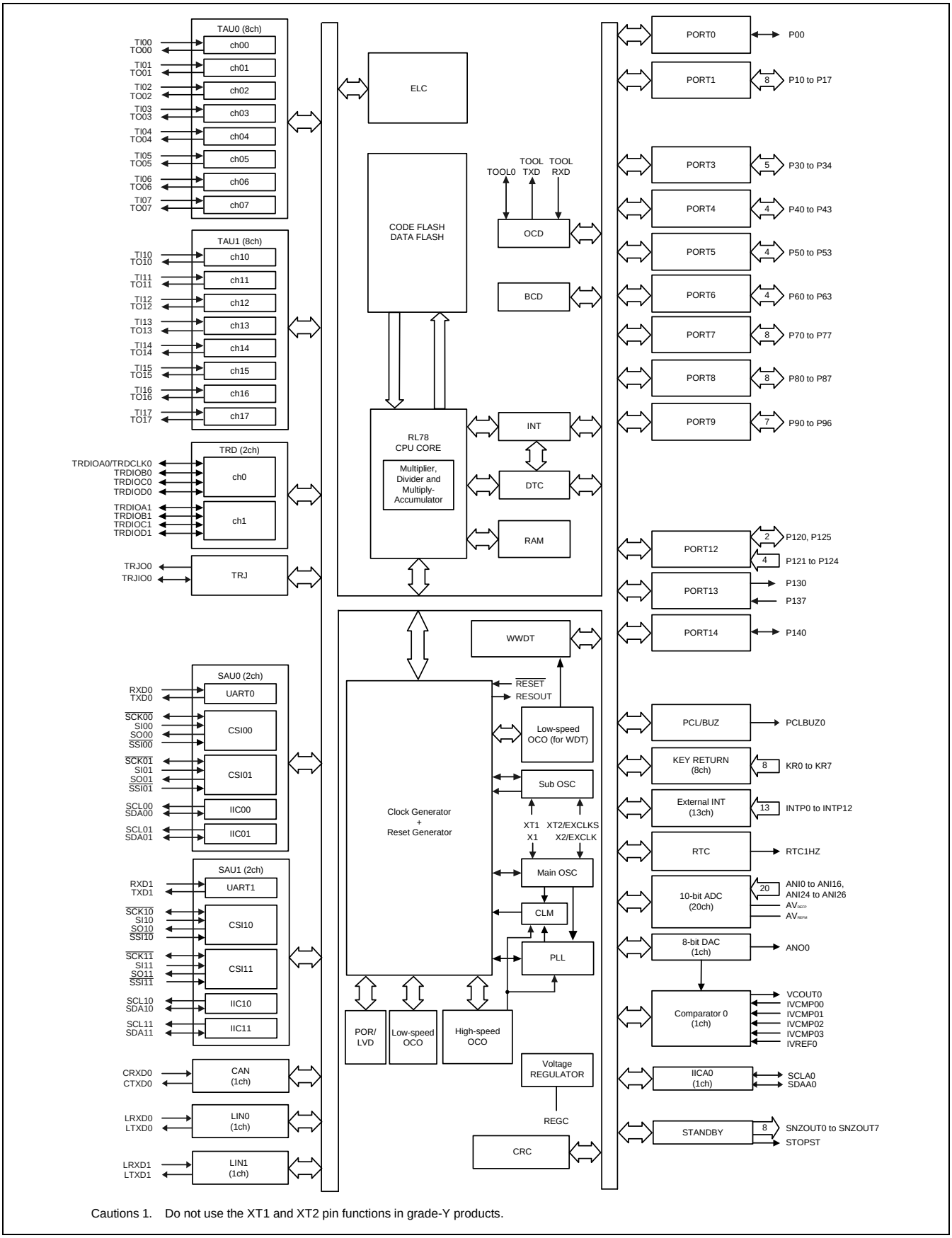



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.3 RL78/F14: Block Diagram of R5F10PLn (n = G, H, J) 64-pin Products

<R>

**Figure 1-3. Block Diagram**

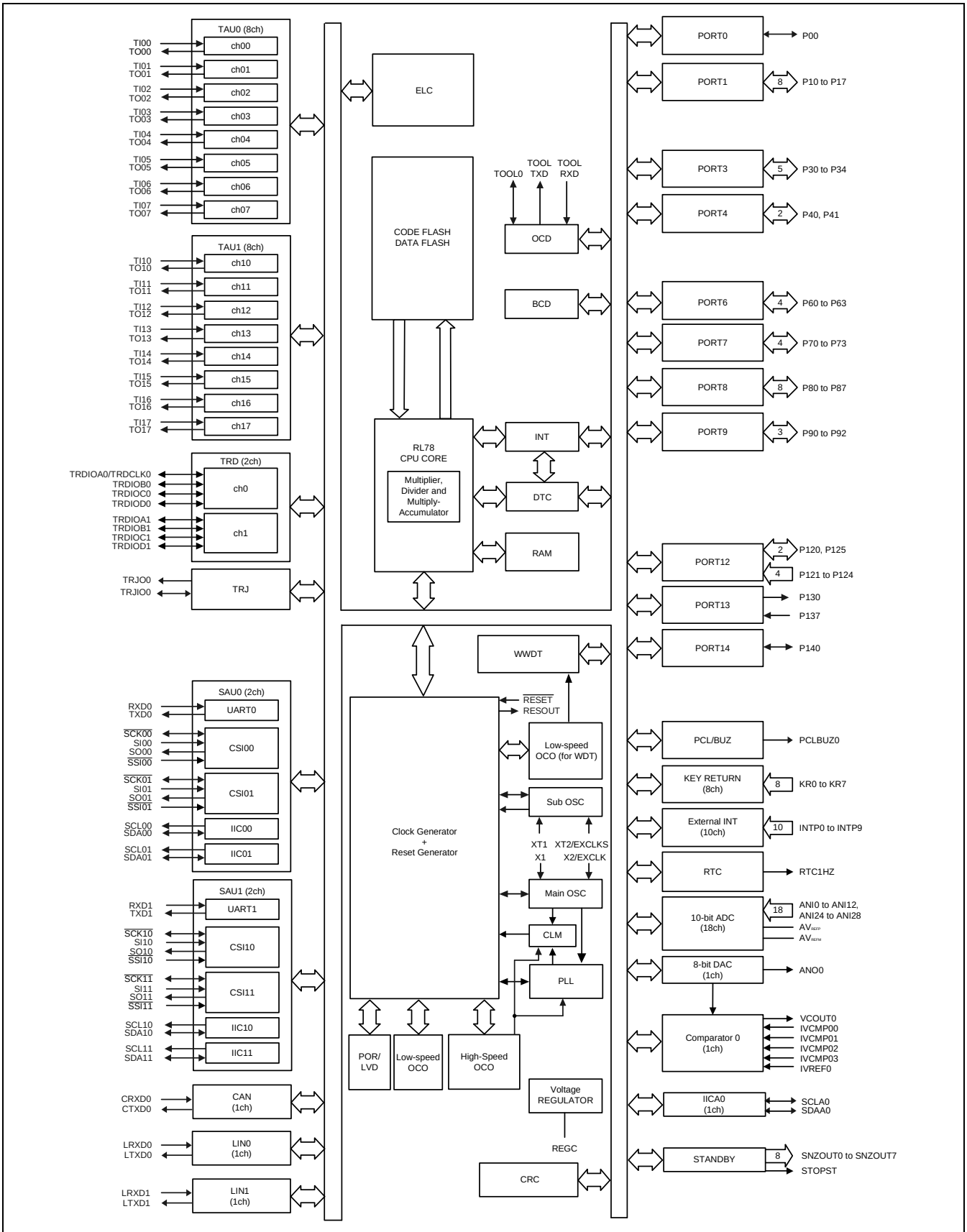



Cautions 1. Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.4 RL78/F14: Block Diagram of R5F10PGn (n = G, H, J) 48-pin Products

<R>

**Figure 1-4. Block Diagram**

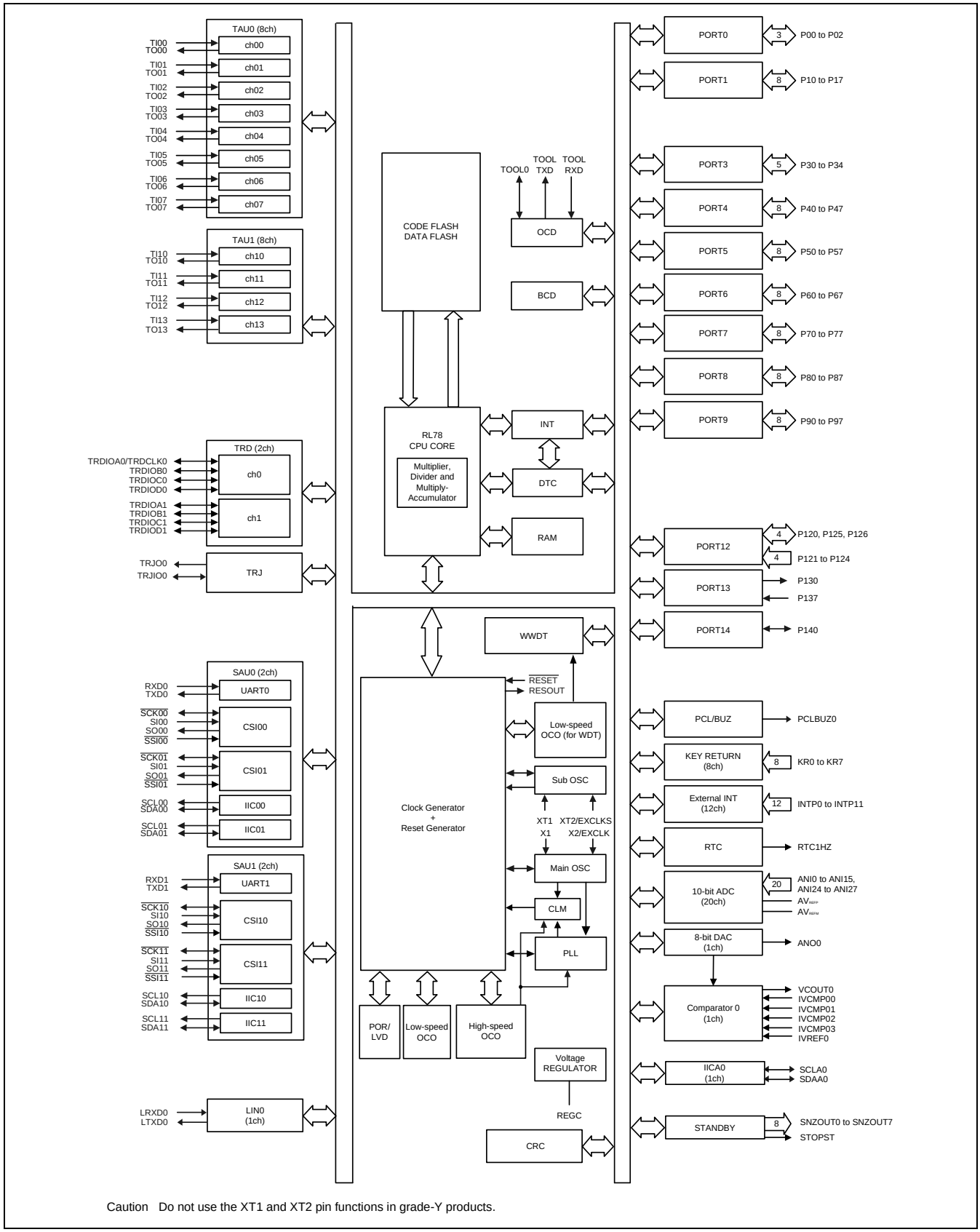



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.5 RL78/F14: Block Diagram of R5F10PMn (n = E, F) 80-pin Products

<R>

**Figure 1-5. Block Diagram**

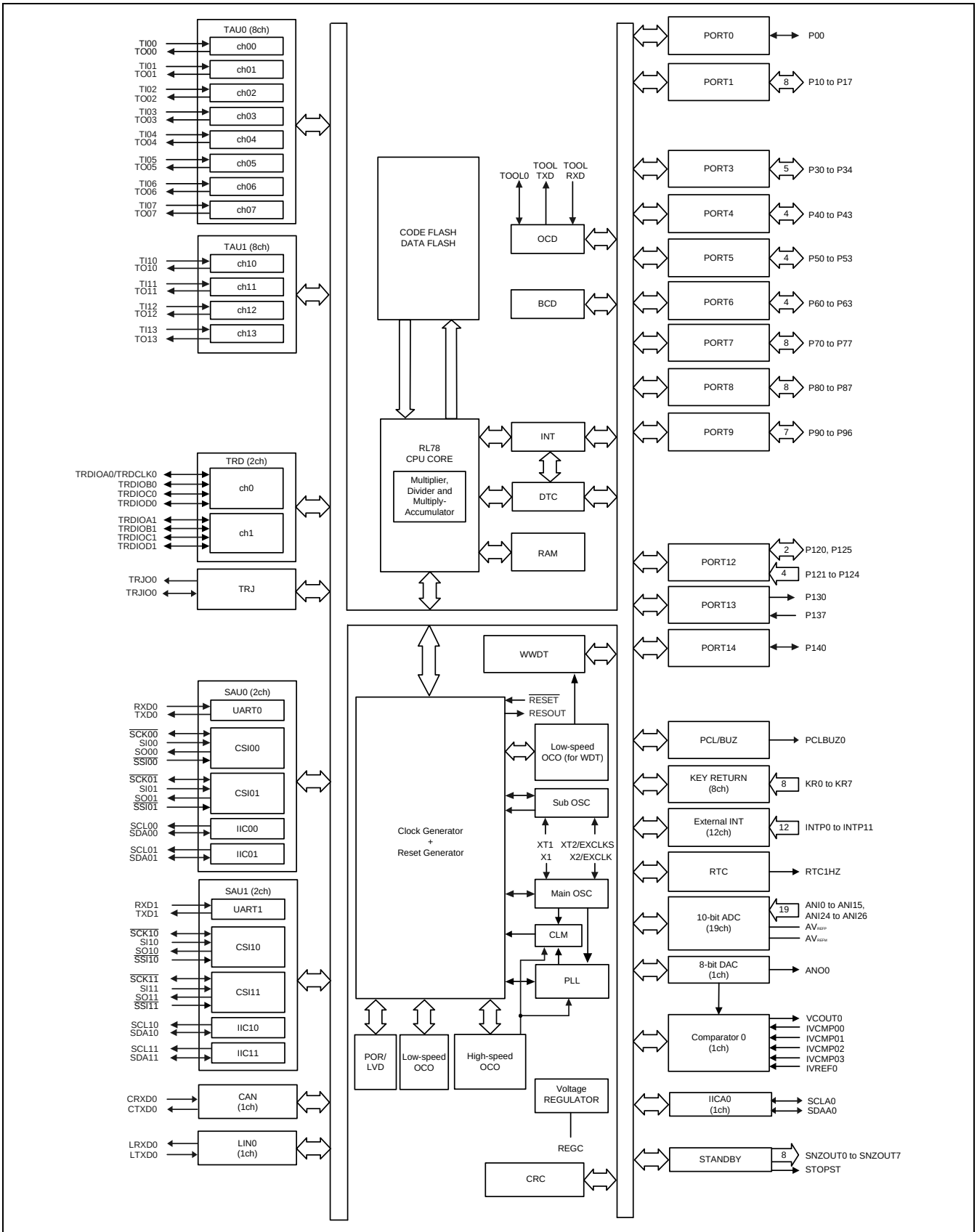



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.6 RL78/F14: Block Diagram of R5F10PLn (n = E, F) 64-pin Products

<R>

**Figure 1-6. Block Diagram**

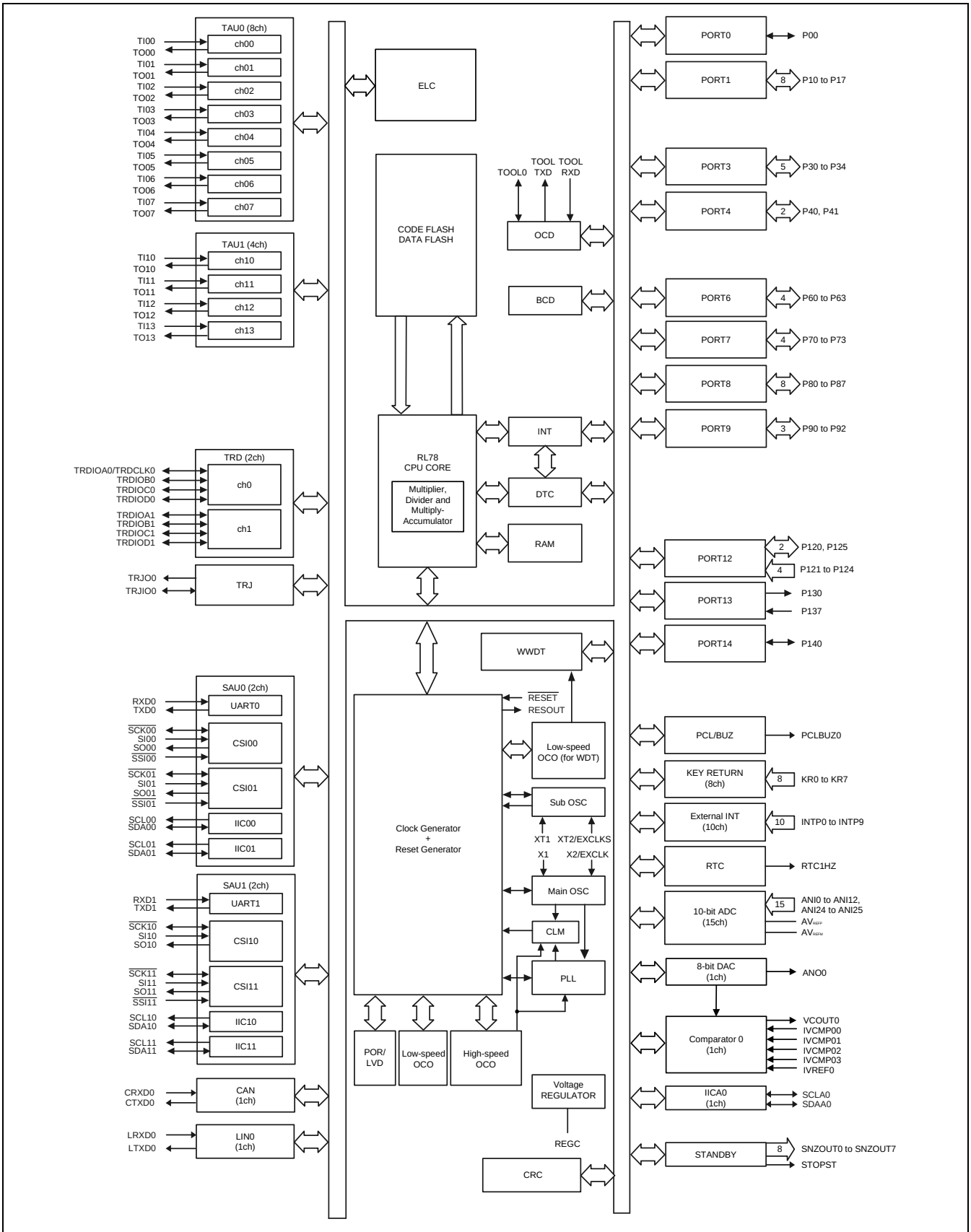



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.7 RL78/F14: Block Diagram of R5F10PGn (n = D, E, F) 48-pin Products

<R>

**Figure 1-7. Block Diagram**

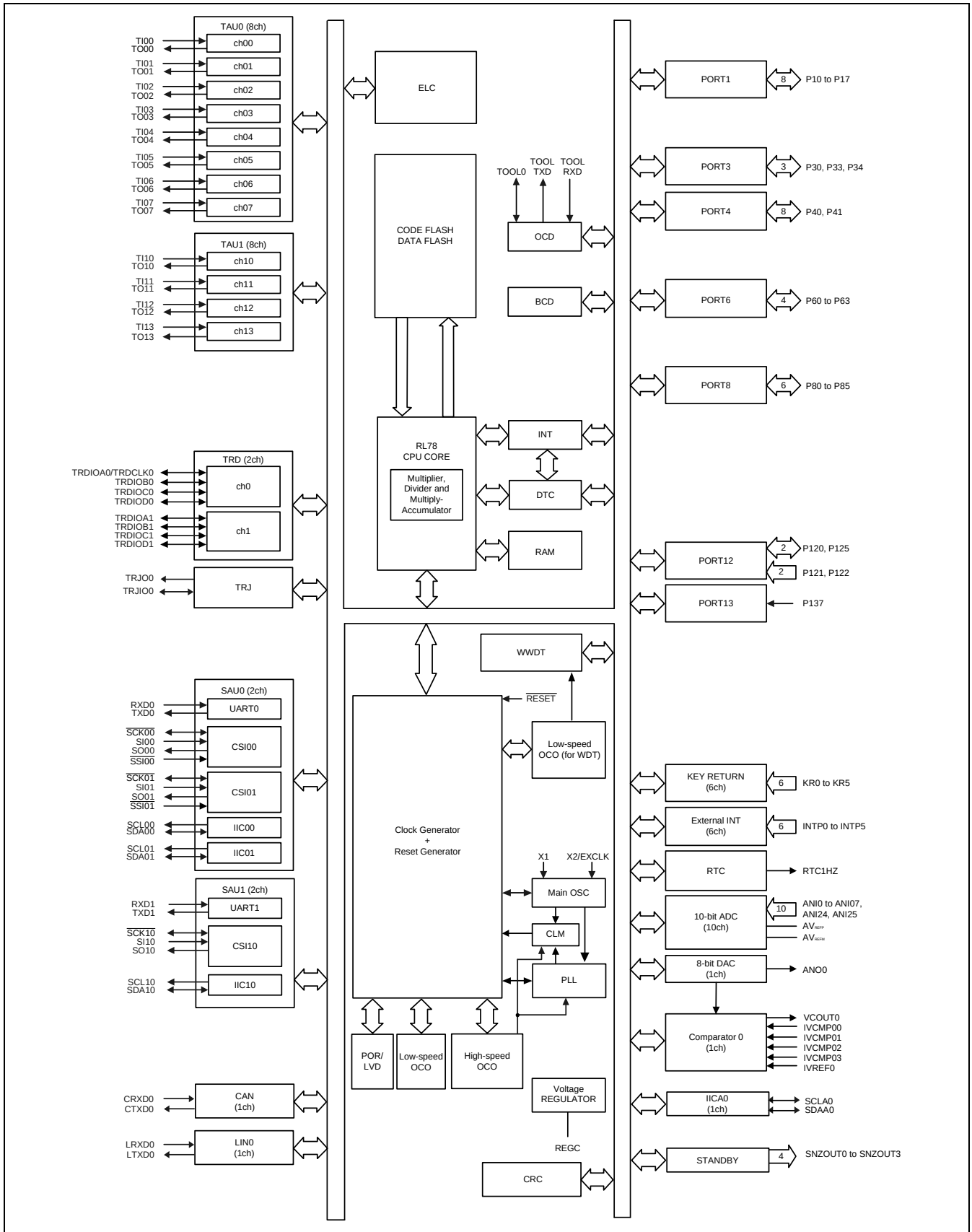



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.8 RL78/F14: Block Diagram of R5F10PBn (n = D, E) 32-pin Products

<R>

**Figure 1-8. Block Diagram**

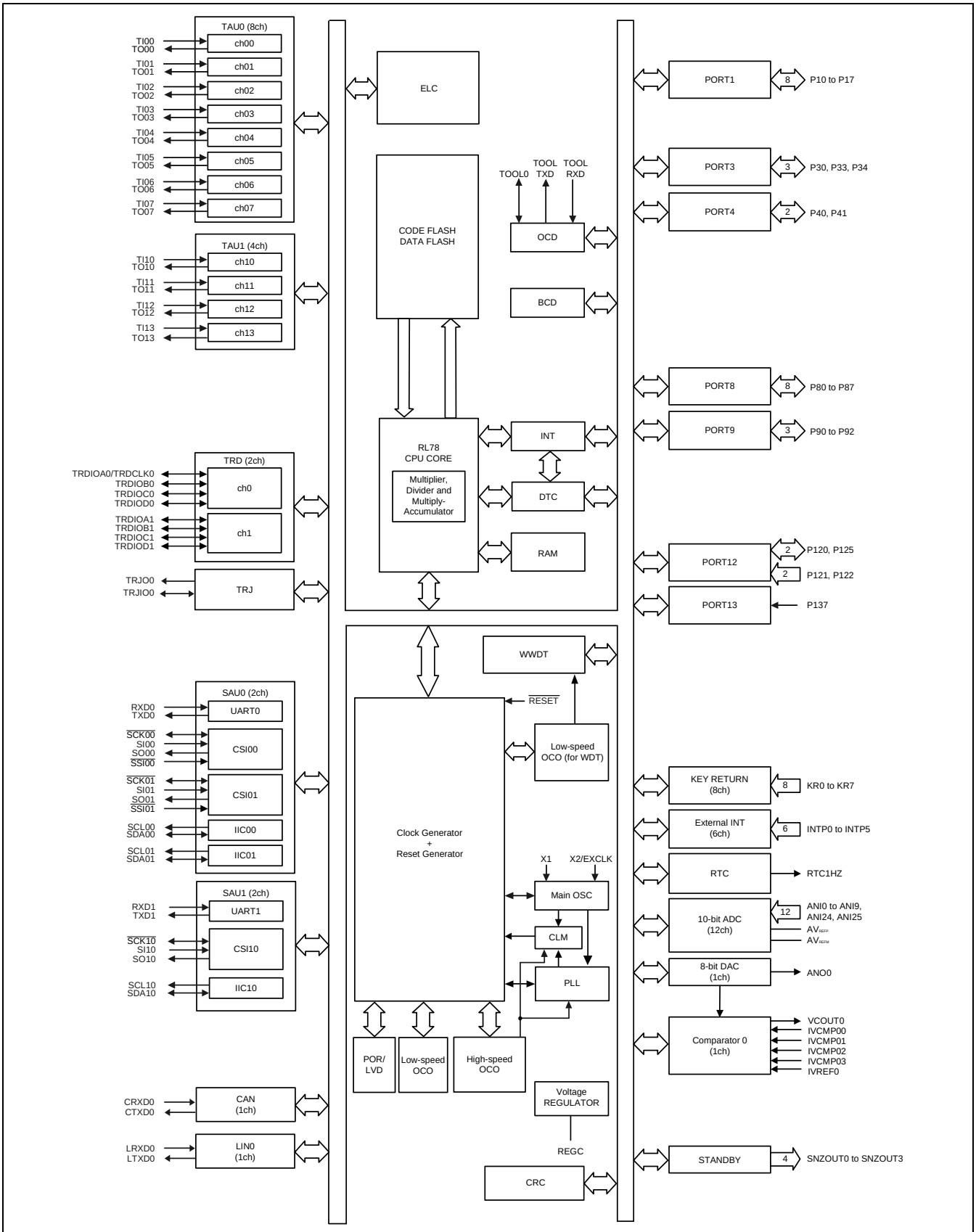



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.9 RL78/F14: Block Diagram of R5F10PAn (n = D, E) 30-pin Products

<R>

**Figure 1-9. Block Diagram**

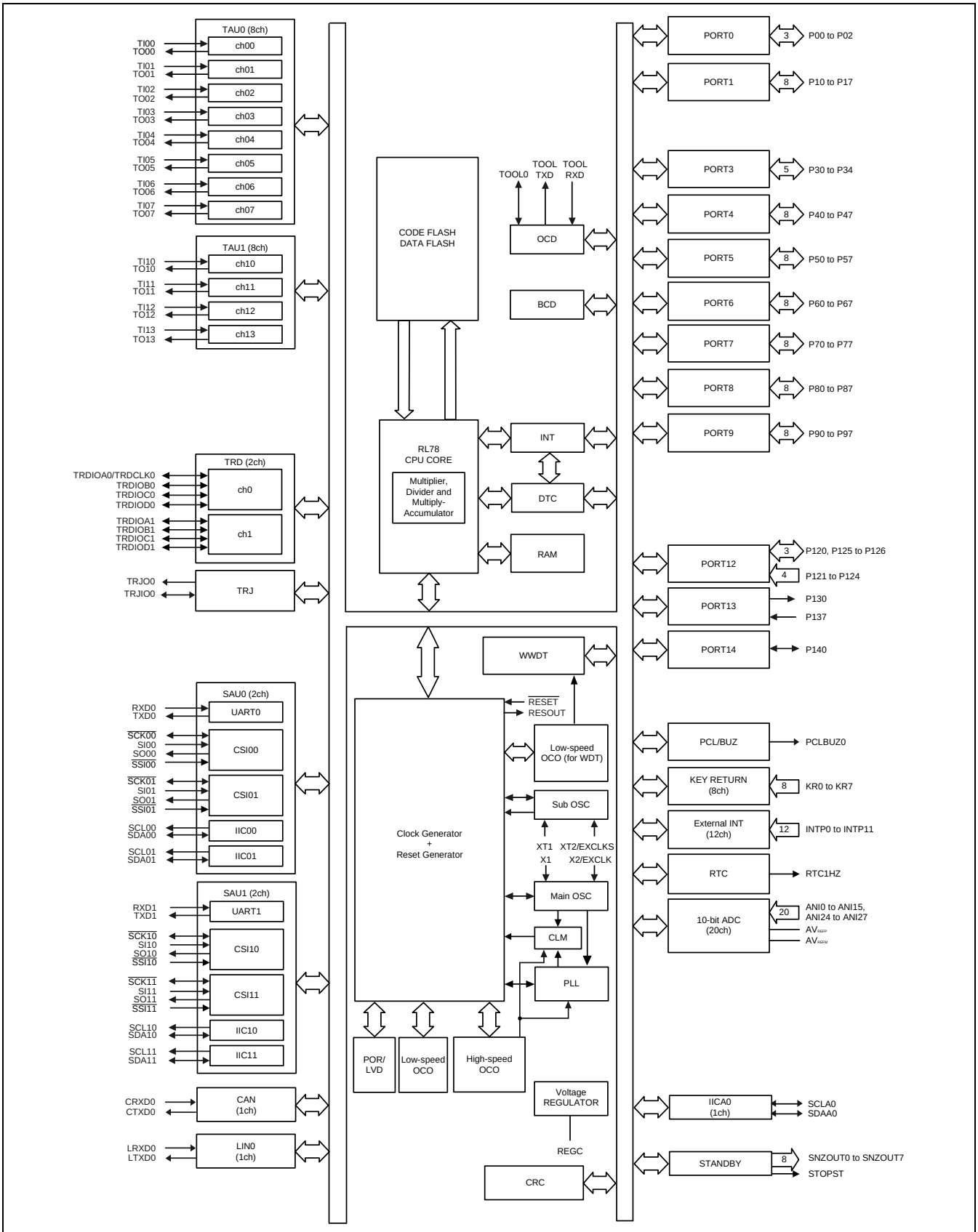



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.10 RL78/F13: Block Diagram of R5F10BMn (n = E, F, G) (CAN and LIN incorporated) 80-pin Products

<R>

**Figure 1-10. Block Diagram**

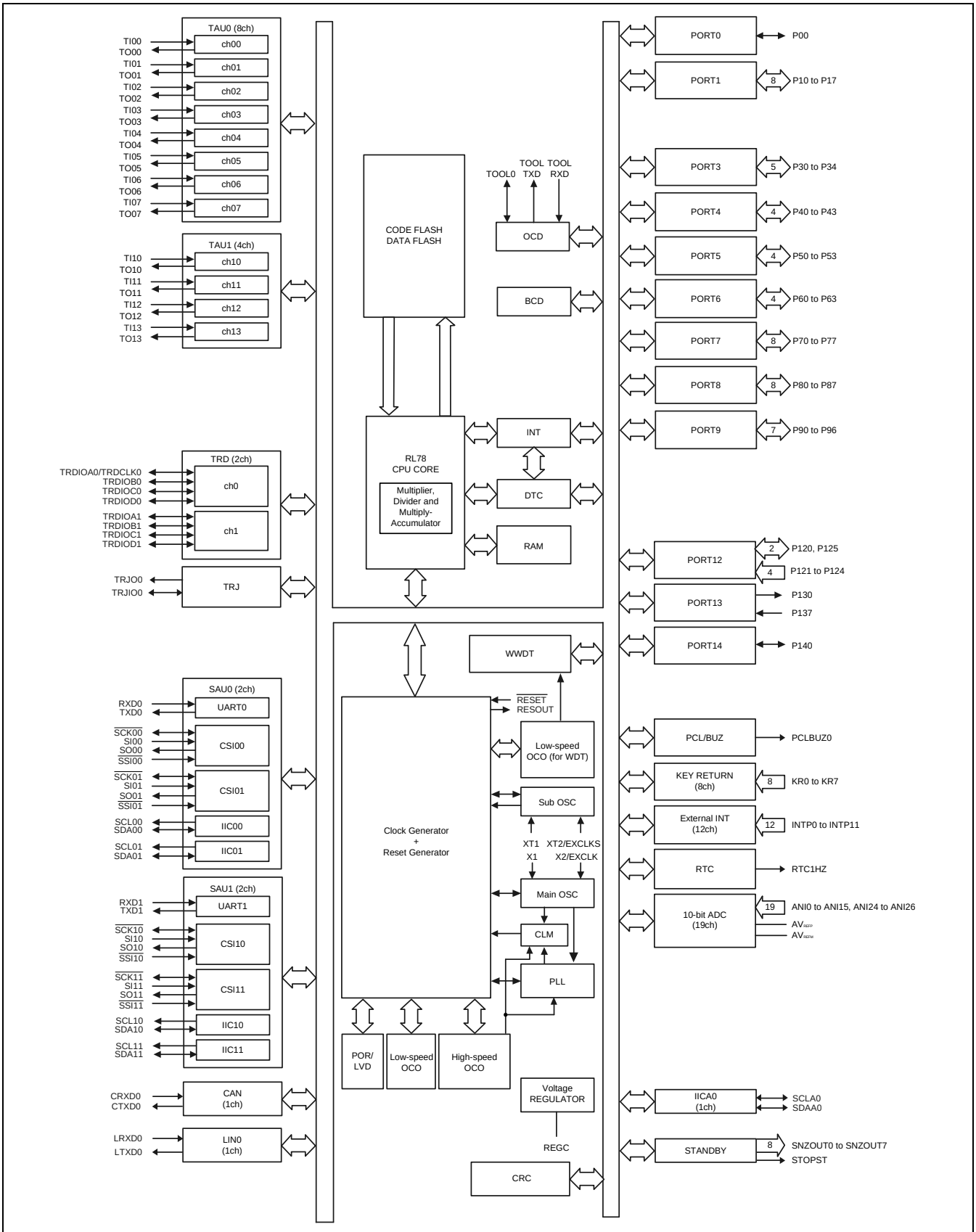



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.11 RL78/F13: Block Diagram of R5F10BLn (n = C, D, E, F, G) (CAN and LIN incorporated) 64-pin Products

<R>

**Figure 1-11. Block Diagram**

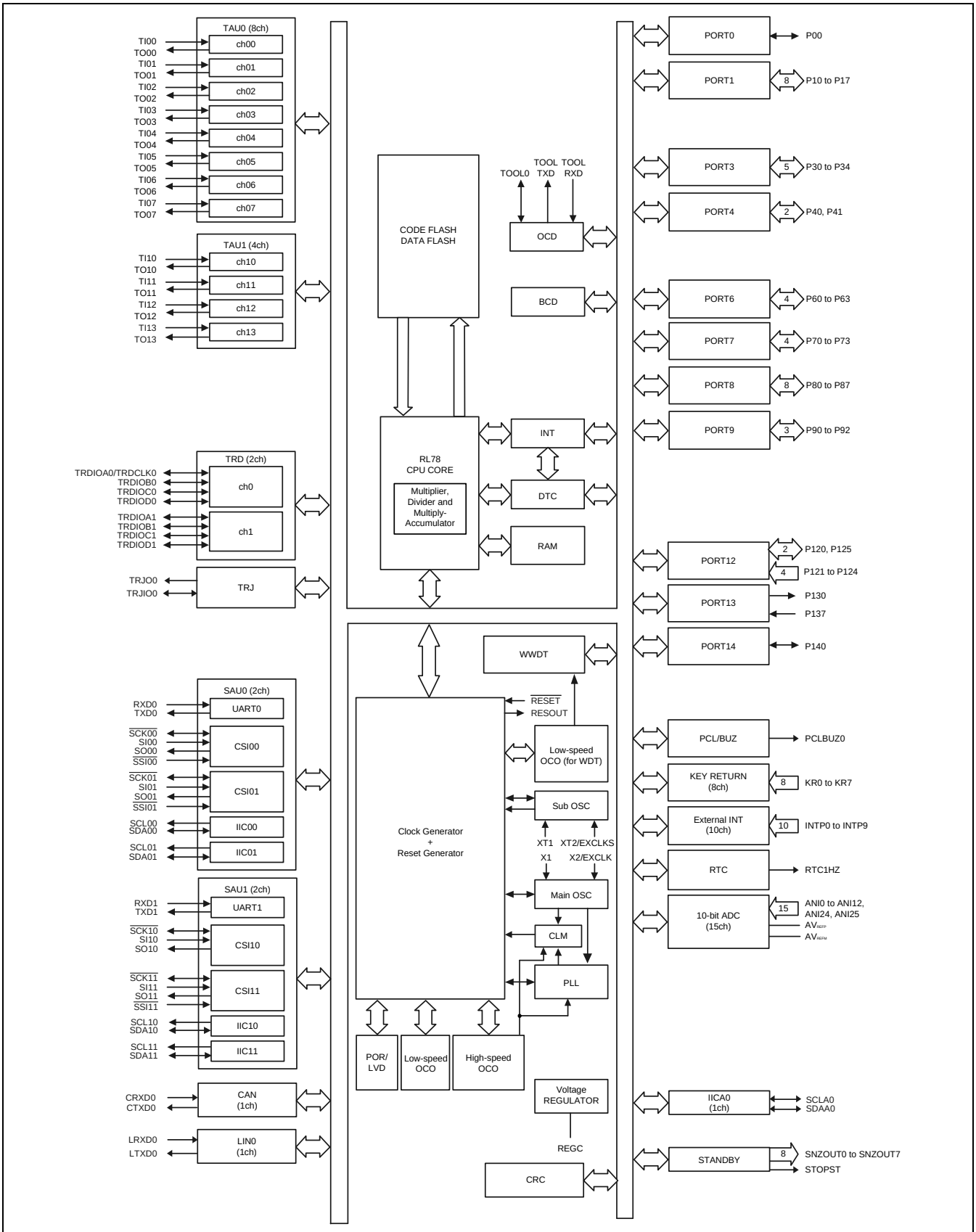



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.12 RL78/F13: Block Diagram of R5F10BGn (n = C, D, E, F, G) (CAN and LIN incorporated) 48-pin Products

<R>

**Figure 1-12. Block Diagram**

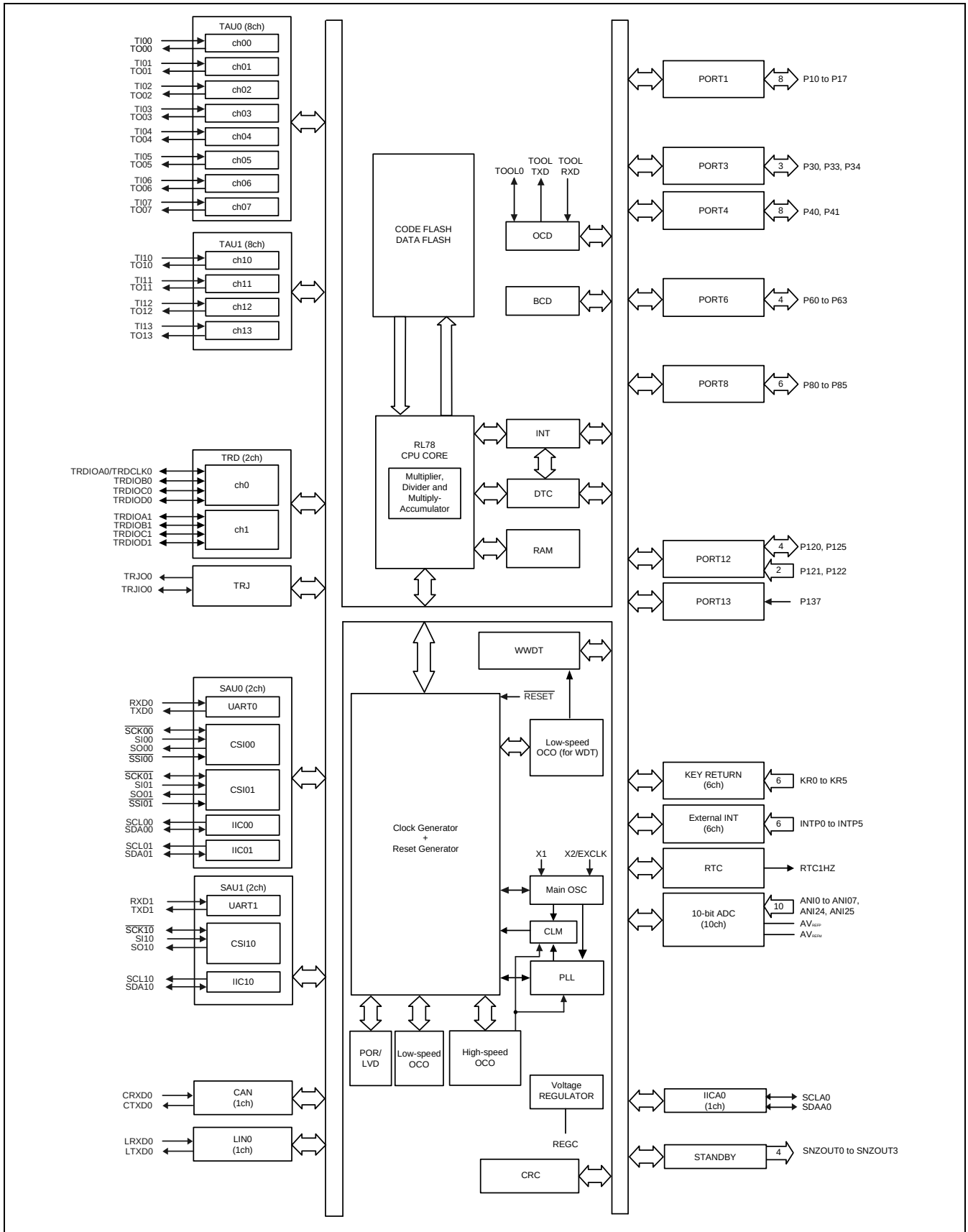



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.13 RL78/F13: Block Diagram of R5F10BBn (n = C, D, E, F, G) (CAN and LIN incorporated) 32-pin Products

<R>

**Figure 1-13. Block Diagram**

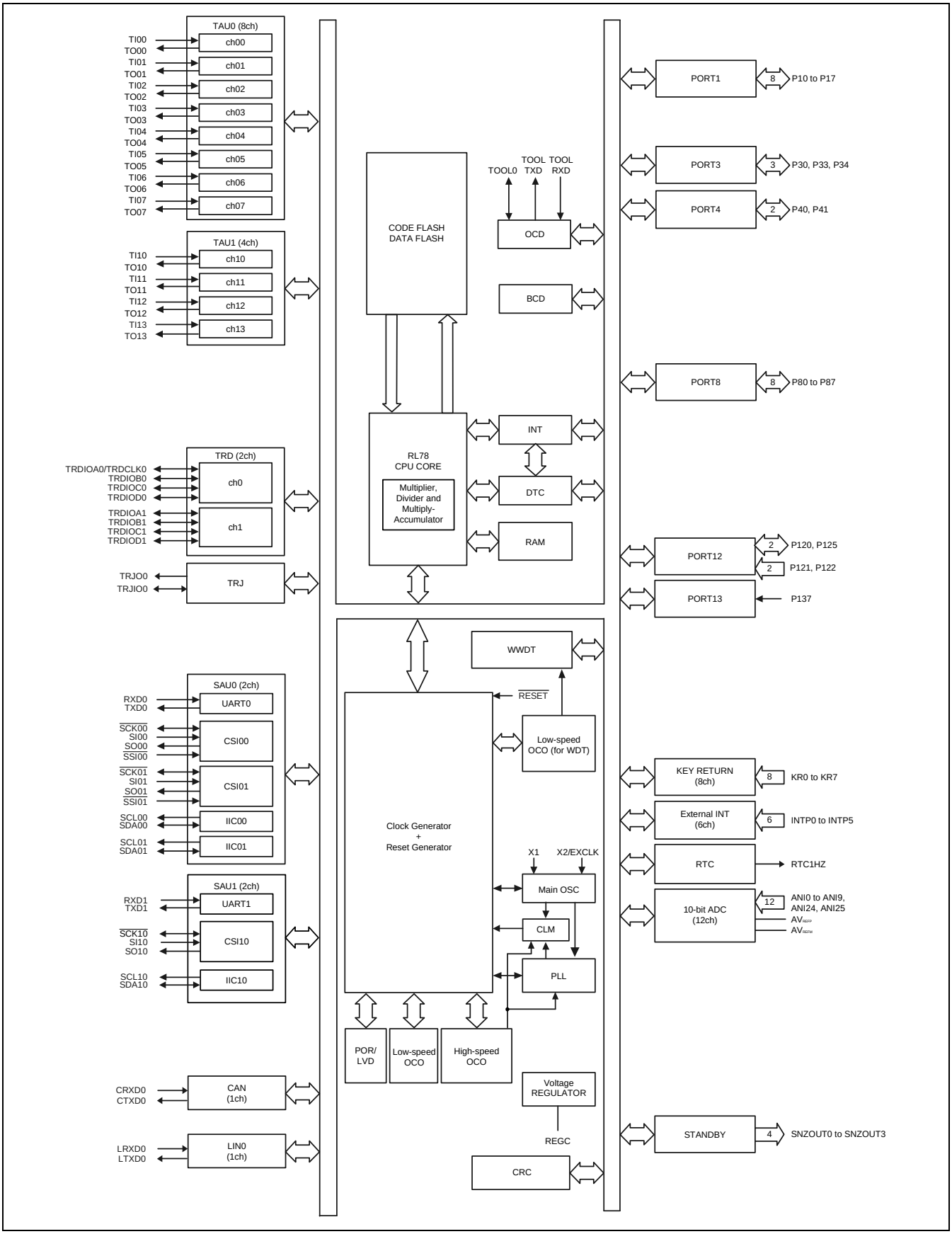



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.14 RL78/F13: Block Diagram of R5F10BAn (n = C, D, E, F, G) (CAN and LIN incorporated) 30-pin Products

<R>

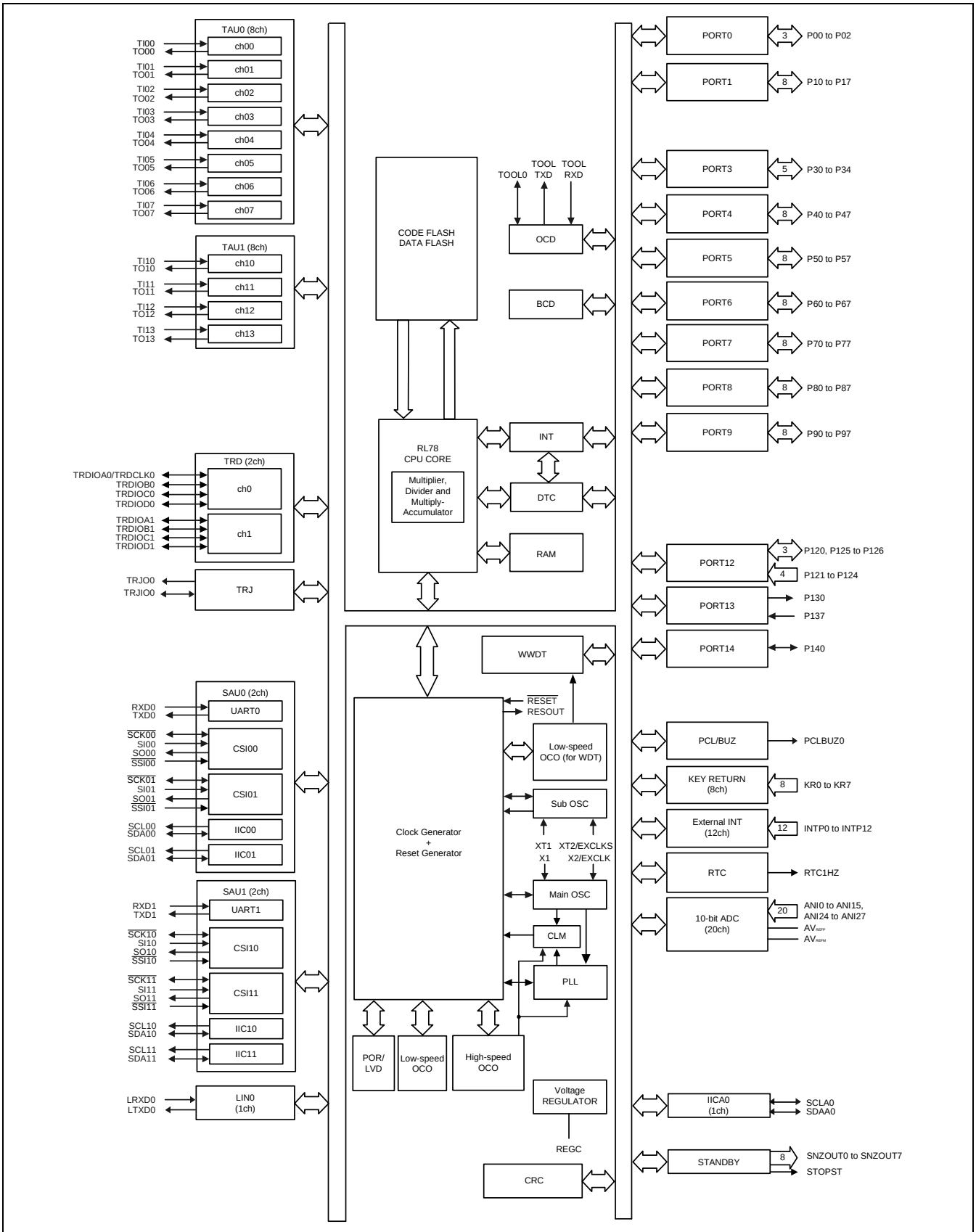
**Figure 1-14. Block Diagram**



### 1.4.15 RL78/F13: Block Diagram of R5F10AMn (n = E, F, G) (LIN incorporated) 80-pin Products

<R>

**Figure 1-15. Block Diagram**

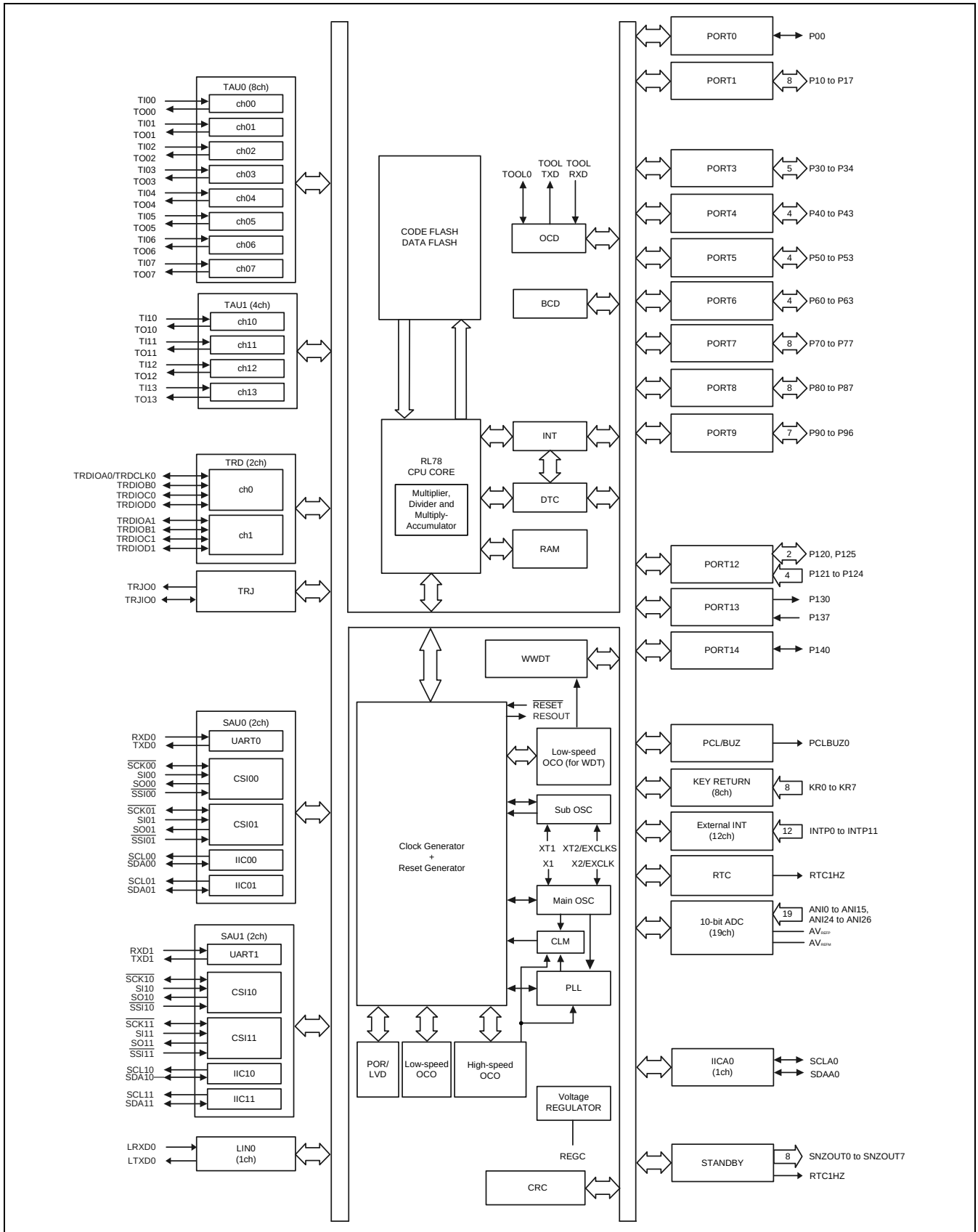



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.16 RL78/F13: Block Diagram of R5F10ALn (n = F, G) (LIN incorporated) 64-pin Products

<R>

**Figure 1-16. Block Diagram**

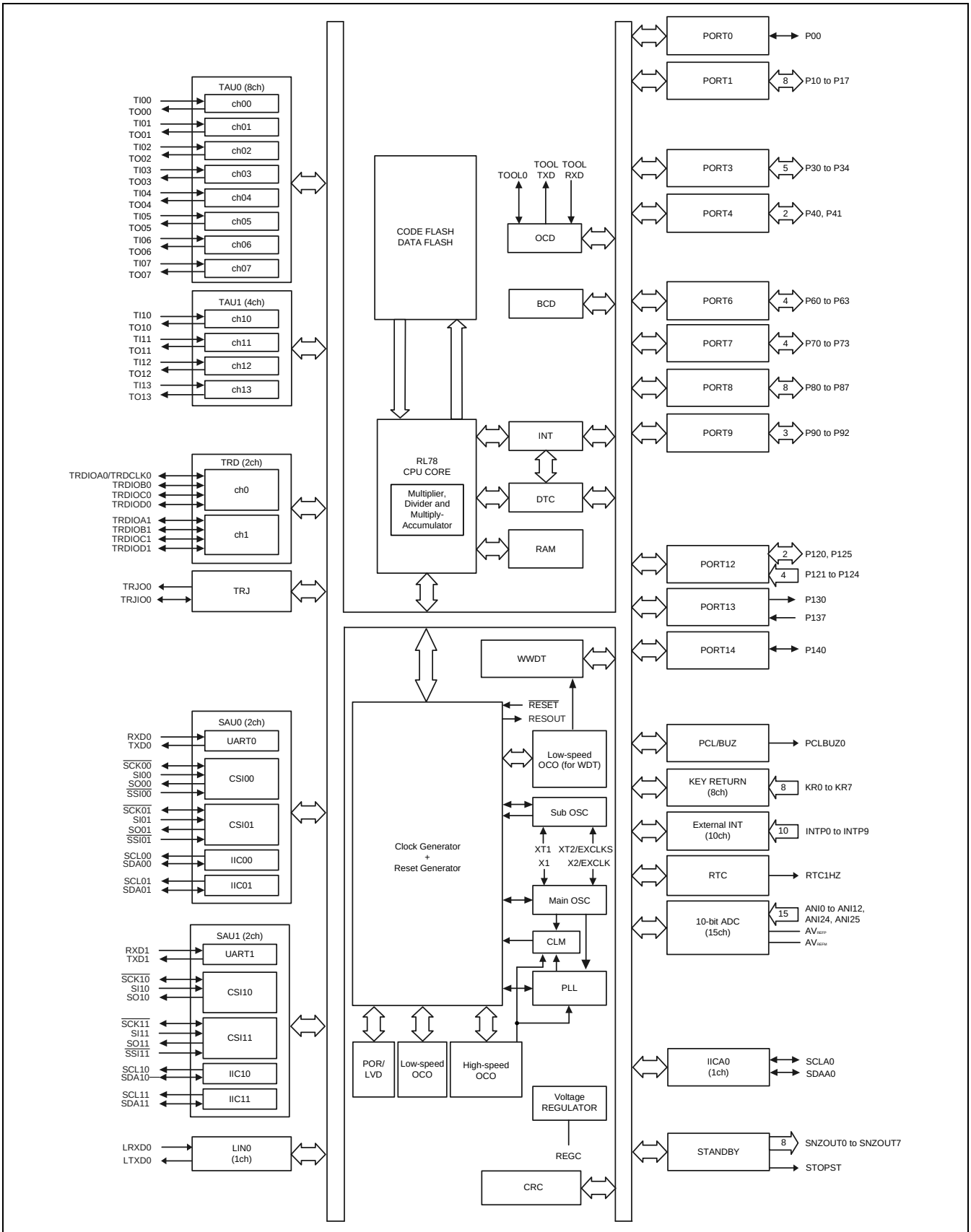



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.17 RL78/F13: Block Diagram of R5F10AGn (n = F, G) (LIN incorporated) 48-pin Products

<R>

**Figure 1-17. Block Diagram**

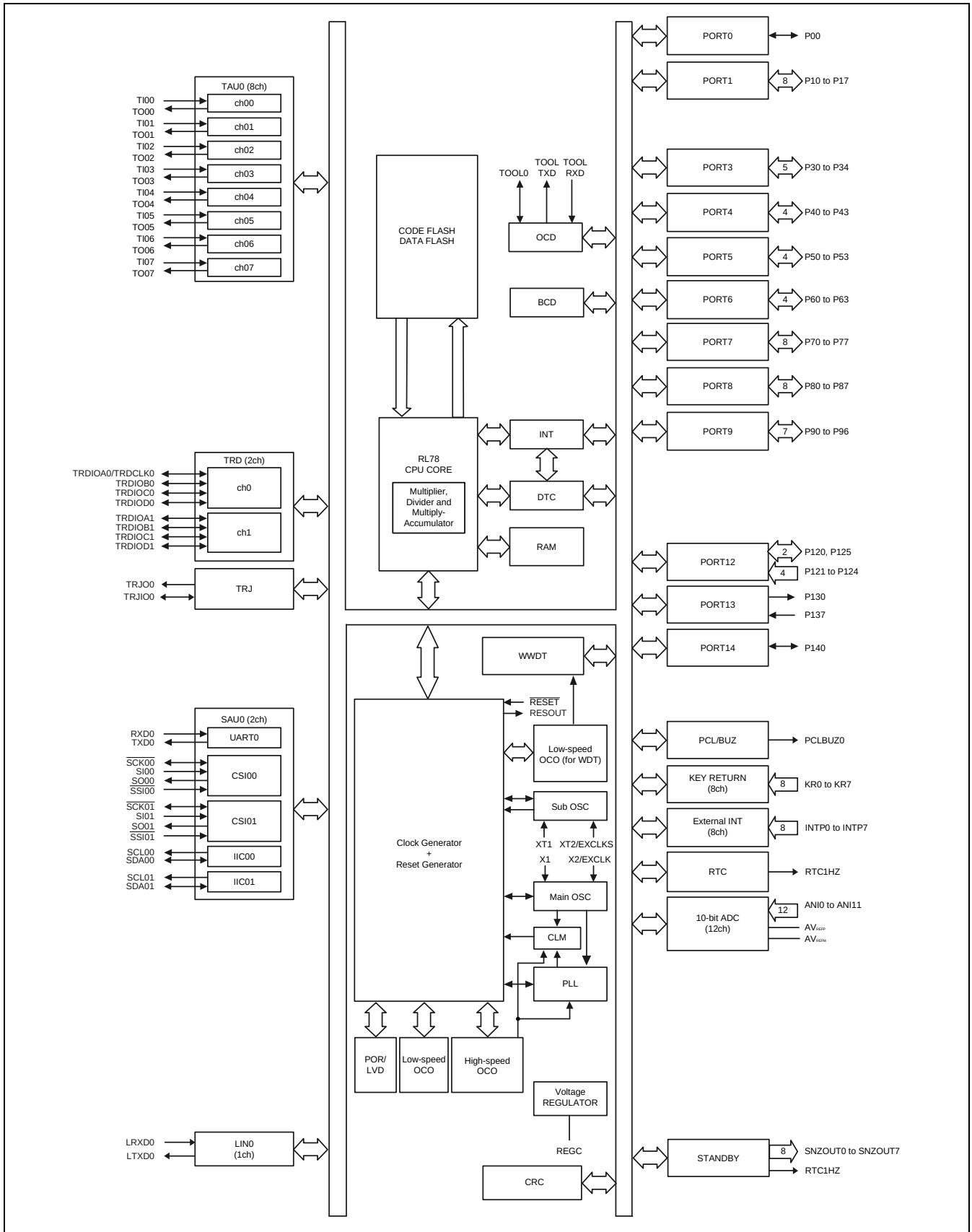



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.18 RL78/F13: Block Diagram of R5F10ALn (n = C, D, E) (LIN incorporated) 64-pin Products

<R>

**Figure 1-18. Block Diagram**

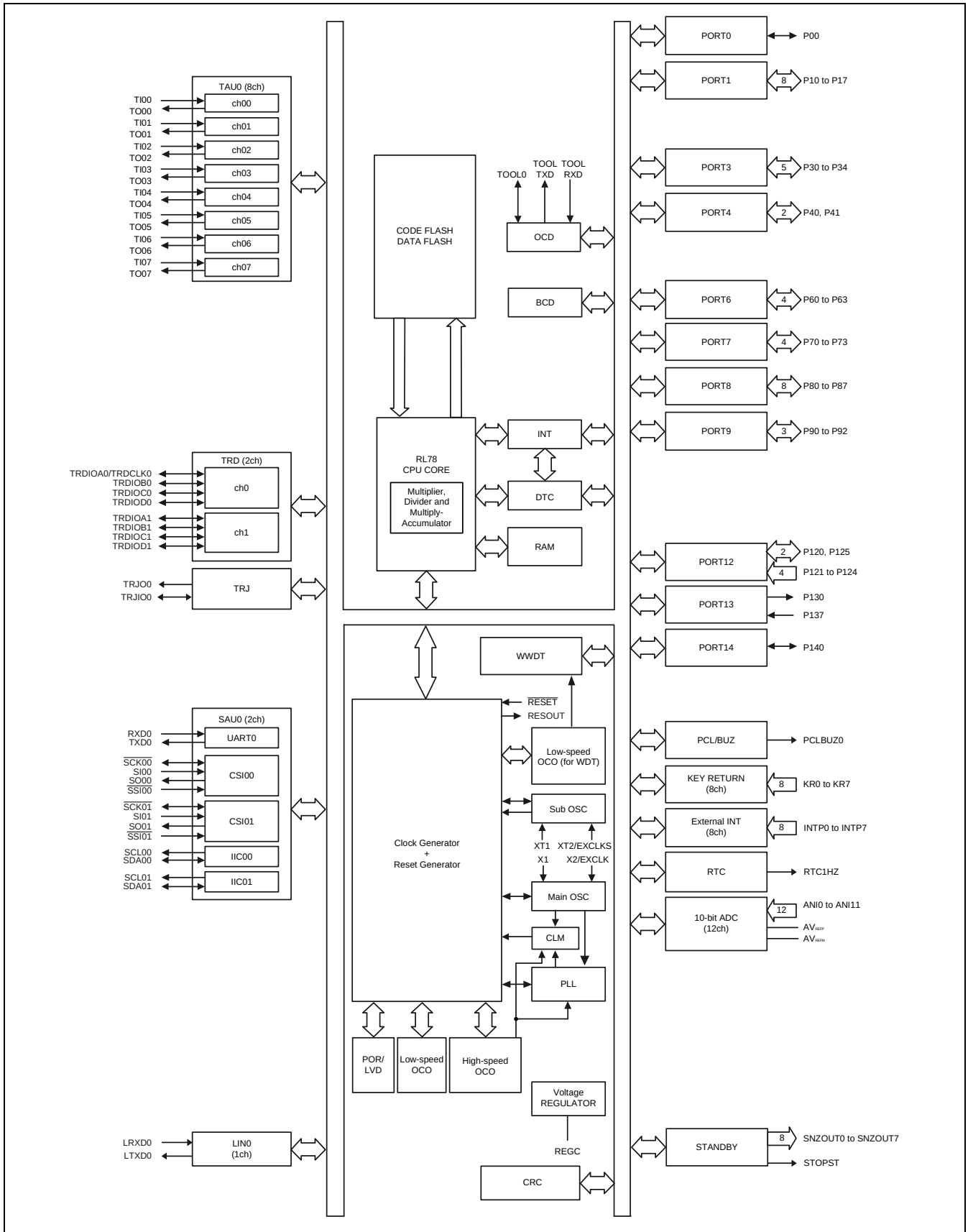



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.19 RL78/F13: Block Diagram of R5F10AGn (n = A, C, D, E) (LIN incorporated) 48-pin Products

<R>

**Figure 1-19. Block Diagram**

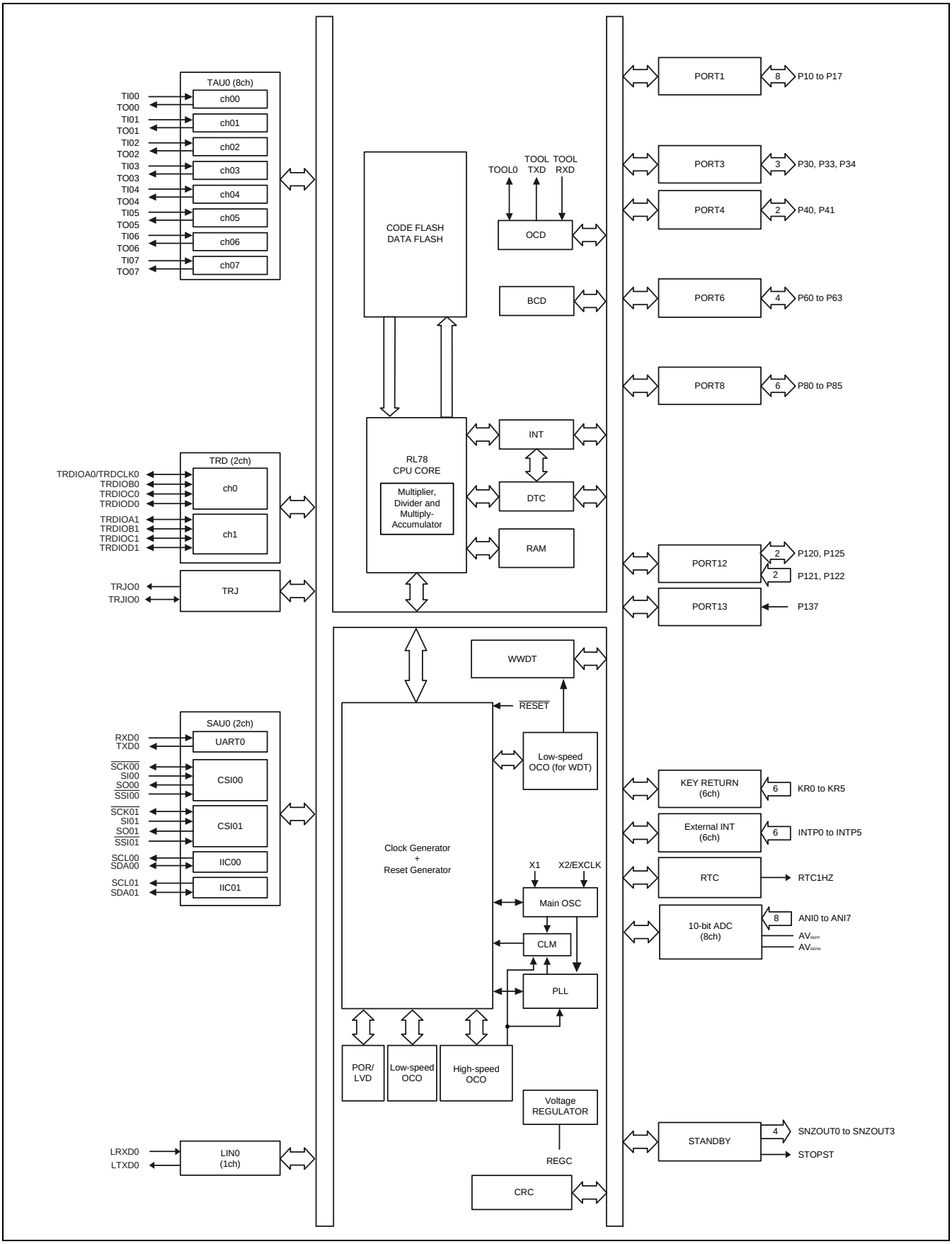



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.4.20 RL78/F13: Block Diagram of R5F10ABn (n = A, C, D, E) (LIN incorporated) 32-pin Products

<R>

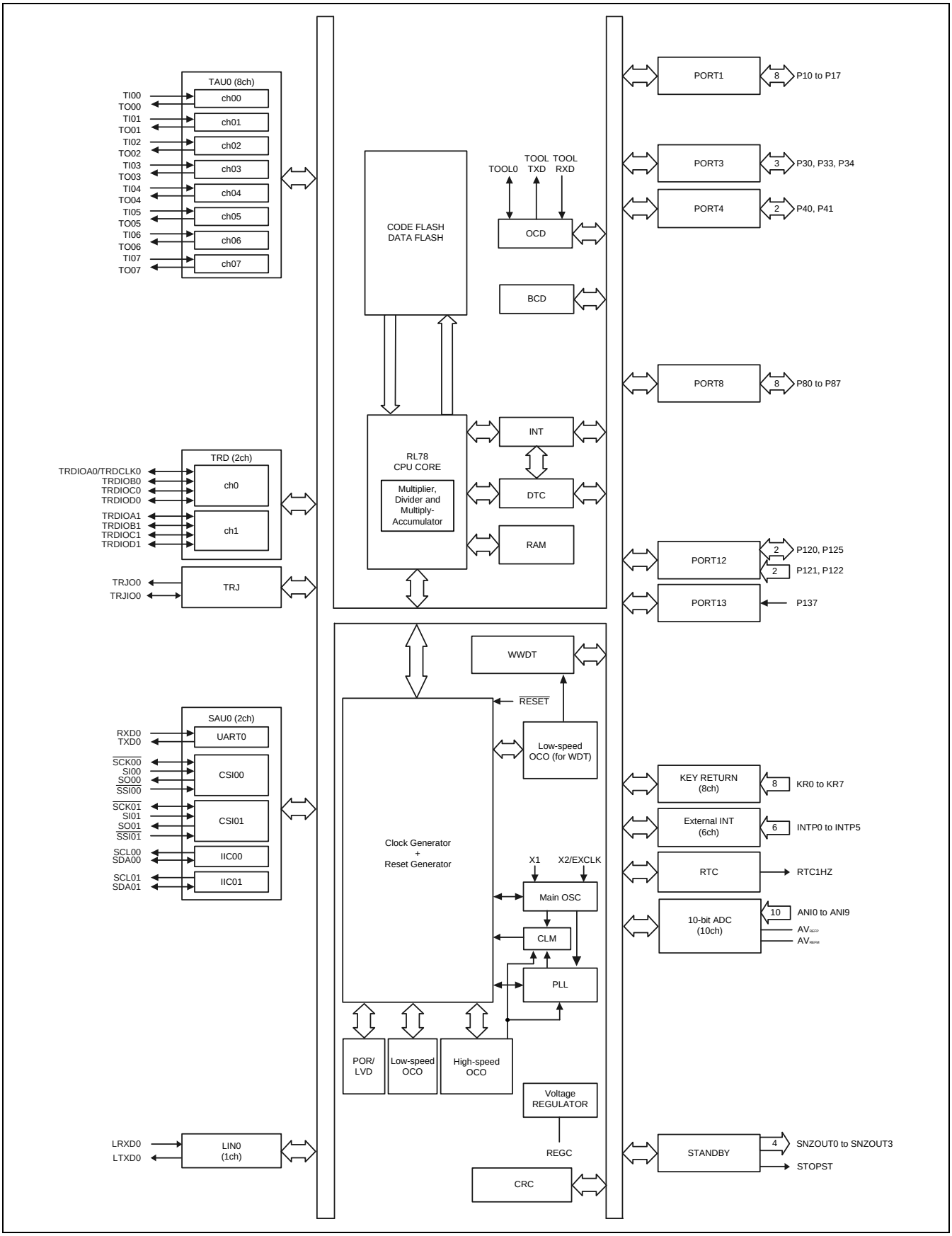
**Figure 1-20. Block Diagram**



### 1.4.21 RL78/F13: Block Diagram of R5F10AAn (n = A, C, D, E) (LIN incorporated) 30-pin Products

<R>

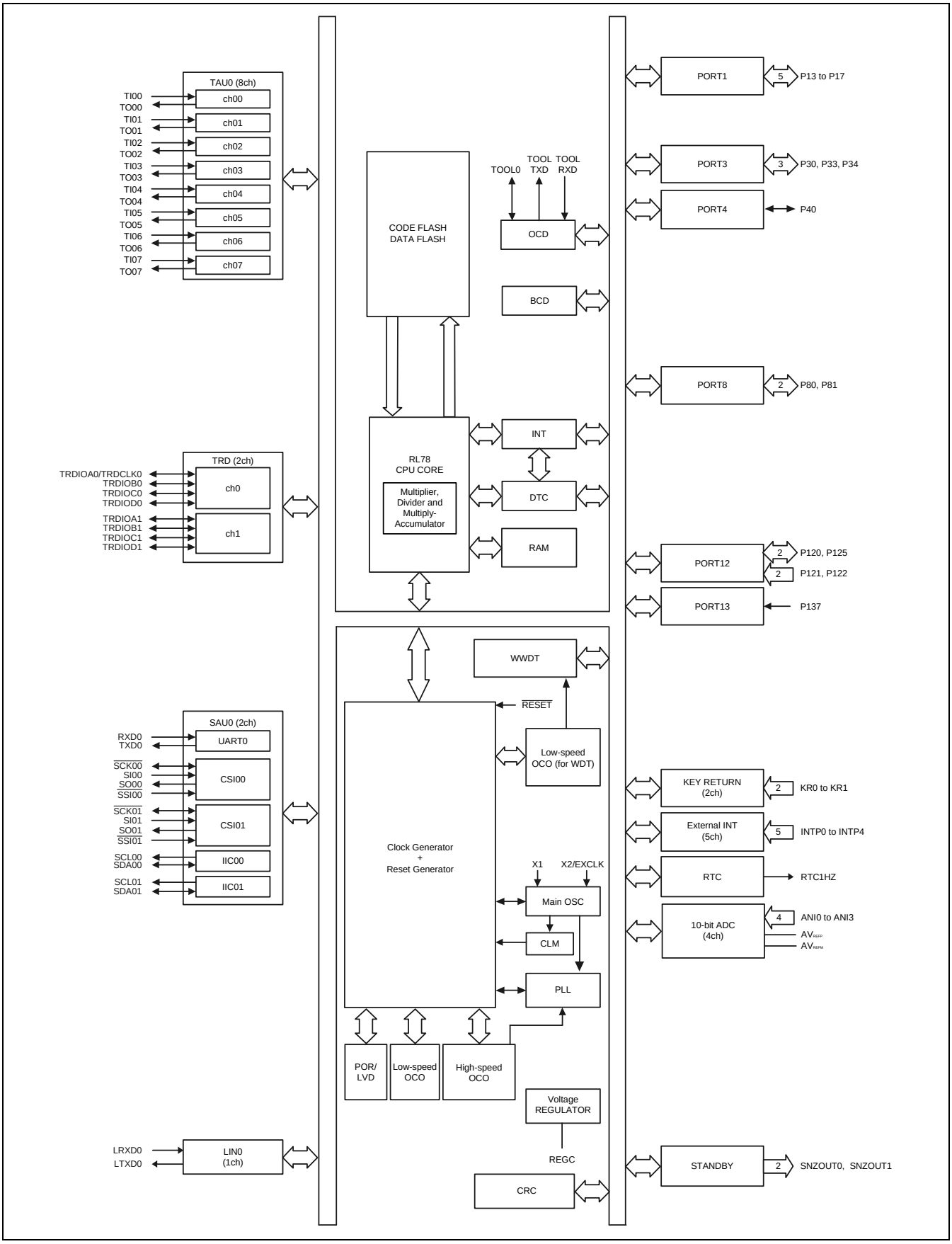
**Figure 1-21. Block Diagram**



### 1.4.22 RL78/F13: Block Diagram of R5F10A6n (n = A, C, D, E) (LIN incorporated) 20-pin Products

<R>

**Figure 1-22. Block Diagram**

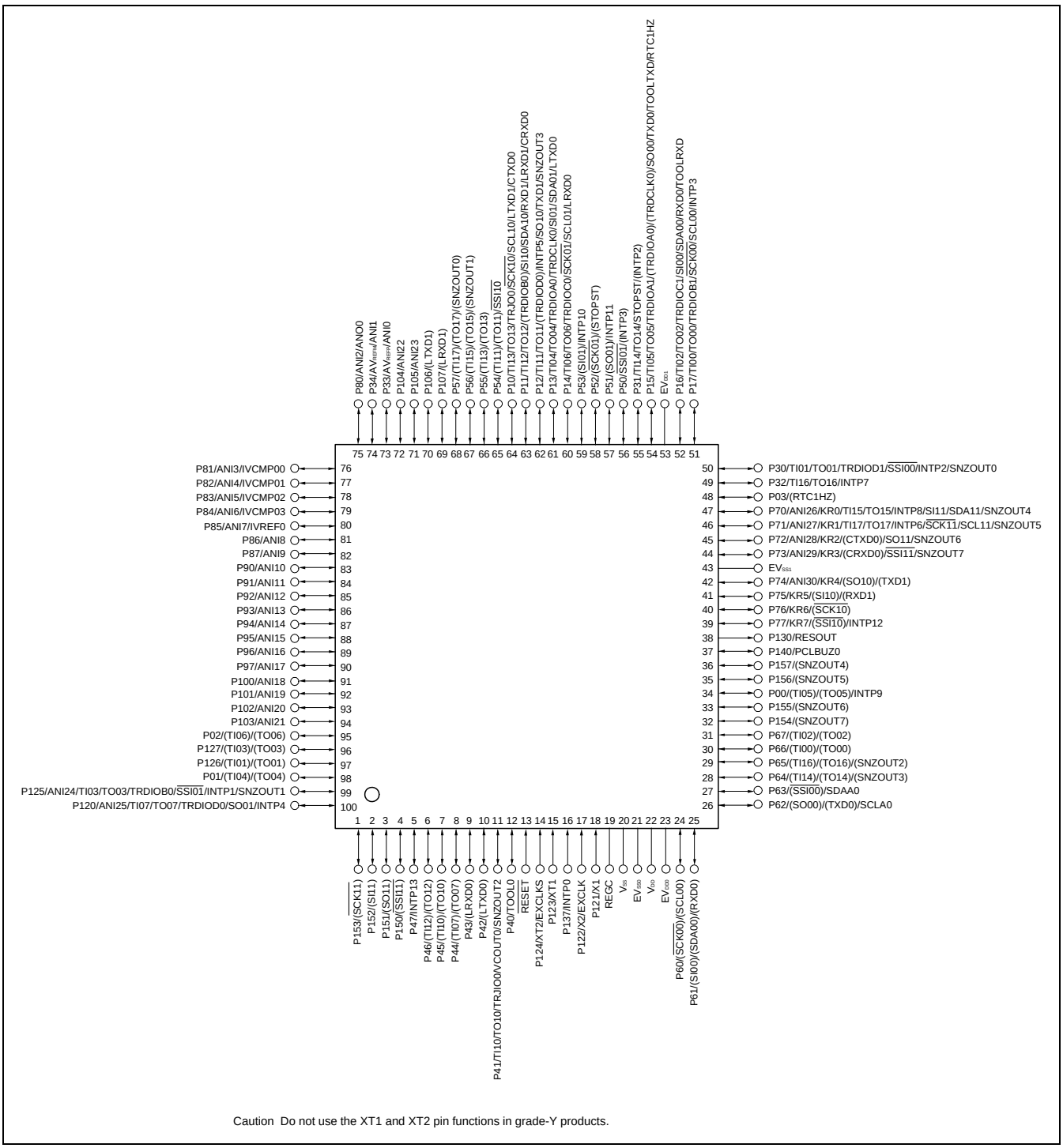


## 1.5 Pin Configurations

### 1.5.1 RL78/F14 Pin Configuration for 100-pin Products

- RL78/F14: 100-pin Plastic QFP (Fine Pitch) (14 x 14)

**Figure 1-23. RL78/F14 Pin Configuration for 100-pin Products**

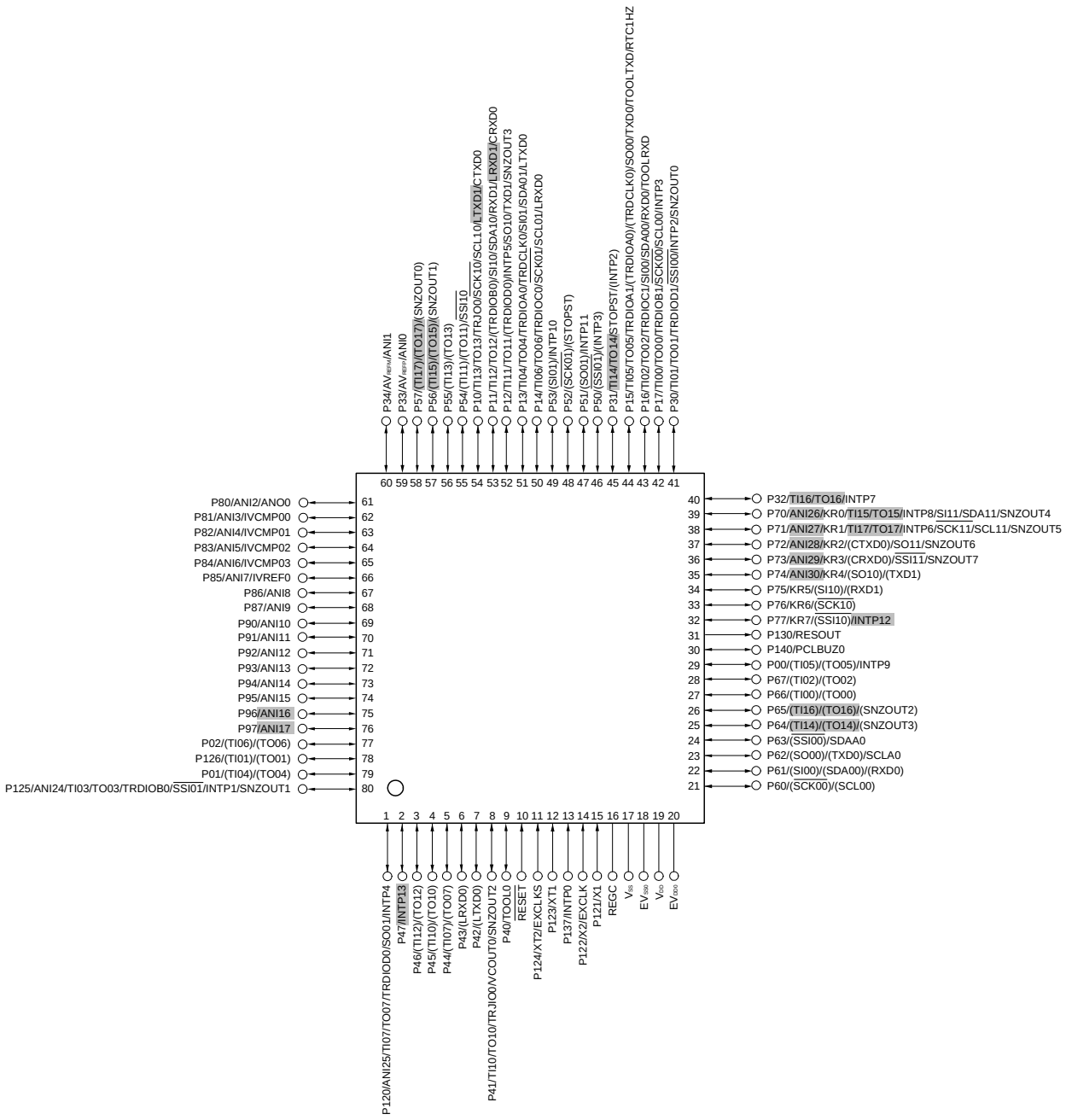



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.5.2 RL78/F14 Pin Configuration for 80-pin Products

- RL78/F14: 80-pin Plastic QFP (Fine Pitch) (12 x 12)

**Figure 1-24. RL78/F14 Pin Configuration for 80-pin Products**



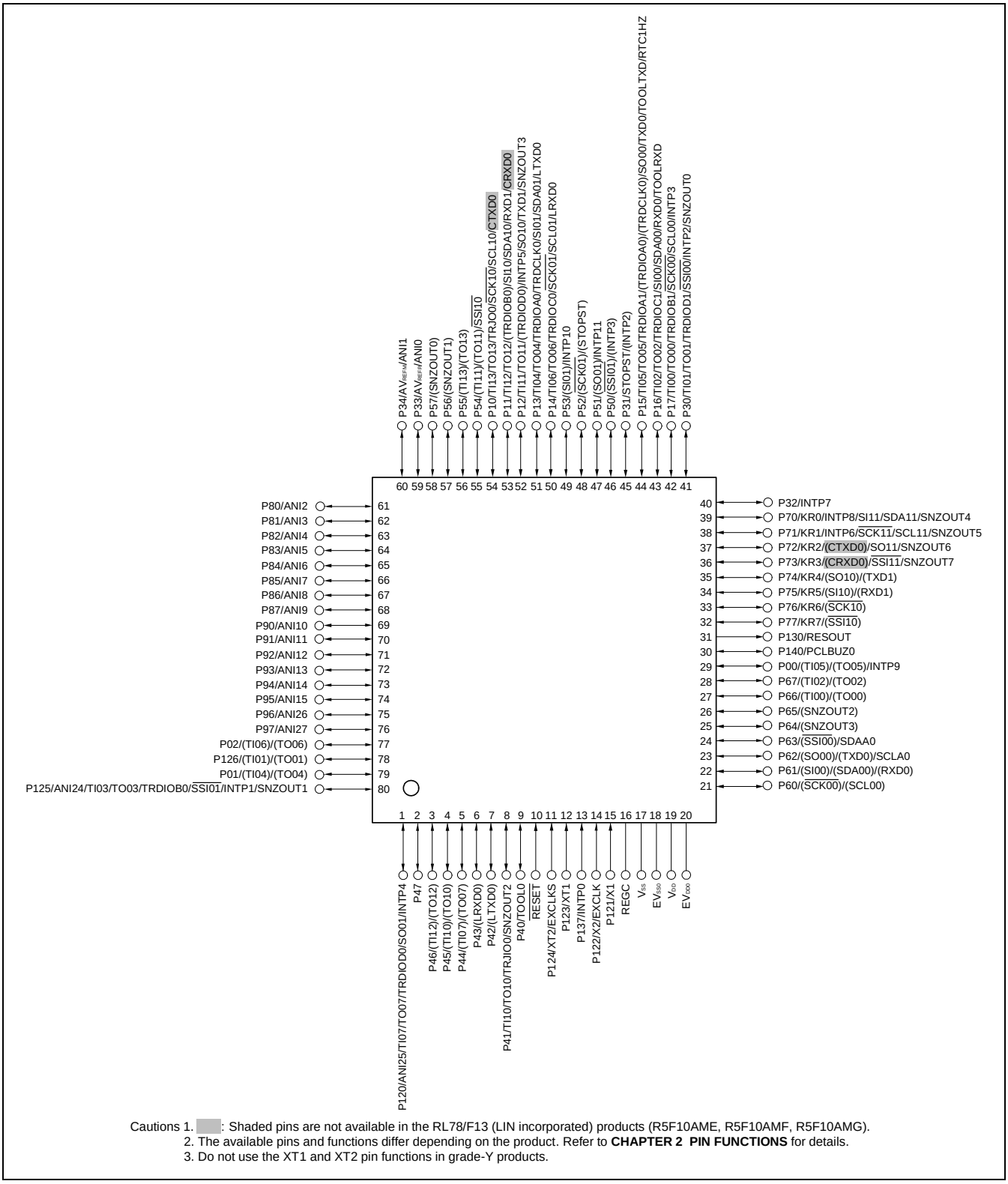

Cautions 1. : Shaded pins are not available in the R5F10PME or R5F10PMF.
Note, however, that the functions of pins 75 and 76, to which ANI16 and ANI17 are assigned, differ depending on the product, as follows:
- R5F10PMG, R5F10PMH, R5F10PMJ: P96/ANI16 (pin 75)
  P97/ANI17 (pin 76)
- R5F10PME, R5F10PMF: P96/ANI26 (pin 75)
  P97/ANI27 (pin 76)

2. The available pins and functions differ depending on the product. Refer to **CHAPTER 2 PIN FUNCTIONS** for details.
3. Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.5.3 RL78/F13 Pin Configuration for 80-pin Products

- RL78/F13: 80-pin Plastic QFP (Fine Pitch) (12 x 12)

**Figure 1-25. RL78/F13 Pin Configuration for 80-pin Products**

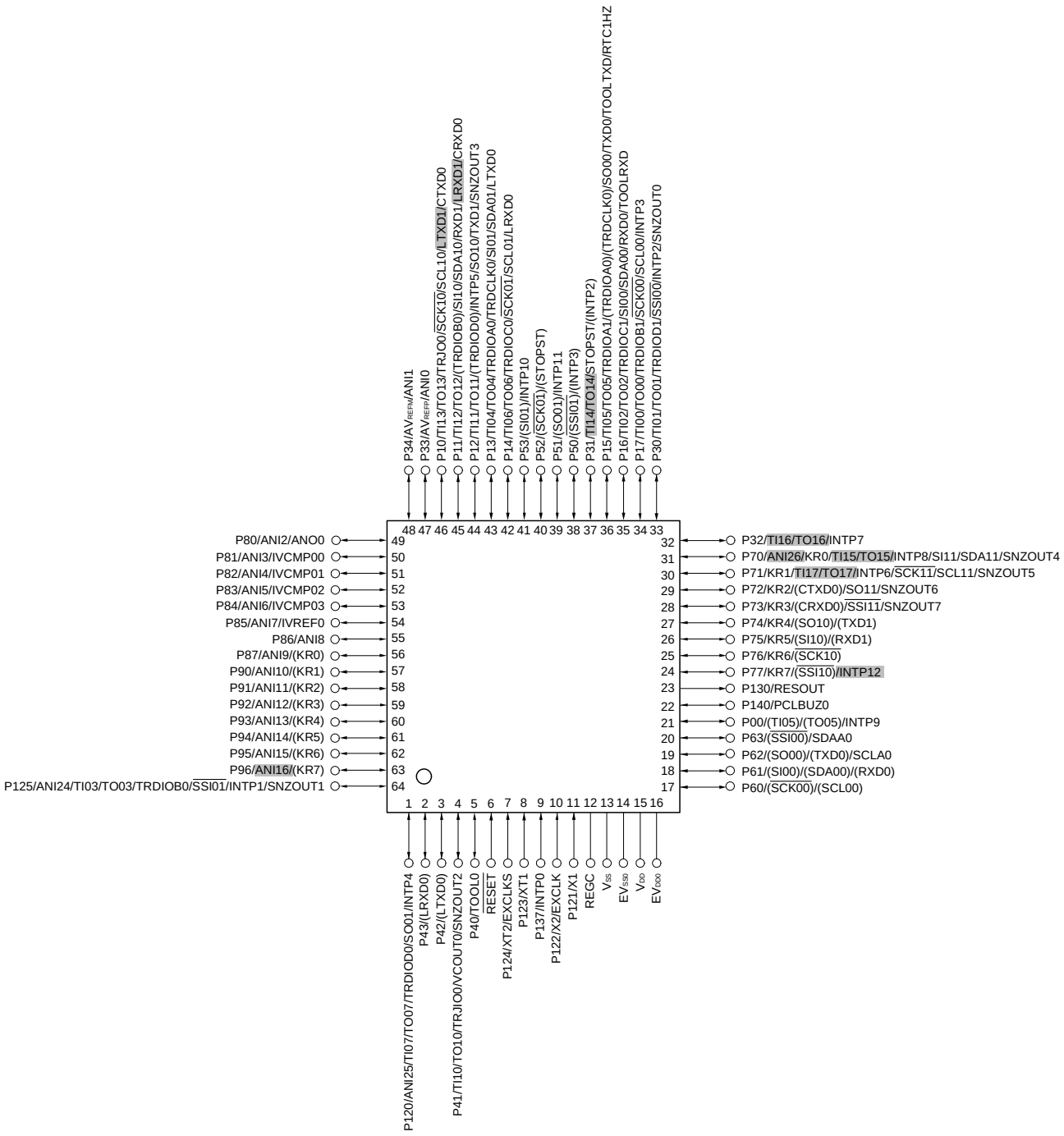



Cautions 1. : Shaded pins are not available in the RL78/F13 (LIN incorporated) products (R5F10AME, R5F10AMF, R5F10AMG).
2. The available pins and functions differ depending on the product. Refer to **CHAPTER 2 PIN FUNCTIONS** for details.
3. Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.5.4 RL78/F14 Pin Configuration for 64-pin Products

- RL78/F14: 64-pin Plastic QFP (Fine Pitch) (10 x 10)

**Figure 1-26. RL78/F14 Pin Configuration for 64-pin Products**

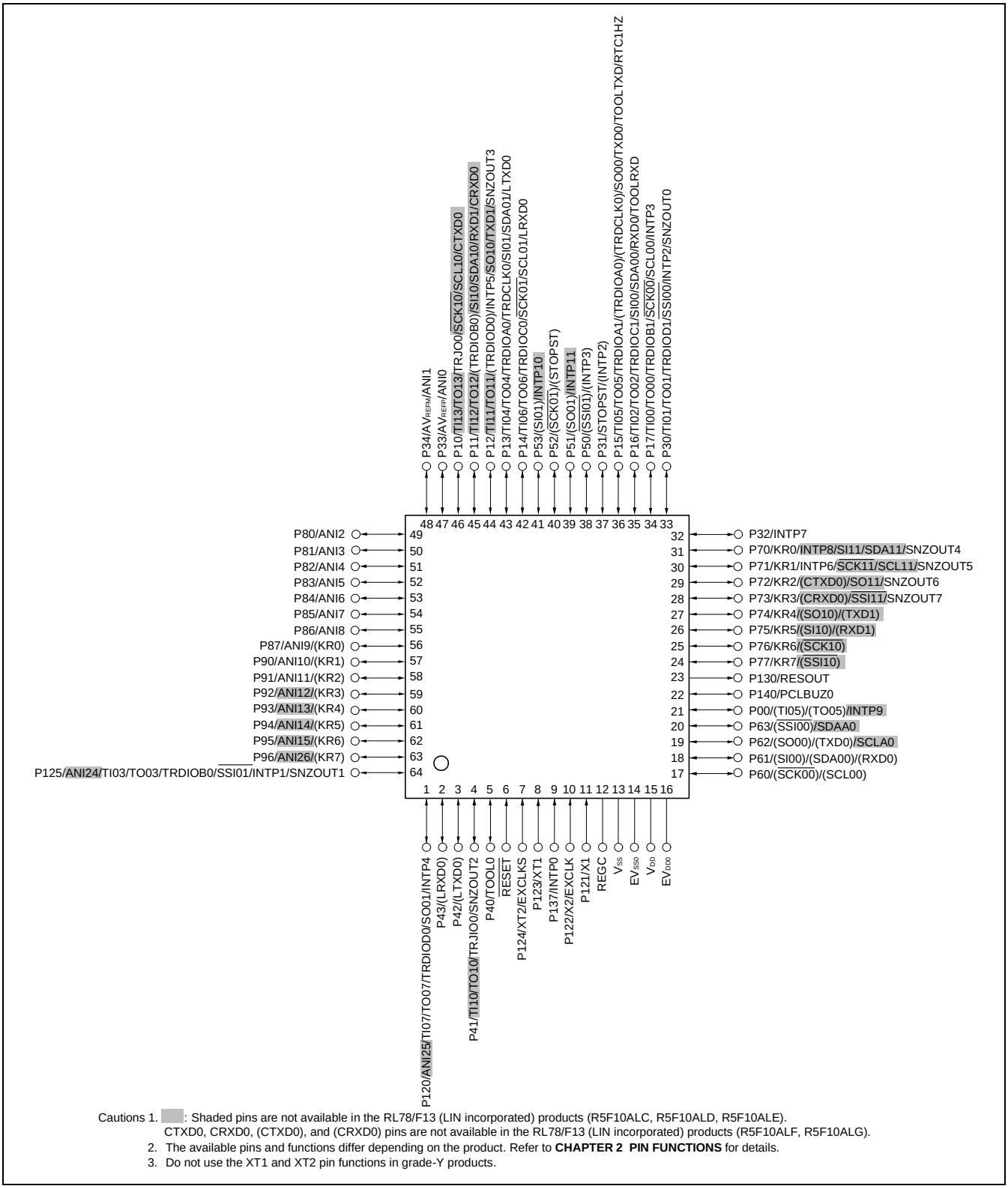



Cautions 1. : Shaded pins are not available in the R5F10PLE or R5F10PLF.
Note, however, that the functions of pin 63 to which ANI16 is assigned, differ depending on the product, as follows:
• R5F10PLG, R5F10PLH, R5F10PLJ: P96/ANI16/(KR7)
• R5F10PLE, R5F10PLF: P96/ANI26/(KR7)
2. The available pins and functions differ depending on the product. Refer to **CHAPTER 2 PIN FUNCTIONS** for details.
3. Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.5.5 RL78/F13 Pin Configuration for 64-pin Product

- RL78/F13: 64-pin Plastic QFP (Fine Pitch) (10 x 10)

**Figure 1-27. RL78/F13 Pin Configuration for 64-pin Products**

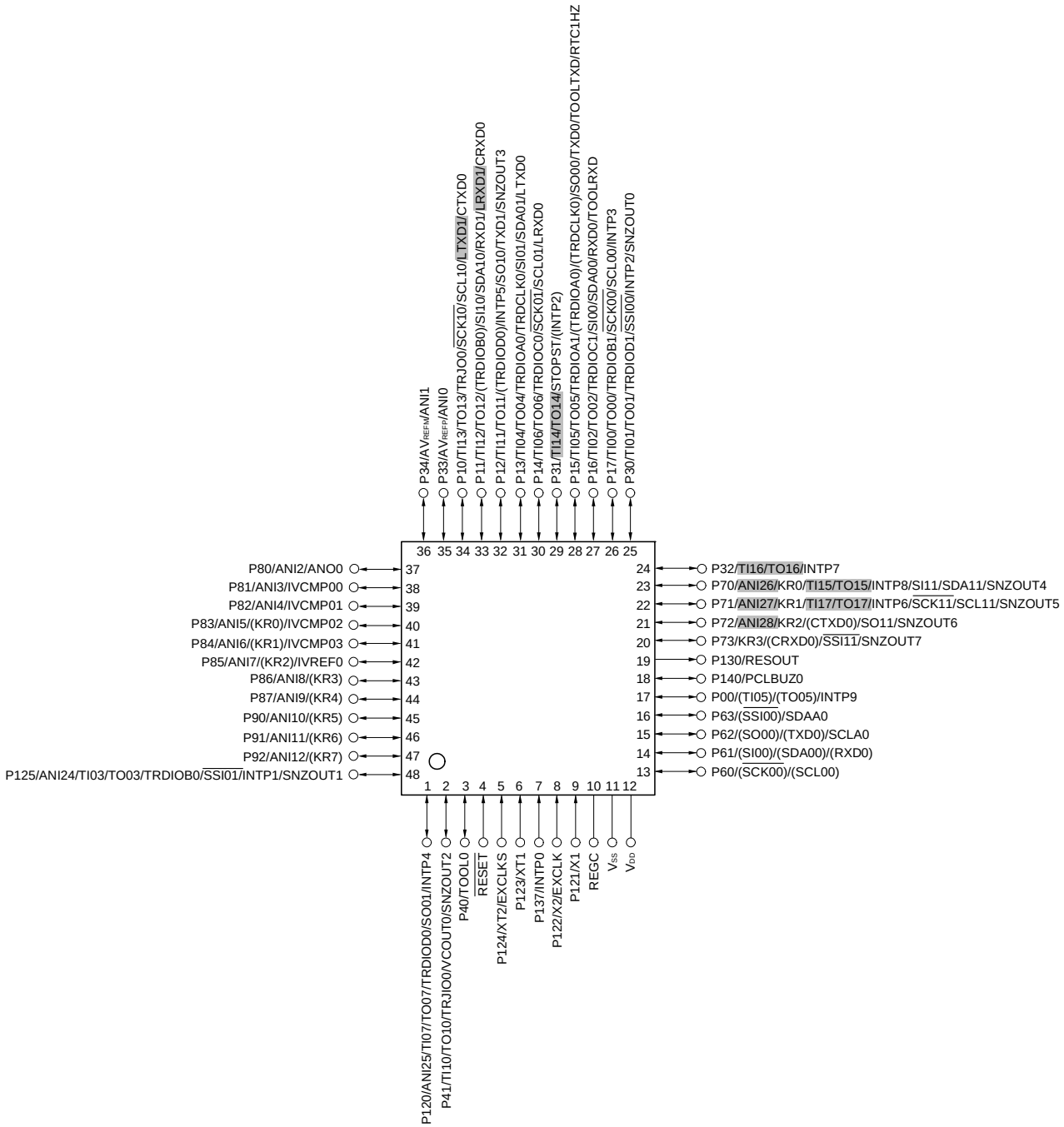



Cautions 1. : Shaded pins are not available in the RL78/F13 (LIN incorporated) products (R5F10ALC, R5F10ALD, R5F10ALE).
CTXD0, CRXD0, (CTXD0), and (CRXD0) pins are not available in the RL78/F13 (LIN incorporated) products (R5F10ALF, R5F10ALG).
2. The available pins and functions differ depending on the product. Refer to **CHAPTER 2 PIN FUNCTIONS** for details.
3. Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.5.6 RL78/F14 Pin Configuration for 48-pin Products

- RL78/F14: 48-pin Plastic QFP and QFN

**Figure 1-28. RL78/F14 Pin Configuration for 48-pin Products**

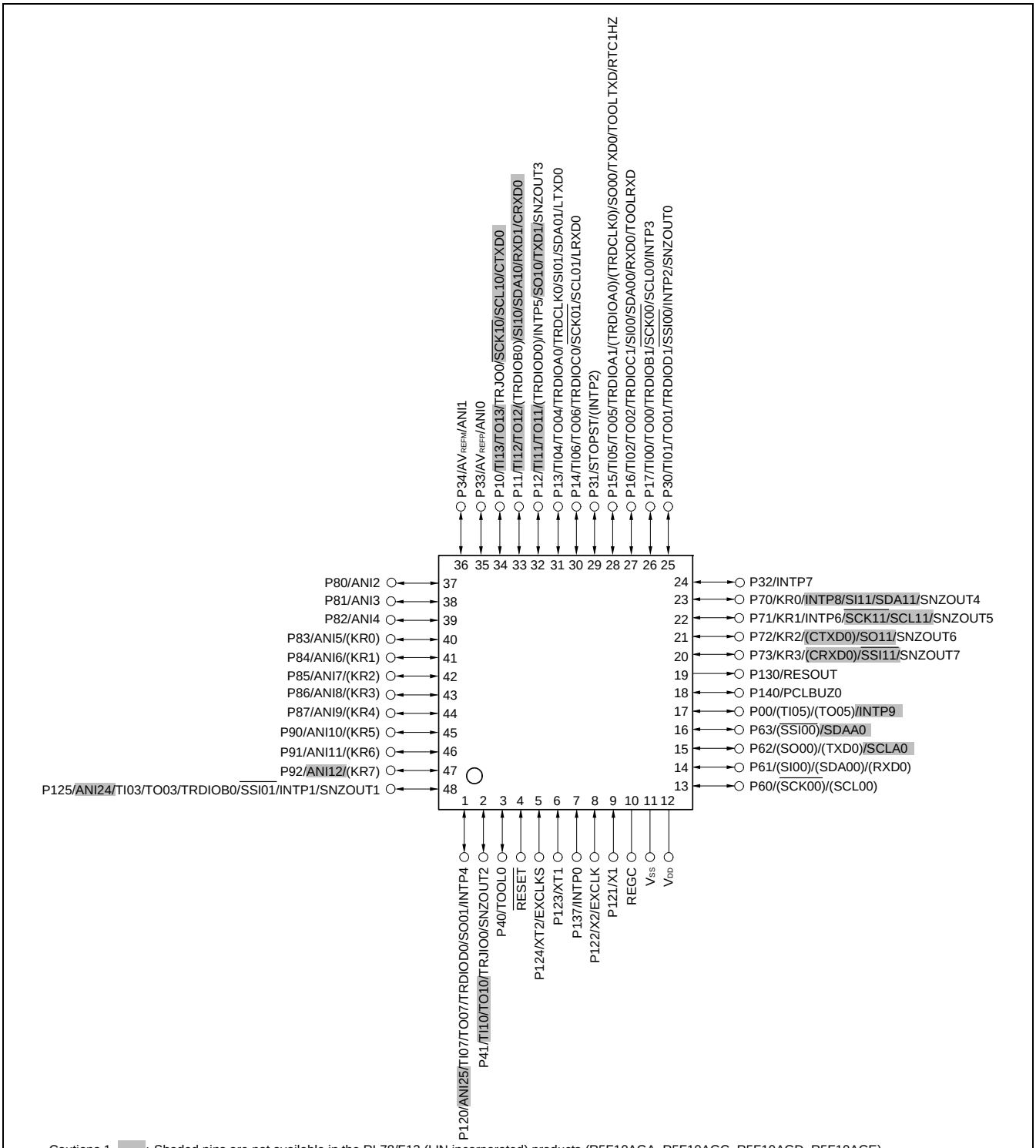



Cautions 1. : Shaded pins are not available in the R5F10PGD, R5F10PGE, or R5F10PGF.
2. The available pins and functions differ depending on the product. Refer to **CHAPTER 2 PIN FUNCTIONS** for details.
3. Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.5.7 RL78/F13 Pin Configuration for 48-pin Products

- RL78/F13: 48-pin Plastic QFP and QFN

**Figure 1-29. RL78/F13 Pin Configuration for 48-pin Products**

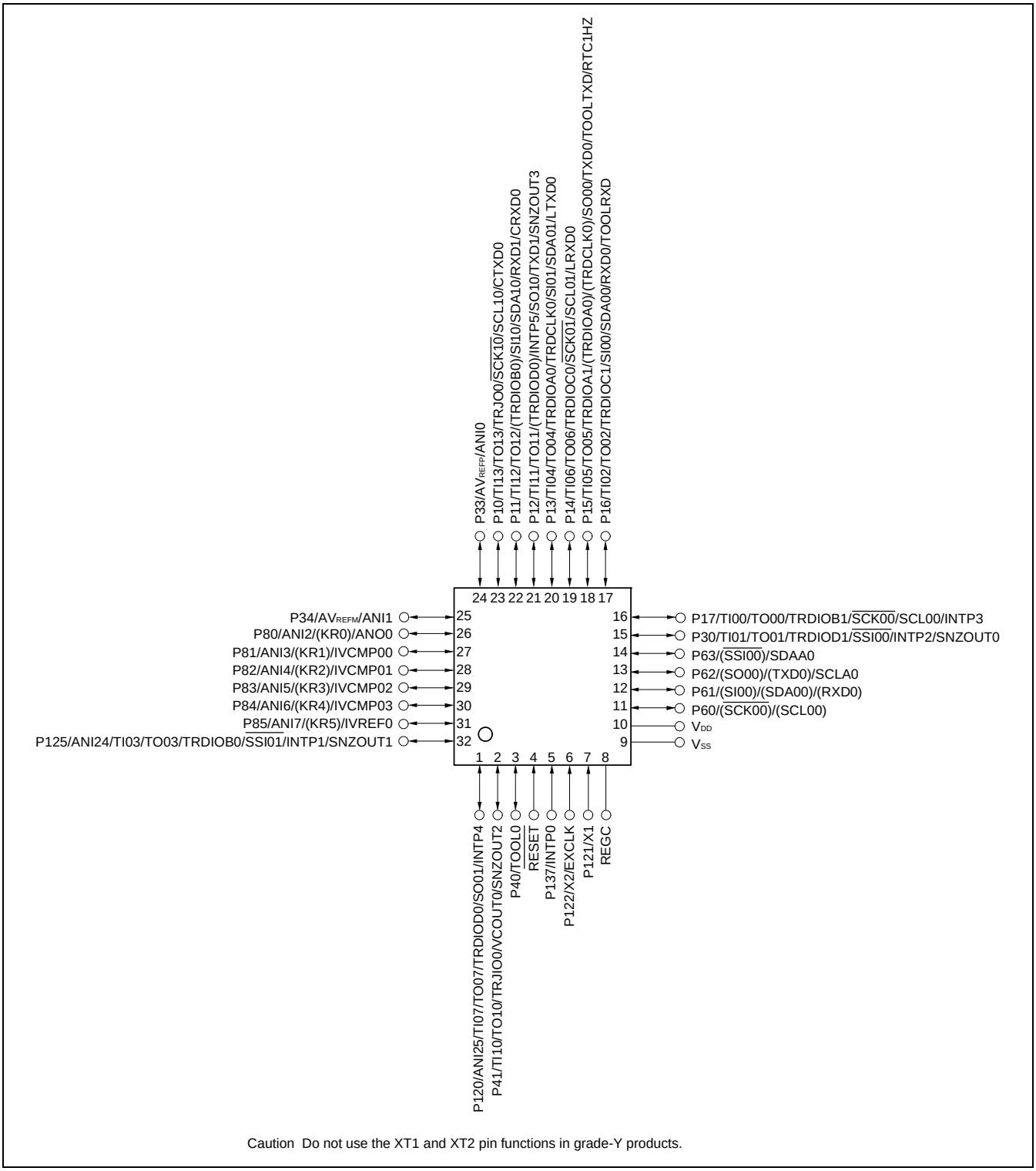



Cautions 1. : Shaded pins are not available in the RL78/F13 (LIN incorporated) products (R5F10AGA, R5F10AGC, R5F10AGD, R5F10AGE). CTXD0, CRXD0, (CTXD0), and (CRXD0) pins are not available in the RL78/F13 (LIN incorporated) products (R5F10AGF, R5F10AGG).
2. The available pins and functions differ depending on the product. Refer to **CHAPTER 2 PIN FUNCTIONS** for details.
3. Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.5.8 RL78/F14 Pin Configuration for 32-pin Products

- RL78/F14: 32-pin Plastic QFN

**Figure 1-30. RL78/F14 Pin Configuration for 32-pin Product**

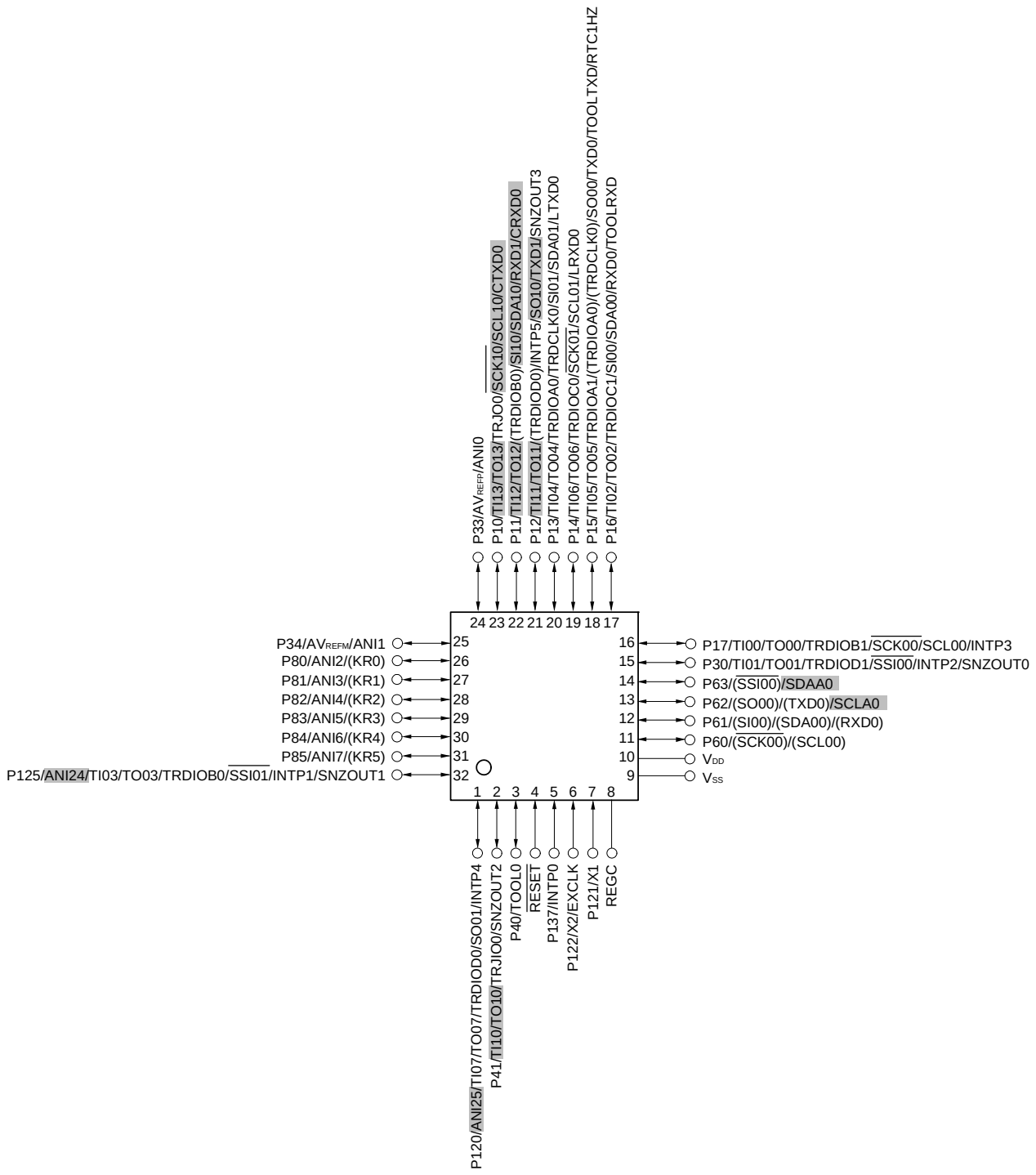



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.5.9 RL78/F13 Pin Configuration for 32-pin Products

- RL78/F13: 32-pin Plastic QFN

**Figure 1-31. RL78/F13 Pin Configuration for 32-pin Products**

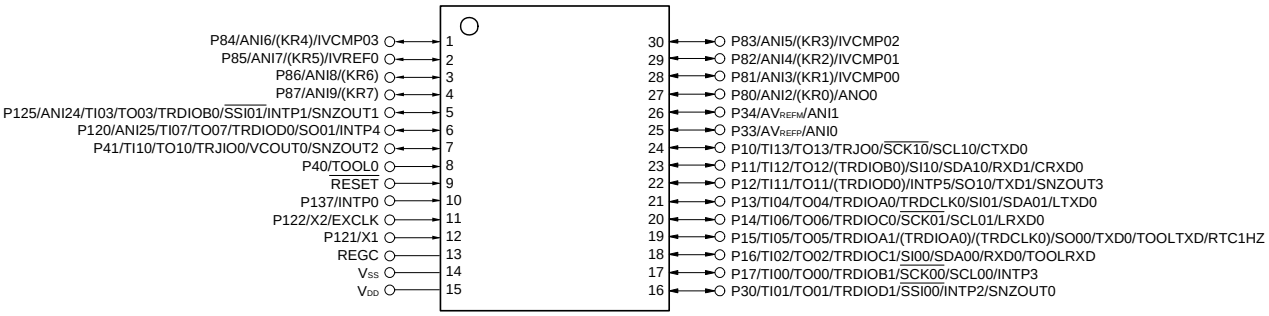



Cautions 1. : Shaded pins are not available in the RL78/F13 (LIN incorporated) products (R5F10ABA, R5F10ABC, R5F10ABD, R5F10ABE).
2. The available pins and functions differ depending on the product. Refer to **CHAPTER 2 PIN FUNCTIONS** for details.
3. Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.5.10 RL78/F14 Pin Configuration for 30-pin Products

- RL78/F14: 30-pin Plastic SSOP

**Figure 1-32. RL78/F14 Pin Configuration for 30-pin Products**

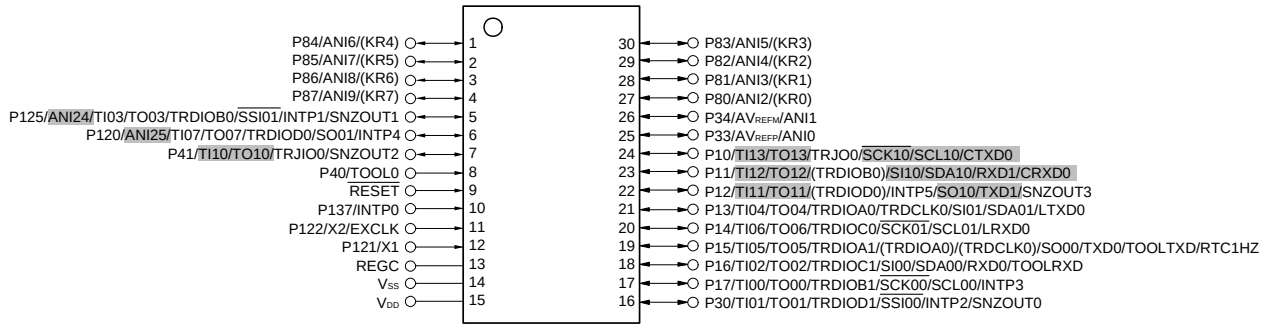



Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.5.11 RL78/F13 Pin Configuration for 30-pin Products

- RL78/F13: 30-pin Plastic SSOP

**Figure 1-33. RL78/F13 Pin Configuration for 30-pin Products**

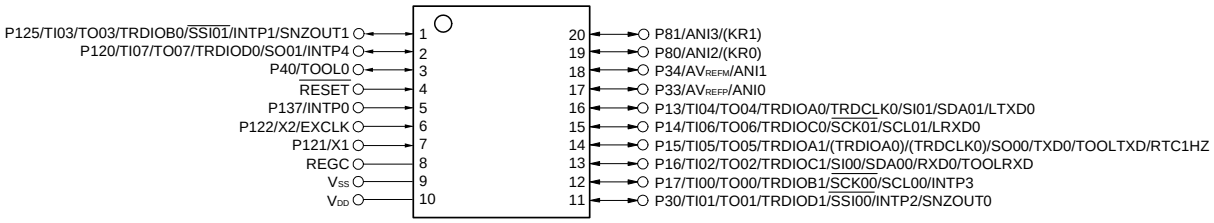



Cautions 1. : Shaded pins are not available in the RL78/F13 (LIN incorporated) products (R5F10AAA, R5F10AAC, R5F10AAD, R5F10AAE).
2. The available pins and functions differ depending on the product. Refer to **CHAPTER 2 PIN FUNCTIONS** for details.
3. Do not use the XT1 and XT2 pin functions in grade-Y products.

### 1.5.12 RL78/F13 Pin Configuration for 20-pin Products

- RL78/F13: 20-pin Plastic SSOP

**Figure 1-34. RL78/F13 Pin Configuration for 20-pin Products**




Caution Do not use the XT1 and XT2 pin functions in grade-Y products.

## 1.6 Order Information

Tables 1-7 to 1-9 show the order information for RL78/F14, RL78/F13 (CAN and LIN incorporated), and RL78/F13 (LIN incorporated).

**Table 1-7. Order Information for RL78/F14**

| Package | Device | Order Name |
|---|---|---|
| 30-pin plastic SSOP | Grade L | R5F10PADLSP, R5F10PAELSP |
| | Grade K | R5F10PADKSP, R5F10PAEKSP |
| | Grade Y | R5F10PADYSP, R5F10PAEYSP |
| 32-pin plastic VQFN | Grade L | R5F10PBDLNA, R5F10PBELNA |
| | Grade K | R5F10PBDKNA, R5F10PBEKNA |
| | Grade Y | R5F10PBDYNA, R5F10PBEYNA |
| 48-pin plastic LQFP | Grade L | R5F10PGDLFB, R5F10PGELFB, R5F10PGFLFB, R5F10PGGLFB, R5F10PGHLFB, R5F10PGJLFB |
| | Grade K | R5F10PGDKFB, R5F10PGEKFB, R5F10PGFKFB, R5F10PGGKFB, R5F10PGHKFB, R5F10PGJKFB |
| | Grade Y | R5F10PGDYFB, R5F10PGEYFB, R5F10PGFYFB, R5F10PGGYFB, R5F10PGHYFB, R5F10PGJYFB |
| 48-pin plastic VQFN | Grade L | R5F10PGDLNA, R5F10PGELNA, R5F10PGFLNA, R5F10PGGLNA, R5F10PGHLNA, R5F10PGJLNA |
| | Grade K | R5F10PGDKNA, R5F10PGEKNA, R5F10PGFKNA, R5F10PGGKNA, R5F10PGHKNA, R5F10PGJKNA |
| | Grade Y | R5F10PGDYNA, R5F10PGEYNA, R5F10PGFYNA, R5F10PGGYNA, R5F10PGHYNA, R5F10PGJYNA |
| 64-pin plastic LQFP | Grade L | R5F10PLELFB, R5F10PLFLFB, R5F10PLGLFB, R5F10PLHLFB, R5F10PLJLFB |
| | Grade K | R5F10PLEKFB, R5F10PLFKFB, R5F10PLGKFB, R5F10PLHKFB, R5F10PLJKFB |
| | Grade Y | R5F10PLEYFB, R5F10PLFYFB, R5F10PLGYFB, R5F10PLHYFB, R5F10PLJYFB |
| 80-pin plastic LQFP | Grade L | R5F10PMELFB, R5F10PMFLFB, R5F10PMGLFB, R5F10PMHLFB, R5F10PMJLFB |
| | Grade K | R5F10PMEKFB, R5F10PMFKFB, R5F10PMGKFB, R5F10PMHKFB, R5F10PMJKFB |
| | Grade Y | R5F10PMEYFB, R5F10PMFYFB, R5F10PMGYFB, R5F10PMHYFB, R5F10PMJYFB |
| 100-pin plastic LQFP | Grade L | R5F10PPELFB, R5F10PPFLFB, R5F10PPGLFB, R5F10PPHLFB, R5F10PPJLFB |
| | Grade K | R5F10PPEKFB, R5F10PPFKFB, R5F10PPGKFB, R5F10PPHKFB, R5F10PPJKFB |
| | Grade Y | R5F10PPEYFB, R5F10PPFYFB, R5F10PPGYFB, R5F10PPHYFB, R5F10PPJYFB |

**Table 1-8. Order Information for RL78/F13 (CAN and LIN incorporated)**

| Package | Device | Order Name |
|---|---|---|
| 30-pin plastic SSOP | Grade L | R5F10BACLSP, R5F10BADLSP, R5F10BAELSP, R5F10BAFLSP, R5F10BAGLSP |
| | Grade K | R5F10BACKSP, R5F10BADKSP, R5F10BAEKSP, R5F10BAFKSP, R5F10BAGKSP |
| | Grade Y | R5F10BACYSP, R5F10BADYSP, R5F10BAEYSP, R5F10BAFYSP, R5F10BAGYSP |
| 32-pin plastic VQFN | Grade L | R5F10BBCLNA, R5F10BBDLNA, R5F10BBELNA, R5F10BBFLNA, R5F10BBGLNA |
| | Grade K | R5F10BBCKNA, R5F10BBDKNA, R5F10BBEKNA, R5F10BBFKNA, R5F10BBGKNA |
| | Grade Y | R5F10BBCYNA, R5F10BBDYNA, R5F10BBEYNA, R5F10BBFYNA, R5F10BBGYNA |
| 48-pin plastic LQFP | Grade L | R5F10BGCLFB, R5F10BGDLFB, R5F10BGELFB, R5F10BGFLFB, R5F10BGGLFB |
| | Grade K | R5F10BGCKFB, R5F10BGDKFB, R5F10BGEKFB, R5F10BGFKFB, R5F10BGGKFB |
| | Grade Y | R5F10BGCYFB, R5F10BGDYFB, R5F10BGEYFB, R5F10BGFYFB, R5F10BGGYFB |
| 48-pin plastic VQFN | Grade L | R5F10BGCLNA, R5F10BGDLNA, R5F10BGELNA, R5F10BGFLNA, R5F10BGGLNA |
| | Grade K | R5F10BGCKNA, R5F10BGDKNA, R5F10BGEKNA, R5F10BGFKNA, R5F10BGGKNA |
| | Grade Y | R5F10BGCYNA, R5F10BGDYNA, R5F10BGEYNA, R5F10BGFYNA, R5F10BGGYNA |
| 64-pin plastic LQFP | Grade L | R5F10BLCLFB, R5F10BLDLFB, R5F10BLELFB, R5F10BLFLFB, R5F10BLGLFB |
| | Grade K | R5F10BLCKFB, R5F10BLDKFB, R5F10BLEKFB, R5F10BLFKFB, R5F10BLGKFB |
| | Grade Y | R5F10BLCYFB, R5F10BLDYFB, R5F10BLEYFB, R5F10BLFYFB, R5F10BLGYFB |
| 80-pin plastic LQFP | Grade L | R5F10BMELFB, R5F10BMFLFB, R5F10BMGLFB |
| | Grade K | R5F10BMEKFB, R5F10BMFKFB, R5F10BMGKFB |
| | Grade Y | R5F10BMEYFB, R5F10BMFYFB, R5F10BMGYFB |

**Table 1-9. Order Information for RL78/F13 (LIN incorporated)**

| Package | Device | Order Name |
|---|---|---|
| 20-pin plastic SSOP | Grade L | R5F10A6ALSP, R5F10A6CLSP, R5F10A6DLSP, R5F10A6ELSP |
| | Grade K | R5F10A6AKSP, R5F10A6CKSP, R5F10A6DKSP, R5F10A6EKSP |
| | Grade Y | R5F10A6AYSP, R5F10A6CYSP, R5F10A6DYSP, R5F10A6EYSP |
| 30-pin plastic SSOP | Grade L | R5F10AAALSP, R5F10AACLSP, R5F10AADLSP, R5F10AAELSP |
| | Grade K | R5F10AAAKSP, R5F10AACKSP, R5F10AADKSP, R5F10AAEKSP |
| | Grade Y | R5F10AAAYSP, R5F10AACYSP, R5F10AADYSP, R5F10AAEYSP |
| 32-pin plastic VQFN | Grade L | R5F10ABALNA, R5F10ABCLNA, R5F10ABDLNA, R5F10ABELNA |
| | Grade K | R5F10ABAKNA, R5F10ABCKNA, R5F10ABDKNA, R5F10ABEKNA |
| | Grade Y | R5F10ABAYNA, R5F10ABCYNA, R5F10ABDYNA, R5F10ABEYNA |
| 48-pin plastic LQFP | Grade L | R5F10AGALFB, R5F10AGCLFB, R5F10AGDLFB, R5F10AGELFB, R5F10AGFLFB, R5F10AGGLFB |
| | Grade K | R5F10AGAKFB, R5F10AGCKFB, R5F10AGDKFB, R5F10AGEKFB, R5F10AGFKFB, R5F10AGGKFB |
| | Grade Y | R5F10AGAYFB, R5F10AGCYFB, R5F10AGDYFB, R5F10AGEYFB, R5F10AGFYFB, R5F10AGGYFB |
| 48-pin plastic VQFN | Grade L | R5F10AGALNA, R5F10AGCLNA, R5F10AGDLNA, R5F10AGELNA, R5F10AGFLNA, R5F10AGGLNA |
| | Grade K | R5F10AGAKNA, R5F10AGCKNA, R5F10AGDKNA, R5F10AGEKNA, R5F10AGFKNA, R5F10AGGKNA |
| | Grade Y | R5F10AGAYNA, R5F10AGCYNA, R5F10AGDYNA, R5F10AGEYNA, R5F10AGFYNA, R5F10AGGYNA |
| 64-pin plastic LQFP | Grade L | R5F10ALCLFB, R5F10ALDLFB, R5F10ALELFB, R5F10ALFLFB, R5F10ALGLFB |
| | Grade K | R5F10ALCKFB, R5F10ALDKFB, R5F10ALEKFB, R5F10ALFKFB, R5F10ALGKFB |
| | Grade Y | R5F10ALCYFB, R5F10ALDYFB, R5F10ALEYFB, R5F10ALFYFB, R5F10ALGYFB |
| 80-pin plastic LQFP | Grade L | R5F10AMELFB, R5F10AMFLFB, R5F10AMGLFB |
| | Grade K | R5F10AMEKFB, R5F10AMFKFB, R5F10AMGKFB |
| | Grade Y | R5F10AMEYFB, R5F10AMFYFB, R5F10AMGYFB |

# CHAPTER 2 PIN FUNCTIONS

## 2.1 Pin Function List

Pin I/O buffer power supplies depend on the product. Table 2-1 shows the relationship between these power supplies and the pins. $EV_{DD}$ indicates $EV_{DD0}$ and $EV_{DD1}$.

**Table 2-1. Pin I/O Buffer Power Supplies**

**(1) 20-pin, 30-pin, 32-pin, and 48-pin products**

| Power Supply | Corresponding Pins |
|---|---|
| $V_{DD}$ | All pins |

**(2) 64-pin products**

| Power Supply | Corresponding Pins |
|---|---|
| $EV_{DD0}$ | • Port pins other than P33, P34, P80 to P87, P90 to P96 **Note**, P121 to P124, and P137 |
| $V_{DD}$ | • P33, P34, P80 to P87, P90 to P96 **Note**, P121 to P124, and P137<br>• Pins other than port pins |

**Note** In R5F10PLE, R5F10PLF, R5F10BLC, R5F10BLD, R5F10BLE, R5F10BLF, R5F10BLG, R5F10ALF, and R5F10ALG, the power supply for P96 is $EV_{DD0}$. In R5F10ALC, R5F10ALD, and R5F10ALE, the power supply for P92 to P97 is $EV_{DD0}$.

**(3) 80-pin products**

| Power Supply | Corresponding Pins |
|---|---|
| $EV_{DD0}$ | • Port pins other than P33, P34, P80 to P87, P90 to P97 **Note**, P121 to P124, and P137 |
| $V_{DD}$ | • P33, P34, P80 to P87, P90 to P97 **Note**, P121 to P124, and P137<br>• Pins other than port pins |

**Note** In R5F10PME, R5F10PMF, R5F10BME, R5F10BMF, R5F10BMG, R5F10AME, R5F10AMF, and R5F10AMG, the power supply for P96 and P97 is $EV_{DD0}$.

**(4) 100-pin products**

| Power Supply | Corresponding Pins |
|---|---|
| $EV_{DD0}$, $EV_{DD1}$ | • Port pins other than P33, P34, P80 to P87, P90 to P97, P100 to P105, P121 to P124, and P137 |
| $V_{DD}$ | • P33, P34, P80 to P87, P90 to P97, P100 to P105, P121 to P124, and P137<br>• Pins other than port pins |

The products are classified into the following five groups according to the product type, pin count, and code flash memory size.

Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory

Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory

Group C: RL78/F13 (CAN and LIN incorporated) products with 30, 32, 48, 64, or 80 pins and 32 Kbytes to 128 Kbytes of code flash memory

Group D: RL78/F14 products with 30, 32, 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory

Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

This subchapter describes the 100-pin products of RL78/F14 and the 80-pin products of RL78/F13 (CAN and LIN incorporated) as examples.

### 2.1.1 RL78/F14 100-pin products

(1/2)

| Function Name | I/O | Function | After Reset | Alternate Function |
|---|---|---|---|---|
| P00 | I/O | Port 0<br>Use of an on-chip pull-up resistor can be specified by a software setting. | Input port | (TI05)/(TO05)/INTP9 |
| P01 | | | | (TI04)/(TO04) |
| P02 | | | | (TI06)/(TO06) |
| P03 | | | | (RTC1HZ) |
| P10 | I/O | Port 1<br>Input of P10, P11, P13, P14, P16, and P17 can be set to TTL input buffer.<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>Output from P10 to P17 can be set to N-ch open-drain output.<br>For input to P10, P11, P13, P14, P16, and P17, the threshold level can be specified. | Input port | TI13/TO13/TRJO0/$\overline{\text{SCK10}}$/SCL10/ LTXD1/CTXD0 |
| P11 | | | | TI12/TO12/(TRDIOB0)/SI10/ SDA10/RXD1/LRXD1/CRXD0 |
| P12 | | | | TI11/TO11/(TRDIOD0)/INTP5/ SO10/TXD1/SNZOUT3 |
| P13 | | | | TI04/TO04/TRDIOA0/TRDCLK0/ SI01/SDA01/LTXD0 |
| P14 | | | | TI06/TO06/TRDIOC0/$\overline{\text{SCK01}}$/ SCL01/LRXD0 |
| P15 | | | | TI05/TO05/TRDIOA1/(TRDIOA0)/ (TRDCLK0)/SO00/TXD0/ TOOLTXD/RTC1HZ |
| P16 | | | | TI02/TO02/TRDIOC1/SI00/ SDA00/RXD0/TOOLRXD |
| P17 | | | | TI00/TO00/TRDIOB1/$\overline{\text{SCK00}}$/ SCL00/INTP3 |
| P30 | I/O | Port 3<br>Input of P30 can be set to TTL input buffer.<br>P33 and P34 can be set to analog input.<br>For input to P30 to P32, use of an on-chip pull-up resistor can be specified by a software setting.<br>For input to P30, the threshold level can be specified. | Input port | TI01/TO01/TRDIOD1/$\overline{\text{SSI00}}$/ INTP2/SNZOUT0 |
| P31 | | | | TI14/TO14/STOPST/(INTP2) |
| P32 | | | | TI16/TO16/INTP7 |
| P33 | | | Analog input port | AVREFP/ANI0 |
| P34 | | | | AVREFM/ANI1 |
| P40 | I/O | Port 4<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>For input to P43, the threshold level can be specified. | Input port | TOOL0 |
| P41 | | | | TI10/TO10/TRJIO0/VCOUT0/ SNZOUT2 |
| P42 | | | | (LTXD0) |
| P43 | | | | (LRXD0) |
| P44 | | | | (TI07)/(TO07) |
| P45 | | | | (TI10)/(TO10) |
| P46 | | | | (TI12)/(TO12) |
| P47 | | | | INTP13 |
| P50 | I/O | Port 5<br>Input of P54 can be set to TTL input buffer.<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>For input to P50 and P52 to P54, the threshold level can be specified. | Input port | ($\overline{\text{SSI01}}$)/(INTP3) |
| P51 | | | | (SO01)/INTP11 |
| P52 | | | | ($\overline{\text{SCK01}}$)/(STOPST) |
| P53 | | | | (SI01)/INTP10 |
| P54 | | | | (TI11)/(TO11)/$\overline{\text{SSI10}}$ |
| P55 | | | | (TI13)/(TO13) |
| P56 | | | | (TI15)/(TO15)/(SNZOUT1) |
| P57 | | | | (TI17)/(TO17)/(SNZOUT0) |
| P60 | I/O | Port 6<br>Input of P62 and P63 can be set to TTL input buffer.<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>Output from P60 to P63 can be set to N-ch open-drain output.<br>For input to P60 to P63, the threshold level can be specified. | Input port | ($\overline{\text{SCK00}}$)/(SCL00) |
| P61 | | | | (SI00)/(SDA00)/(RXD0) |
| P62 | | | | (SO00)/(TXD0)/SCLA0 |
| P63 | | | | ($\overline{\text{SSI00}}$)/SDAA0 |
| P64 | | | | (TI14)/(TO14)/(SNZOUT3) |
| P65 | | | | (TI16)/(TO16)/(SNZOUT2) |
| P66 | | | | (TI00)/(TO00) |
| P67 | | | | (TI02)/(TO02) |

**Remark** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIOR). Only the STOPST function of P52 can be assigned via settings in the STOP status output control register (STPSTC).

(2/2)

| Function Name | I/O | Function | After Reset | Alternate Function |
|---|---|---|---|---|
| P70 | I/O | Port 7<br>Input of P70, P71, and P73 can be set to TTL input buffer.<br>P70 to P74 can be set to analog input.<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>Output from P70 to P72 can be set to N-ch open-drain output.<br>For input to P70, P71, P73, and P75 to P77, the threshold level can be specified. | Analog input port | ANI26/KR0/TI15/TO15/INTP8/SI11/SDA11/SNZOUT4 |
| P71 | | | | $\overline{\text{ANI27}}$/KR1/TI17/TO17/INTP6/SCK11/SCL11/SNZOUT5 |
| P72 | | | | ANI28/KR2/(CTXD0)/SO11/SNZOUT6 |
| P73 | | | | ANI29/KR3/(CRXD0)/$\overline{\text{SSI11}}$/SNZOUT7 |
| P74 | | | | ANI30/KR4/(SO10)/(TXD1) |
| P75 | | | Input port | KR5/(SI10)/(RXD1) |
| P76 | | | | KR6/($\overline{\text{SCK10}}$) |
| P77 | | | | KR7/($\overline{\text{SSI10}}$)/INTP12 |
| P80 | I/O | Port 8<br>P80 to P87 can be set to analog input. | Analog input port | ANI2/ANO0 |
| P81 | | | | ANI3/IVCMP00 |
| P82 | | | | ANI4/IVCMP01 |
| P83 | | | | ANI5/IVCMP02 |
| P84 | | | | ANI6/IVCMP03 |
| P85 | | | | ANI7/IVREF0 |
| P86 | | | | ANI8 |
| P87 | | | | ANI9 |
| P90 | I/O | Port 9<br>P90 to P97 can be set to analog input. | Analog input port | ANI10 |
| P91 | | | | ANI11 |
| P92 | | | | ANI12 |
| P93 | | | | ANI13 |
| P94 | | | | ANI14 |
| P95 | | | | ANI15 |
| P96 | | | | ANI16 |
| P97 | | | | ANI17 |
| P100 | I/O | Port 10<br>P100 to P105 can be set to analog input.<br>For P106 and P107, use of an on-chip pull-up resistor can be specified by a software setting.<br>For input to P107, the threshold level can be specified. | Analog input port | ANI18 |
| P101 | | | | ANI19 |
| P102 | | | | ANI20 |
| P103 | | | | ANI21 |
| P104 | | | | ANI22 |
| P105 | | | | ANI23 |
| P106 | | | Input port | (LTXD1) |
| P107 | | | | (LRXD1) |
| P120 | I/O | Port 12<br>Input of P125 can be set to TTL input buffer.<br>P120 and P125 can be set to analog input.<br>For P120 and P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting.<br>Output from P120 can be set to N-ch open-drain output.<br>For input to P125, the threshold level can be specified. | Analog input port | ANI25/TI07/TO07/TRDIOD0/SO01/INTP4 |
| P121 | Input | | Input port | X1 |
| P122 | | | | X2/EXCLK |
| P123 | | | | XT1 |
| P124 | | | | XT2/EXCLKS |
| P125 | I/O | | Analog input port | $\overline{\text{ANI24}}$/TI03/TO03/TRDIOB0/$\overline{\text{SSI01}}$/INTP1/SNZOUT1 |
| P126 | | | Input port | (TI01)/(TO01) |
| P127 | | | | (TI03)/(TO03) |
| P130 | Output | Port 13 | Output port | RESOUT |
| P137 | Input | | Input port | INTP0 |
| P140 | I/O | Port 14<br>Use of an on-chip pull-up resistor can be specified by a software setting. | Input port | PCLBUZ0 |
| P150 | I/O | Port 15<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>For input to P150, P152, and P153, the threshold level can be specified. | Input port | ($\overline{\text{SSI11}}$) |
| P151 | | | | (SO11) |
| P152 | | | | (SI11) |
| P153 | | | | ($\overline{\text{SCK11}}$) |
| P154 | | | | (SNZOUT7) |
| P155 | | | | (SNZOUT6) |
| P156 | | | | (SNZOUT5) |
| P157 | | | | (SNZOUT4) |

**Remark** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIOR).

### 2.1.2 RL78/F13 (CAN and LIN incorporated) 80-pin products

(1/2)

| Function Name | I/O | Function | After Reset | Alternate Function |
|---|---|---|---|---|
| P00 | I/O | Port 0<br>Use of an on-chip pull-up resistor can be specified by a software setting. | Input port | (TI05)/(TO05)/INTP9 |
| P01 | | | | (TI04)/(TO04) |
| P02 | | | | (TI06)/(TO06) |
| P10 | I/O | Port 1<br>Input of P10, P11, P13, P14, P16, and P17 can be set to TTL input buffer.<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>Output from P10 to P17 can be set to N-ch open-drain output.<br>For input to P10, P11, P13, P14, P16, and P17, the threshold level can be specified. | Input port | TI13/TO13/TRJO0/$\overline{\text{SCK10}}$/SCL10/CTXD0 |
| P11 | | | | TI12/TO12/(TRDIOB0)/SI10/SDA10/RXD1/CRXD0 |
| P12 | | | | TI11/TO11/(TRDIOD0)/INTP5/SO10/TXD1/SNZOUT3 |
| P13 | | | | TI04/TO04/TRDIOA0/TRDCLK0/SI01/SDA01/LTXD0 |
| P14 | | | | TI06/TO06/TRDIOC0/$\overline{\text{SCK01}}$/SCL01/LRXD0 |
| P15 | | | | TI05/TO05/TRDIOA1/(TRDIOA0)/(TRDCLK0)/SO00/TXD0/TOOLTXD/RTC1HZ |
| P16 | | | | TI02/TO02/TRDIOC1/SI00/SDA00/RXD0/TOOLRXD |
| P17 | | | | TI00/TO00/TRDIOB1/$\overline{\text{SCK00}}$/SCL00/INTP3 |
| P30 | I/O | Port 3<br>Input of P30 can be set to TTL input buffer.<br>P33 and P34 can be set to analog input.<br>For input to P30 to P32, use of an on-chip pull-up resistor can be specified by a software setting.<br>For input to P30, the threshold level can be specified. | Input port | TI01/TO01/TRDIOD1/$\overline{\text{SSI00}}$/INTP2/SNZOUT0 |
| P31 | | | | STOPST/(INTP2) |
| P32 | | | | INTP7 |
| P33 | | | Analog input port | $AV_{REFP}$/ANI0 |
| P34 | | | | $AV_{REFM}$/ANI1 |
| P40 | I/O | Port 4<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>For input to P43, the threshold level can be specified. | Input port | TOOL0 |
| P41 | | | | TI10/TO10/TRJIO0/SNZOUT2 |
| P42 | | | | (LTXD0) |
| P43 | | | | (LRXD0) |
| P44 | | | | (TI07)/(TO07) |
| P45 | | | | (TI10)/(TO10) |
| P46 | | | | (TI12)/(TO12) |
| P47 | | | | — |
| P50 | I/O | Port 5<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>For input to P50 and P52 to P54, the threshold level can be specified.<br>Input of P54 can be set to TTL input buffer. | Input port | ($\overline{\text{SSI01}}$)/(INTP3) |
| P51 | | | | (SO01)/INTP11 |
| P52 | | | | ($\overline{\text{SCK01}}$)/(STOPST) |
| P53 | | | | (SI01)/INTP10 |
| P54 | | | | (TI11)/(TO11)/$\overline{\text{SSI10}}$ |
| P55 | | | | (TI13)/(TO13) |
| P56 | | | | (SNZOUT1) |
| P57 | | | | (SNZOUT0) |
| P60 | I/O | Port 6<br>Input of P62 and P63 can be set to TTL input buffer.<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>Output from P60 to P63 can be set to N-ch open-drain output.<br>For input to P60 to P63, the threshold level can be specified. | Input port | ($\overline{\text{SCK00}}$)/(SCL00) |
| P61 | | | | (SI00)/(SDA00)/(RXD0) |
| P62 | | | | (SO00)/(TXD0)/SCLA0 |
| P63 | | | | ($\overline{\text{SSI00}}$)/SDAA0 |
| P64 | | | | (SNZOUT3) |
| P65 | | | | (SNZOUT2) |
| P66 | | | | (TI00)/(TO00) |
| P67 | | | | (TI02)/(TO02) |

**Remark** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIOR). Only the STOPST function of P52 can be assigned via settings in the STOP status output control register (STPSTC).

(2/2)

| Function Name | I/O | Function | After Reset | Alternate Function |
|---|---|---|---|---|
| P70 | I/O | Port 7<br>Input of P70, P71, and P73 can be set to TTL input buffer.<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>Output from P70 to P72 can be set to N-ch open-drain output.<br>For input to P70, P71, P73, and P75 to P77, the threshold level can be specified. | Input port | KR0/INTP8/SI11/SDA11/ SNZOUT4 |
| P71 | | | | KR1/INTP6/$\overline{\text{SCK11}}$/SCL11/ SNZOUT5 |
| P72 | | | | KR2/(CTXD0)/SO11/SNZOUT6 |
| P73 | | | | KR3/(CRXD0)/$\overline{\text{SSI11}}$/SNZOUT7 |
| P74 | | | | KR4/(SO10)/(TXD1) |
| P75 | | | | KR5/(SI10)/(RXD1) |
| P76 | | | | KR6/($\overline{\text{SCK10}}$) |
| P77 | | | | KR7/($\overline{\text{SSI10}}$) |
| P80 | I/O | Port 8<br>P80 to P87 can be set to analog input. | Analog input port | ANI2 |
| P81 | | | | ANI3 |
| P82 | | | | ANI4 |
| P83 | | | | ANI5 |
| P84 | | | | ANI6 |
| P85 | | | | ANI7 |
| P86 | | | | ANI8 |
| P87 | | | | ANI9 |
| P90 | I/O | Port 9<br>P90 to P97 can be set to analog input.<br>For P96 and P97, use of an on-chip pull-up resistor can be specified by a software setting. | Analog input port | ANI10 |
| P91 | | | | ANI11 |
| P92 | | | | ANI12 |
| P93 | | | | ANI13 |
| P94 | | | | ANI14 |
| P95 | | | | ANI15 |
| P96 | | | | ANI26 |
| P97 | | | | ANI27 |
| P120 | I/O | Port 12<br>Input of P125 can be set to TTL input buffer.<br>P120 and P125 can be set to analog input.<br>For P120, P125, and P126, use of an on-chip pull-up resistor can be specified by a software setting.<br>Output from P120 can be set to N-ch open-drain output.<br>For input to P125, the threshold level can be specified. | Analog input port | ANI25/TI07/TO07/TRDIOD0/ SO01/INTP4 |
| P121 | Input | | Input port | X1 |
| P122 | | | | X2/EXCLK |
| P123 | | | | XT1 |
| P124 | | | | XT2/EXCLKS |
| P125 | I/O | | Analog input port | ANI24/TI03/TO03/TRDIOB0/ $\overline{\text{SSI01}}$/INTP1/SNZOUT1 |
| P126 | | | Input port | (TI01)/(TO01) |
| P130 | Output | Port 13 | Output port | RESOUT |
| P137 | Input | | Input port | INTP0 |
| P140 | I/O | Port 14<br>Use of an on-chip pull-up resistor can be specified by a software setting. | Input port | PCLBUZ0 |

**Remark** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIOR).

### 2.1.3 RL78/F13 (LIN incorporated) 80-pin products

(1/2)

| Function Name | I/O | Function | After Reset | Alternate Function |
|---|---|---|---|---|
| P00 | I/O | Port 0<br>Use of an on-chip pull-up resistor can be specified by a software setting. | Input port | (TI05)/(TO05)/INTP9 |
| P01 | | | | (TI04)/(TO04) |
| P02 | | | | (TI06)/(TO06) |
| P10 | I/O | Port 1<br>Input of P10, P11, P13, P14, P16, and P17 can be set to TTL input buffer.<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>Output from P10 to P17 can be set to N-ch open-drain output.<br>For input to P10, P11, P13, P14, P16, and P17, the threshold level can be specified. | Input port | TI13/TO13/TRJO0/$\overline{\text{SCK10}}$/SCL10 |
| P11 | | | | TI12/TO12/(TRDIOB0)/SI10/SDA10/RXD1 |
| P12 | | | | TI11/TO11/(TRDIOD0)/INTP5/SO10/TXD1/SNZOUT3 |
| P13 | | | | TI04/TO04/TRDIOA0/TRDCLK0/SI01/SDA01/LTXD0 |
| P14 | | | | TI06/TO06/TRDIOC0/$\overline{\text{SCK01}}$/SCL01/LRXD0 |
| P15 | | | | TI05/TO05/TRDIOA1/(TRDIOA0)/(TRDCLK0)/SO00/TXD0/TOOLTXD/RTC1HZ |
| P16 | | | | TI02/TO02/TRDIOC1/SI00/SDA00/RXD0/TOOLRXD |
| P17 | | | | TI00/TO00/TRDIOB1/$\overline{\text{SCK00}}$/SCL00/INTP3 |
| P30 | I/O | Port 3<br>Input of P30 can be set to TTL input buffer.<br>P33 and P34 can be set to analog input.<br>For input to P30 to P32, use of an on-chip pull-up resistor can be specified by a software setting.<br>For input to P30, the threshold level can be specified. | Input port | TI01/TO01/TRDIOD1/$\overline{\text{SSI00}}$/INTP2/SNZOUT0 |
| P31 | | | | STOPST/(INTP2) |
| P32 | | | | INTP7 |
| P33 | | | Analog input port | AVREFP/ANI0 |
| P34 | | | | AVREFM/ANI1 |
| P40 | I/O | Port 4<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>For input to P43, the threshold level can be specified. | Input port | TOOL0 |
| P41 | | | | TI10/TO10/TRJIO0/SNZOUT2 |
| P42 | | | | (LTXD0) |
| P43 | | | | (LRXD0) |
| P44 | | | | (TI07)/(TO07) |
| P45 | | | | (TI10)/(TO10) |
| P46 | | | | (TI12)/(TO12) |
| P47 | | | | — |
| P50 | I/O | Port 5<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>For input to P50 and P52 to P54, the threshold level can be specified.<br>Input of P54 can be set to TTL input buffer. | Input port | ($\overline{\text{SSI01}}$)/(INTP3) |
| P51 | | | | (SO01)/INTP11 |
| P52 | | | | ($\overline{\text{SCK01}}$)/(STOPST) |
| P53 | | | | (SI01)/INTP10 |
| P54 | | | | (TI11)/(TO11)/$\overline{\text{SSI10}}$ |
| P55 | | | | (TI13)/(TO13) |
| P56 | | | | (SNZOUT1) |
| P57 | | | | (SNZOUT0) |
| P60 | I/O | Port 6<br>Input of P62 and P63 can be set to TTL input buffer.<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>Output from P60 to P63 can be set to N-ch open-drain output.<br>For input to P60 to P63, the threshold level can be specified. | Input port | ($\overline{\text{SCK00}}$)/(SCL00) |
| P61 | | | | (SI00)/(SDA00)/(RXD0) |
| P62 | | | | (SO00)/(TXD0)/SCLA0 |
| P63 | | | | ($\overline{\text{SSI00}}$)/SDAA0 |
| P64 | | | | (SNZOUT3) |
| P65 | | | | (SNZOUT2) |
| P66 | | | | (TI00)/(TO00) |
| P67 | | | | (TI02)/(TO02) |

**Remark** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIOR). Only the STOPST function of P52 can be assigned via settings in the STOP status output control register (STPSTC).

(2/2)

| Function Name | I/O | Function | After Reset | Alternate Function |
|---|---|---|---|---|
| P70 | I/O | Port 7<br>Input of P70, P71, and P73 can be set to TTL input buffer.<br>Use of an on-chip pull-up resistor can be specified by a software setting.<br>Output from P70 to P72 can be set to N-ch open-drain output.<br>For input to P70, P71, P73, and P75 to P77, the threshold level can be specified. | Input port | KR0/INTP8/SI11/SDA11/SNZOUT4 |
| P71 | | | | KR1/INTP6/$\overline{\text{SCK11}}$/SCL11/SNZOUT5 |
| P72 | | | | KR2/SO11/SNZOUT6 |
| P73 | | | | KR3/$\overline{\text{SSI11}}$/SNZOUT7 |
| P74 | | | | KR4/(SO10)/(TXD1) |
| P75 | | | | KR5/(SI10)/(RXD1) |
| P76 | | | | KR6/($\overline{\text{SCK10}}$) |
| P77 | | | | KR7/($\overline{\text{SSI10}}$) |
| P80 | I/O | Port 8<br>P80 to P87 can be set to analog input. | Analog input port | ANI2 |
| P81 | | | | ANI3 |
| P82 | | | | ANI4 |
| P83 | | | | ANI5 |
| P84 | | | | ANI6 |
| P85 | | | | ANI7 |
| P86 | | | | ANI8 |
| P87 | | | | ANI9 |
| P90 | I/O | Port 9<br>P90 to P97 can be set to analog input.<br>For P96 and P97, use of an on-chip pull-up resistor can be specified by a software setting. | Analog input port | ANI10 |
| P91 | | | | ANI11 |
| P92 | | | | ANI12 |
| P93 | | | | ANI13 |
| P94 | | | | ANI14 |
| P95 | | | | ANI15 |
| P96 | | | | ANI26 |
| P97 | | | | ANI27 |
| P120 | I/O | Port 12<br>Input of P125 can be set to TTL input buffer.<br>P120 and P125 can be set to analog input.<br>For P120, P125, and P126, use of an on-chip pull-up resistor can be specified by a software setting.<br>Output from P120 can be set to N-ch open-drain output.<br>For input to P125, the threshold level can be specified. | Analog input port | ANI25/TI07/TO07/TRDIOD0/SO01/INTP4 |
| P121 | Input | | Input port | X1 |
| P122 | | | | X2/EXCLK |
| P123 | | | | XT1 |
| P124 | | | | XT2/EXCLKS |
| P125 | I/O | | Analog input port | ANI24/TI03/TO03/TRDIOB0/$\overline{\text{SSI01}}$/INTP1/SNZOUT1 |
| P126 | | | Input port | (TI01)/(TO01) |
| P130 | Output | Port 13 | Output port | RESOUT |
| P137 | Input | | Input port | INTP0 |
| P140 | I/O | Port 14<br>Use of an on-chip pull-up resistor can be specified by a software setting. | Input port | PCLBUZ0 |

**Remark** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIOR).

### 2.1.4 Pins for each product (pins other than port pins)

This subchapter shows the pins other than the ports shown in tables 2-2 to 2-4 for each product.
√ indicates the pin that is provided in the product and — indicates the pin that is not provided.

**Table 2-2. List of RL78/F14 Pins Other than Port Pins (1/5)**

| Pin Function | I/O | Function | Pin Count | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 100-pin | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin |
| ANI0 | Input | A/D converter analog input ($V_{DD}$ connection) | √ | √ | √ | √ | √ | √ |
| ANI1 | Input | | √ | √ | √ | √ | √ | √ |
| ANI2 | Input | | √ | √ | √ | √ | √ | √ |
| ANI3 | Input | | √ | √ | √ | √ | √ | √ |
| ANI4 | Input | | √ | √ | √ | √ | √ | √ |
| ANI5 | Input | | √ | √ | √ | √ | √ | √ |
| ANI6 | Input | | √ | √ | √ | √ | √ | √ |
| ANI7 | Input | | √ | √ | √ | √ | √ | √ |
| ANI8 | Input | | √ | √ | √ | √ | — | √ |
| ANI9 | Input | | √ | √ | √ | √ | — | √ |
| ANI10 | Input | | √ | √ | √ | √ | — | — |
| ANI11 | Input | | √ | √ | √ | √ | — | — |
| ANI12 | Input | | √ | √ | √ | √ | — | — |
| ANI13 | Input | | √ | √ | √ | — | — | — |
| ANI14 | Input | | √ | √ | √ | — | — | — |
| ANI15 | Input | | √ | √ | √ | — | — | — |
| ANI16 | Input | | √ | √ Note 1 | √ Note 1 | — | — | — |
| ANI17 | Input | | √ | √ Note 1 | — | — | — | — |
| ANI18 | Input | | √ | — | — | — | — | — |
| ANI19 | Input | | √ | — | — | — | — | — |
| ANI20 | Input | | √ | — | — | — | — | — |
| ANI21 | Input | | √ | — | — | — | — | — |
| ANI22 | Input | | √ | — | — | — | — | — |
| ANI23 | Input | | √ | — | — | — | — | — |
| ANI24 | Input | A/D converter analog input ($EV_{DD}$ connection) | √ | √ | √ | √ | √ | √ |
| ANI25 | Input | | √ | √ | √ | √ | √ | √ |
| ANI26 | Input | | √ | √ | √ | √ Note 1 | — | — |
| ANI27 | Input | | √ | √ | — | √ Note 1 | — | — |
| ANI28 | Input | | √ | √ Note 1 | — | √ Note 1 | — | — |
| ANI29 | Input | | √ | √ Note 1 | — | — | — | — |
| ANI30 | Input | | √ | √ Note 1 | — | — | — | — |
| IVCMP00 | Input | Comparator analog voltage input | √ | √ | √ | √ | √ | √ |
| IVCMP01 | Input | | √ | √ | √ | √ | √ | √ |
| IVCMP02 | Input | | √ | √ | √ | √ | √ | √ |
| IVCMP03 | Input | | √ | √ | √ | √ | √ | √ |
| IVREF0 | Input | Comparator reference voltage input | √ | √ | √ | √ | √ | √ |

**Note 1.** Provided only in Group E products.

**Table 2-2. List of RL78/F14 Pins Other than Port Pins (2/5)**

| Pin Function | I/O | Function | Pin Count | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 100-pin | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin |
| KR0 | Input | Key interrupt input | √ | √ | √ | √ | √ | √ |
| KR1 | Input | | √ | √ | √ | √ | √ | √ |
| KR2 | Input | | √ | √ | √ | √ | √ | √ |
| KR3 | Input | | √ | √ | √ | √ | √ | √ |
| KR4 | Input | | √ | √ | √ | √ | √ | √ |
| KR5 | Input | | √ | √ | √ | √ | √ | √ |
| KR6 | Input | | √ | √ | √ | √ | — | √ |
| KR7 | Input | | √ | √ | √ | √ | — | √ |
| ANO0 | Output | D/A converter output | √ | √ | √ | √ | √ | √ |
| VCOUT0 | Output | Comparator output | √ | √ | √ | √ | √ | √ |
| TI00 | Input | 16-bit timer 00 input | √ | √ | √ | √ | √ | √ |
| TI01 | Input | 16-bit timer 01 input (8-bit mode available) | √ | √ | √ | √ | √ | √ |
| TI02 | Input | 16-bit timer 02 input | √ | √ | √ | √ | √ | √ |
| TI03 | Input | 16-bit timer 03 input (8-bit mode available) | √ | √ | √ | √ | √ | √ |
| TI04 | Input | 16-bit timer 04 input | √ | √ | √ | √ | √ | √ |
| TI05 | Input | 16-bit timer 05 input | √ | √ | √ | √ | √ | √ |
| TI06 | Input | 16-bit timer 06 input | √ | √ | √ | √ | √ | √ |
| TI07 | Input | 16-bit timer 07 input | √ | √ | √ | √ | √ | √ |
| TI10 | Input | 16-bit timer 10 input | √ | √ | √ | √ | √ | √ |
| TI11 | Input | 16-bit timer 11 input (8-bit mode available) | √ | √ | √ | √ | √ | √ |
| TI12 | Input | 16-bit timer 12 input | √ | √ | √ | √ | √ | √ |
| TI13 | Input | 16-bit timer 13 input (8-bit mode available) | √ | √ | √ | √ | √ | √ |
| TI14 | Input | 16-bit timer 14 input | √ | √ Note 1 | √ Note 1 | √ Note 1 | — | — |
| TI15 | Input | 16-bit timer 15 input | √ | √ Note 1 | √ Note 1 | √ Note 1 | — | — |
| TI16 | Input | 16-bit timer 16 input | √ | √ Note 1 | √ Note 1 | √ Note 1 | — | — |
| TI17 | Input | 16-bit timer 17 input | √ | √ Note 1 | √ Note 1 | √ Note 1 | — | — |
| TO00 | Output | 16-bit timer 00 output | √ | √ | √ | √ | √ | √ |
| TO01 | Output | 16-bit timer 01 output (8-bit mode available) | √ | √ | √ | √ | √ | √ |
| TO02 | Output | 16-bit timer 02 output | √ | √ | √ | √ | √ | √ |
| TO03 | Output | 16-bit timer 03 output (8-bit mode available) | √ | √ | √ | √ | √ | √ |
| TO04 | Output | 16-bit timer 04 output | √ | √ | √ | √ | √ | √ |
| TO05 | Output | 16-bit timer 05 output | √ | √ | √ | √ | √ | √ |
| TO06 | Output | 16-bit timer 06 output | √ | √ | √ | √ | √ | √ |
| TO07 | Output | 16-bit timer 07 output | √ | √ | √ | √ | √ | √ |
| TO10 | Output | 16-bit timer 10 output | √ | √ | √ | √ | √ | √ |
| TO11 | Output | 16-bit timer 11 output (8-bit mode available) | √ | √ | √ | √ | √ | √ |
| TO12 | Output | 16-bit timer 12 output | √ | √ | √ | √ | √ | √ |
| TO13 | Output | 16-bit timer 13 output (8-bit mode available) | √ | √ | √ | √ | √ | √ |
| TO14 | Output | 16-bit timer 14 output | √ | √ Note 1 | √ Note 1 | √ Note 1 | — | — |
| TO15 | Output | 16-bit timer 15 output | √ | √ Note 1 | √ Note 1 | √ Note 1 | — | — |
| TO16 | Output | 16-bit timer 16 output | √ | √ Note 1 | √ Note 1 | √ Note 1 | — | — |
| TO17 | Output | 16-bit timer 17 output | √ | √ Note 1 | √ Note 1 | √ Note 1 | — | — |

**Note 1.** Provided only in Group E products.

**Table 2-2. List of RL78/F14 Pins Other than Port Pins (3/5)**

| Pin Function | I/O | Function | Pin Count | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 100-pin | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin |
| TRJIO0 | I/O | Timer RJ input/output | √ | √ | √ | √ | √ | √ |
| TRJO0 | Output | Timer RJ output | √ | √ | √ | √ | √ | √ |
| TRDCLK0 | Input | Timer RD external clock input | √ | √ | √ | √ | √ | √ |
| TRDIOA0 | I/O | Timer RD0 input/output | √ | √ | √ | √ | √ | √ |
| TRDIOB0 | I/O | | √ | √ | √ | √ | √ | √ |
| TRDIOC0 | I/O | | √ | √ | √ | √ | √ | √ |
| TRDIOD0 | I/O | | √ | √ | √ | √ | √ | √ |
| TRDIOA1 | I/O | Timer RD1 input/output | √ | √ | √ | √ | √ | √ |
| TRDIOB1 | I/O | | √ | √ | √ | √ | √ | √ |
| TRDIOC1 | I/O | | √ | √ | √ | √ | √ | √ |
| TRDIOD1 | I/O | | √ | √ | √ | √ | √ | √ |
| RXD0 | Input | Serial data input to UART0 | √ | √ | √ | √ | √ | √ |
| RXD1 | Input | Serial data input to UART1 | √ | √ | √ | √ | √ | √ |
| TXD0 | Output | Serial data output from UART0 | √ | √ | √ | √ | √ | √ |
| TXD1 | Output | Serial data output from UART1 | √ | √ | √ | √ | √ | √ |
| SCLA0 | I/O | Clock input/output for IICA0 | √ | √ | √ | √ | √ | — |
| SCL00 | Output | Clock output from simplified $I^2C$ | √ | √ | √ | √ | √ | √ |
| SCL01 | Output | | √ | √ | √ | √ | √ | √ |
| SCL10 | Output | | √ | √ | √ | √ | √ | √ |
| SCL11 | Output | | √ | √ | √ | √ | — | — |
| SDAA0 | I/O | Serial data input/output for IICA0 | √ | √ | √ | √ | √ | — |
| SDA00 | I/O | Serial data input/output for simplified $I^2C$ | √ | √ | √ | √ | √ | √ |
| SDA01 | I/O | | √ | √ | √ | √ | √ | √ |
| SDA10 | I/O | | √ | √ | √ | √ | √ | √ |
| SDA11 | I/O | | √ | √ | √ | √ | — | — |
| $\overline{SCK00}$ | I/O | Clock input/output for CSI00 | √ | √ | √ | √ | √ | √ |
| $\overline{SCK01}$ | I/O | Clock input/output for CSI01 | √ | √ | √ | √ | √ | √ |
| $\overline{SCK10}$ | I/O | Clock input/output for CSI10 | √ | √ | √ | √ | √ | √ |
| $\overline{SCK11}$ | I/O | Clock input/output for CSI11 | √ | √ | √ | √ | — | — |
| SI00 | Input | Serial data input to CSI00 | √ | √ | √ | √ | √ | √ |
| SI01 | Input | Serial data input to CSI01 | √ | √ | √ | √ | √ | √ |
| SI10 | Input | Serial data input to CSI10 | √ | √ | √ | √ | √ | √ |
| SI11 | Input | Serial data input to CSI11 | √ | √ | √ | √ | — | — |
| SO00 | Output | Serial data output from CSI00 | √ | √ | √ | √ | √ | √ |
| SO01 | Output | Serial data output from CSI01 | √ | √ | √ | √ | √ | √ |
| SO10 | Output | Serial data output from CSI10 | √ | √ | √ | √ | √ | √ |
| SO11 | Output | Serial data output from CSI11 | √ | √ | √ | √ | — | — |
| $\overline{SSI00}$ | Input | Slave select input to CSI00 (SPI00) | √ | √ | √ | √ | √ | √ |
| $\overline{SSI01}$ | Input | Slave select input to CSI01 (SPI01) | √ | √ | √ | √ | √ | √ |
| $\overline{SSI10}$ | Input | Slave select input to CSI10 (SPI10) | √ | √ | √ | — | — | — |
| $\overline{SSI11}$ | Input | Slave select input to CSI11 (SPI11) | √ | √ | √ | √ | — | — |
| CRXD0 | Input | Serial data input to CAN | √ | √ | √ | √ | √ | √ |
| CTXD0 | Output | Serial data output from CAN | √ | √ | √ | √ | √ | √ |

**able 2-2. List of RL78/F14 Pins Other than Port Pins (4/5)**

| Pin Function | I/O | Function | Pin Count | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 100-pin | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin |
| LRXD0 | Input | Serial data input to LIN | √ | √ | √ | √ | √ | √ |
| LRXD1 | Input | | √ | √ Note 1 | √ Note 1 | √ Note 1 | — | — |
| LTXD0 | Output | Serial data output from LIN | √ | √ | √ | √ | √ | √ |
| LTXD1 | Output | | √ | √ Note 1 | √ Note 1 | √ Note 1 | — | — |
| INTP0 | Input | External interrupt input | √ | √ | √ | √ | √ | √ |
| INTP1 | Input | | √ | √ | √ | √ | √ | √ |
| INTP2 | Input | | √ | √ | √ | √ | √ | √ |
| INTP3 | Input | | √ | √ | √ | √ | √ | √ |
| INTP4 | Input | | √ | √ | √ | √ | √ | √ |
| INTP5 | Input | | √ | √ | √ | √ | √ | √ |
| INTP6 | Input | | √ | √ | √ | √ | — | — |
| INTP7 | Input | | √ | √ | √ | √ | — | — |
| INTP8 | Input | | √ | √ | √ | √ | — | — |
| INTP9 | Input | | √ | √ | √ | √ | — | — |
| INTP10 | Input | | √ | √ | √ | — | — | — |
| INTP11 | Input | | √ | √ | √ | — | — | — |
| INTP12 | Input | | √ | √ Note 1 | √ Note 1 | — | — | — |
| INTP13 | Input | | √ | √ Note 1 | — | — | — | — |
| PCLBUZ0 | Output | Clock output/buzzer output 0 | √ | √ | √ | √ | — | — |
| RESOUT | Output | Reset output | √ | √ | √ | √ | — | — |
| STOPST | Output | STOP status output | √ | √ | √ | √ | — | — |
| SNZOUT0 | Output | SNOOZE status output | √ | √ | √ | √ | √ | √ |
| SNZOUT1 | Output | | √ | √ | √ | √ | √ | √ |
| SNZOUT2 | Output | | √ | √ | √ | √ | √ | √ |
| SNZOUT3 | Output | | √ | √ | √ | √ | √ | √ |
| SNZOUT4 | Output | | √ | √ | √ | √ | — | — |
| SNZOUT5 | Output | | √ | √ | √ | √ | — | — |
| SNZOUT6 | Output | | √ | √ | √ | √ | — | — |
| SNZOUT7 | Output | | √ | √ | √ | √ | — | — |
| RTC1HZ | Output | Real-time clock correction clock (1 Hz) output | √ | √ | √ | √ | √ | √ |
| EXCLK | Input | External clock input for main system clock | √ | √ | √ | √ | √ | √ |
| EXCLKS | Input | External clock input for subsystem clock | √ | √ | √ | √ | — | — |
| X1 | — | Resonator connection for main system clock | √ | √ | √ | √ | √ | √ |
| X2 | — | | √ | √ | √ | √ | √ | √ |
| XT1 Note 3 | — | Resonator connection for subsystem clock | √ | √ | √ | √ | — | — |
| XT2 Note 3 | — | | √ | √ | √ | √ | — | — |
| $\overline{\text{RESET}}$ | Input | External reset input | √ | √ | √ | √ | √ | √ |
| REGC | — | Regulator output stabilization capacitance connection for internal operation. Connect to Vss via the capacitor (0.47 to 1 μF). | √ | √ | √ | √ | √ | √ |
| $V_{DD}$ | — | Positive power supply for the P33, P34, P80 to P87, P90 to P97 Note 2, P100 to P105, P121 to P124, P137, and $\overline{\text{RESET}}$ pins | √ | √ | √ | √ | √ | √ |

**Notes 1.** Provided only in Group E products.

**2.** In products of Groups A to D, the positive power supply for P96 and P97 is $EV_{DD0}$.

**3.** Do not use the XT1 and XT2 pin functions in grade-Y products.

**Table 2-2. List of RL78/F14 Pins Other than Port Pins (5/5)**

| Pin Function | I/O | Function | Pin Count | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 100-pin | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin |
| $EV_{DD0}$ | — | Positive power supply for the pins that are not connected to $V_{DD}$ | √ | √ | √ | — | — | — |
| $EV_{DD1}$ | — | | √ | — | — | — | — | — |
| $AV_{REFP}$ | Input | A/D converter reference voltage (+ side) input | √ | √ | √ | √ | √ | √ |
| $AV_{REFM}$ | Input | A/D converter reference voltage (- side) input | √ | √ | √ | √ | √ | √ |
| $V_{SS}$ | — | Ground potential for the P33, P34, P80 to P87, P90 to P97, P100 to P105, P121 to P124, P137, and $\overline{\text{RESET}}$ pins | √ | √ | √ | √ | √ | √ |
| $EV_{SS0}$ | — | Ground potential for the pins that are not connected to $V_{SS}$ | √ | √ | √ | — | — | — |
| $EV_{SS1}$ | — | | √ | — | — | — | — | — |
| TOOLRXD | Input | UART reception pin for the external device connection used during flash memory programming | √ | √ | √ | √ | √ | √ |
| TOOLTXD | Output | UART transmission pin for the external device connection used during flash memory programming | √ | √ | √ | √ | √ | √ |
| TOOL0 | I/O | Data input/output for flash memory programmer/debugger | √ | √ | √ | √ | √ | √ |

**Table 2-3. List of RL78/F13 (CAN and LIN incorporated) Pins Other than Port Pins (1/4)**

| Pin Function | I/O | Function | Pin Count | | | | |
|---|---|---|---|---|---|---|---|
| | | | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin |
| ANI0 | Input | A/D converter analog input ($V_{DD}$ connection) | √ | √ | √ | √ | √ |
| ANI1 | Input | | √ | √ | √ | √ | √ |
| ANI2 | Input | | √ | √ | √ | √ | √ |
| ANI3 | Input | | √ | √ | √ | √ | √ |
| ANI4 | Input | | √ | √ | √ | √ | √ |
| ANI5 | Input | | √ | √ | √ | √ | √ |
| ANI6 | Input | | √ | √ | √ | √ | √ |
| ANI7 | Input | | √ | √ | √ | √ | √ |
| ANI8 | Input | | √ | √ | √ | — | √ |
| ANI9 | Input | | √ | √ | √ | — | √ |
| ANI10 | Input | | √ | √ | √ | — | — |
| ANI11 | Input | | √ | √ | √ | — | — |
| ANI12 | Input | | √ | √ | √ | — | — |
| ANI13 | Input | | √ | √ | — | — | — |
| ANI14 | Input | | √ | √ | — | — | — |
| ANI15 | Input | | √ | √ | — | — | — |
| ANI24 | Input | A/D converter analog input ($EV_{DD}$ connection) | √ | √ | √ | √ | √ |
| ANI25 | Input | | √ | √ | √ | √ | √ |
| ANI26 | Input | | √ | √ | — | — | — |
| ANI27 | Input | | √ | — | — | — | — |
| KR0 | Input | Key interrupt input | √ | √ | √ | √ | √ |
| KR1 | Input | | √ | √ | √ | √ | √ |
| KR2 | Input | | √ | √ | √ | √ | √ |
| KR3 | Input | | √ | √ | √ | √ | √ |
| KR4 | Input | | √ | √ | √ | √ | √ |
| KR5 | Input | | √ | √ | √ | √ | √ |
| KR6 | Input | | √ | √ | √ | — | √ |
| KR7 | Input | | √ | √ | √ | — | √ |
| TI00 | Input | 16-bit timer 00 input | √ | √ | √ | √ | √ |
| TI01 | Input | 16-bit timer 01 input (8-bit mode available) | √ | √ | √ | √ | √ |
| TI02 | Input | 16-bit timer 02 input | √ | √ | √ | √ | √ |
| TI03 | Input | 16-bit timer 03 input (8-bit mode available) | √ | √ | √ | √ | √ |
| TI04 | Input | 16-bit timer 04 input | √ | √ | √ | √ | √ |
| TI05 | Input | 16-bit timer 05 input | √ | √ | √ | √ | √ |
| TI06 | Input | 16-bit timer 06 input | √ | √ | √ | √ | √ |
| TI07 | Input | 16-bit timer 07 input | √ | √ | √ | √ | √ |
| TI10 | Input | 16-bit timer 10 input | √ | √ | √ | √ | √ |
| TI11 | Input | 16-bit timer 11 input (8-bit mode available) | √ | √ | √ | √ | √ |
| TI12 | Input | 16-bit timer 12 input | √ | √ | √ | √ | √ |
| TI13 | Input | 16-bit timer 13 input (8-bit mode available) | √ | √ | √ | √ | √ |
| TO00 | Output | 16-bit timer 00 output | √ | √ | √ | √ | √ |
| TO01 | Output | 16-bit timer 01 output (8-bit mode available) | √ | √ | √ | √ | √ |

**Table 2-3. List of RL78/F13 (CAN and LIN incorporated) Pins Other than Port Pins (2/4)**

| Pin Function | I/O | Function | Pin Count | | | | |
|---|---|---|---|---|---|---|---|
| | | | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin |
| TO02 | Output | 16-bit timer 02 output | √ | √ | √ | √ | √ |
| TO03 | Output | 16-bit timer 03 output (8-bit mode available) | √ | √ | √ | √ | √ |
| TO04 | Output | 16-bit timer 04 output | √ | √ | √ | √ | √ |
| TO05 | Output | 16-bit timer 05 output | √ | √ | √ | √ | √ |
| TO06 | Output | 16-bit timer 06 output | √ | √ | √ | √ | √ |
| TO07 | Output | 16-bit timer 07 output | √ | √ | √ | √ | √ |
| TO10 | Output | 16-bit timer 10 output | √ | √ | √ | √ | √ |
| TO11 | Output | 16-bit timer 11 output (8-bit mode available) | √ | √ | √ | √ | √ |
| TO12 | Output | 16-bit timer 12 output | √ | √ | √ | √ | √ |
| TO13 | Output | 16-bit timer 13 output (8-bit mode available) | √ | √ | √ | √ | √ |
| TRJIO0 | I/O | Timer RJ input/output | √ | √ | √ | √ | √ |
| TRJO0 | Output | Timer RJ output | √ | √ | √ | √ | √ |
| TRDCLK0 | Input | Timer RD external clock input | √ | √ | √ | √ | √ |
| TRDIOA0 | I/O | Timer RD0 input/output | √ | √ | √ | √ | √ |
| TRDIOB0 | I/O | | √ | √ | √ | √ | √ |
| TRDIOC0 | I/O | | √ | √ | √ | √ | √ |
| TRDIOD0 | I/O | | √ | √ | √ | √ | √ |
| TRDIOA1 | I/O | Timer RD1 input/output | √ | √ | √ | √ | √ |
| TRDIOB1 | I/O | | √ | √ | √ | √ | √ |
| TRDIOC1 | I/O | | √ | √ | √ | √ | √ |
| TRDIOD1 | I/O | | √ | √ | √ | √ | √ |
| RXD0 | Input | Serial data input to UART0 | √ | √ | √ | √ | √ |
| RXD1 | Input | Serial data input to UART1 | √ | √ | √ | √ | √ |
| TXD0 | Output | Serial data output from UART0 | √ | √ | √ | √ | √ |
| TXD1 | Output | Serial data output from UART1 | √ | √ | √ | √ | √ |
| SCLA0 | I/O | Clock input/output for IICA0 | √ | √ | √ | √ | — |
| SCL00 | Output | Clock output from simplified $I^2C$ | √ | √ | √ | √ | √ |
| SCL01 | Output | | √ | √ | √ | √ | √ |
| SCL10 | Output | | √ | √ | √ | √ | √ |
| SCL11 | Output | | √ | √ | √ | — | — |
| SDAA0 | I/O | Serial data input/output for IICA0 | √ | √ | √ | √ | — |
| SDA00 | I/O | Serial data input/output for simplified $I^2C$ | √ | √ | √ | √ | √ |
| SDA01 | I/O | | √ | √ | √ | √ | √ |
| SDA10 | I/O | | √ | √ | √ | √ | √ |
| SDA11 | I/O | | √ | √ | √ | — | — |
| $\overline{SCK00}$ | I/O | Clock input/output for CSI00 | √ | √ | √ | √ | √ |
| $\overline{SCK01}$ | I/O | Clock input/output for CSI01 | √ | √ | √ | √ | √ |
| $\overline{SCK10}$ | I/O | Clock input/output for CSI10 | √ | √ | √ | √ | √ |
| $\overline{SCK11}$ | I/O | Clock input/output for CSI11 | √ | √ | √ | — | — |
| SI00 | Input | Serial data input to CSI00 | √ | √ | √ | √ | √ |
| SI01 | Input | Serial data input to CSI01 | √ | √ | √ | √ | √ |
| SI10 | Input | Serial data input to CSI10 | √ | √ | √ | √ | √ |

**Table 2-3. List of RL78/F13 (CAN and LIN incorporated) Pins Other than Port Pins (3/4)**

| Pin Function | I/O | Function | Pin Count | | | | |
|---|---|---|---|---|---|---|---|
| | | | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin |
| SI11 | Input | Serial data input to CSI11 | √ | √ | √ | — | — |
| SO00 | Output | Serial data output from CSI00 | √ | √ | √ | √ | √ |
| SO01 | Output | Serial data output from CSI01 | √ | √ | √ | √ | √ |
| SO10 | Output | Serial data output from CSI10 | √ | √ | √ | √ | √ |
| SO11 | Output | Serial data output from CSI11 | √ | √ | √ | — | — |
| $\overline{SSI00}$ | Input | Slave select input to CSI00 (SPI00) | √ | √ | √ | √ | √ |
| $\overline{SSI01}$ | Input | Slave select input to CSI01 (SPI01) | √ | √ | √ | √ | √ |
| $\overline{SSI10}$ | Input | Slave select input to CSI10 (SPI10) | √ | √ | — | — | — |
| $\overline{SSI11}$ | Input | Slave select input to CSI11 (SPI11) | √ | √ | √ | — | — |
| CRXD0 | Input | Serial data input to CAN | √ | √ | √ | √ | √ |
| CTXD0 | Output | Serial data output from CAN | √ | √ | √ | √ | √ |
| LRXD0 | Input | Serial data input to LIN | √ | √ | √ | √ | √ |
| LTXD0 | Output | Serial data output from LIN | √ | √ | √ | √ | √ |
| INTP0 | Input | External interrupt input | √ | √ | √ | √ | √ |
| INTP1 | Input | | √ | √ | √ | √ | √ |
| INTP2 | Input | | √ | √ | √ | √ | √ |
| INTP3 | Input | | √ | √ | √ | √ | √ |
| INTP4 | Input | | √ | √ | √ | √ | √ |
| INTP5 | Input | | √ | √ | √ | √ | √ |
| INTP6 | Input | | √ | √ | √ | — | — |
| INTP7 | Input | | √ | √ | √ | — | — |
| INTP8 | Input | | √ | √ | √ | — | — |
| INTP9 | Input | | √ | √ | √ | — | — |
| INTP10 | Input | | √ | √ | — | — | — |
| INTP11 | Input | | √ | √ | — | — | — |
| PCLBUZ0 | Output | Clock output/buzzer output 0 | √ | √ | √ | — | — |
| RESOUT | Output | Reset output | √ | √ | √ | — | — |
| STOPST | Output | STOP status output | √ | √ | √ | — | — |
| SNZOUT0 | Output | SNOOZE status output | √ | √ | √ | √ | √ |
| SNZOUT1 | Output | | √ | √ | √ | √ | √ |
| SNZOUT2 | Output | | √ | √ | √ | √ | √ |
| SNZOUT3 | Output | | √ | √ | √ | √ | √ |
| SNZOUT4 | Output | | √ | √ | √ | — | — |
| SNZOUT5 | Output | | √ | √ | √ | — | — |
| SNZOUT6 | Output | | √ | √ | √ | — | — |
| SNZOUT7 | Output | | √ | √ | √ | — | — |
| RTC1HZ | Output | Real-time clock correction clock (1 Hz) output | √ | √ | √ | √ | √ |
| EXCLK | Input | External clock input for main system clock | √ | √ | √ | √ | √ |
| EXCLKS | Input | External clock input for subsystem clock | √ | √ | √ | — | — |
| X1 | — | Resonator connection for main system clock | √ | √ | √ | √ | √ |
| X2 | — | | √ | √ | √ | √ | √ |

**Table 2-3. List of RL78/F13 (CAN and LIN incorporated) Pins Other than Port Pins (4/4)**

| Pin Function | I/O | Function | Pin Count | | | | |
|---|---|---|---|---|---|---|---|
| | | | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin |
| XT1Note | — | Resonator connection for subsystem clock | √ | √ | √ | — | — |
| XT2Note | — | | √ | √ | √ | — | — |
| $\overline{\text{RESET}}$ | Input | External reset input | √ | √ | √ | √ | √ |
| REGC | — | Regulator output stabilization capacitance connection for internal operation. Connect to Vss via the capacitor (0.47 to 1 μF). | √ | √ | √ | √ | √ |
| $V_{DD}$ | — | Positive power supply for the P33, P34, P80 to P87, P90 to P95, P121 to P124, P137, and $\overline{\text{RESET}}$ pins | √ | √ | √ | √ | √ |
| $EV_{DD0}$ | — | Positive power supply for the pins that are not connected to $V_{DD}$ | √ | √ | — | — | — |
| $AV_{REFP}$ | Input | A/D converter reference voltage (+ side) input | √ | √ | √ | √ | √ |
| $AV_{REFM}$ | Input | A/D converter reference voltage (- side) input | √ | √ | √ | √ | √ |
| $V_{SS}$ | — | Ground potential for the P33, P34, P80 to P87, P90 to P95, P121 to P124, P137, and $\overline{\text{RESET}}$ pins | √ | √ | √ | √ | √ |
| $EV_{SS0}$ | — | Ground potential for the pins that are not connected to $V_{SS}$ | √ | √ | — | — | — |
| TOOLRXD | Input | UART reception pin for the external device connection used during flash memory programming | √ | √ | √ | √ | √ |
| TOOLTXD | Output | UART transmission pin for the external device connection used during flash memory programming | √ | √ | √ | √ | √ |
| TOOL0 | I/O | Data input/output for flash memory programmer/debugger | √ | √ | √ | √ | √ |

**Note** Do not use the XT1 and XT2 pin functions in grade-Y products.

**Table 2-4. List of RL78/F13 (LIN incorporated) Pins Other than Port Pins (1/4)**

| Pin Function | I/O | Function | Pin Count | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
| ANI0 | Input | A/D converter analog input ($V_{DD}$ connection) | √ | √ | √ | √ | √ | √ |
| ANI1 | Input | | √ | √ | √ | √ | √ | √ |
| ANI2 | Input | | √ | √ | √ | √ | √ | √ |
| ANI3 | Input | | √ | √ | √ | √ | √ | √ |
| ANI4 | Input | | √ | √ | √ | √ | √ | — |
| ANI5 | Input | | √ | √ | √ | √ | √ | — |
| ANI6 | Input | | √ | √ | √ | √ | √ | — |
| ANI7 | Input | | √ | √ | √ | √ | √ | — |
| ANI8 | Input | | √ | √ | √ | — | √ | — |
| ANI9 | Input | | √ | √ | √ | — | √ | — |
| ANI10 | Input | | √ | √ | √ | — | — | — |
| ANI11 | Input | | √ | √ | √ | — | — | — |
| ANI12 | Input | | √ | √ Note 1 | √ Note 1 | — | — | — |
| ANI13 | Input | | √ | √ Note 1 | — | — | — | — |
| ANI14 | Input | | √ | √ Note 1 | — | — | — | — |
| ANI15 | Input | | √ | √ Note 1 | — | — | — | — |
| ANI24 | Input | A/D converter analog input ($EV_{DD}$ connection) | √ | √ Note 1 | √ Note 1 | — | — | — |
| ANI25 | Input | | √ | √ Note 1 | √ Note 1 | — | — | — |
| ANI26 | Input | | √ | √ Note 1 | — | — | — | — |
| ANI27 | Input | | √ | — | — | — | — | — |
| KR0 | Input | Key interrupt input | √ | √ | √ | √ | √ | √ |
| KR1 | Input | | √ | √ | √ | √ | √ | √ |
| KR2 | Input | | √ | √ | √ | √ | √ | — |
| KR3 | Input | | √ | √ | √ | √ | √ | — |
| KR4 | Input | | √ | √ | √ | √ | √ | — |
| KR5 | Input | | √ | √ | √ | √ | √ | — |
| KR6 | Input | | √ | √ | √ | — | √ | — |
| KR7 | Input | | √ | √ | √ | — | √ | — |
| TI00 | Input | 16-bit timer 00 input | √ | √ | √ | √ | √ | √ |
| TI01 | Input | 16-bit timer 01 input (8-bit mode available) | √ | √ | √ | √ | √ | √ |
| TI02 | Input | 16-bit timer 02 input | √ | √ | √ | √ | √ | √ |
| TI03 | Input | 16-bit timer 03 input (8-bit mode available) | √ | √ | √ | √ | √ | √ |
| TI04 | Input | 16-bit timer 04 input | √ | √ | √ | √ | √ | √ |
| TI05 | Input | 16-bit timer 05 input | √ | √ | √ | √ | √ | √ |
| TI06 | Input | 16-bit timer 06 input | √ | √ | √ | √ | √ | √ |
| TI07 | Input | 16-bit timer 07 input | √ | √ | √ | √ | √ | √ |
| TI10 | Input | 16-bit timer 10 input | √ | √ Note 1 | √ Note 1 | — | — | — |
| TI11 | Input | 16-bit timer 11 input | √ | √ Note 1 | √ Note 1 | — | — | — |
| TI12 | Input | 16-bit timer 12 input | √ | √ Note 1 | √ Note 1 | — | — | — |
| TI13 | Input | 16-bit timer 13 input | √ | √ Note 1 | √ Note 1 | — | — | — |

**Note 1.** Provided only in the products with 96 Kbytes or 128 Kbytes of ROM.

**Table 2-4. List of RL78/F13 (LIN incorporated) Pins Other than Port Pins (2/4)**

| Pin Function | I/O | Function | Pin Count | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
| TO00 | Output | 16-bit timer 00 output | √ | √ | √ | √ | √ | √ |
| TO01 | Output | 16-bit timer 01 output (8-bit mode available) | √ | √ | √ | √ | √ | √ |
| TO02 | Output | 16-bit timer 02 output | √ | √ | √ | √ | √ | √ |
| TO03 | Output | 16-bit timer 03 output (8-bit mode available) | √ | √ | √ | √ | √ | √ |
| TO04 | Output | 16-bit timer 04 output | √ | √ | √ | √ | √ | √ |
| TO05 | Output | 16-bit timer 05 output | √ | √ | √ | √ | √ | √ |
| TO06 | Output | 16-bit timer 06 output | √ | √ | √ | √ | √ | √ |
| TO07 | Output | 16-bit timer 07 output | √ | √ | √ | √ | √ | √ |
| TO10 | Output | 16-bit timer 10 output | √ | √ Note 1 | √ Note 1 | — | — | — |
| TO11 | Output | 16-bit timer 11 output (8-bit mode available) | √ | √ Note 1 | √ Note 1 | — | — | — |
| TO12 | Output | 16-bit timer 12 output | √ | √ Note 1 | √ Note 1 | — | — | — |
| TO13 | Output | 16-bit timer 13 output (8-bit mode available) | √ | √ Note 1 | √ Note 1 | — | — | — |
| TRJIO0 | I/O | Timer RJ input/output | √ | √ | √ | √ | √ | — |
| TRJO0 | Output | Timer RJ output | √ | √ | √ | √ | √ | — |
| TRDCLK0 | Input | Timer RD external clock input | √ | √ | √ | √ | √ | √ |
| TRDIOA0 | I/O | Timer RD0 input/output | √ | √ | √ | √ | √ | √ |
| TRDIOB0 | I/O | | √ | √ | √ | √ | √ | √ |
| TRDIOC0 | I/O | | √ | √ | √ | √ | √ | √ |
| TRDIOD0 | I/O | | √ | √ | √ | √ | √ | √ |
| TRDIOA1 | I/O | Timer RD1 input/output | √ | √ | √ | √ | √ | √ |
| TRDIOB1 | I/O | | √ | √ | √ | √ | √ | √ |
| TRDIOC1 | I/O | | √ | √ | √ | √ | √ | √ |
| TRDIOD1 | I/O | | √ | √ | √ | √ | √ | √ |
| RXD0 | Input | Serial data input to UART0 | √ | √ | √ | √ | √ | √ |
| RXD1 | Input | Serial data input to UART1 | √ | √ Note 1 | √ Note 1 | — | — | — |
| TXD0 | Output | Serial data output from UART0 | √ | √ | √ | √ | √ | √ |
| TXD1 | Output | Serial data output from UART1 | √ | √ Note 1 | √ Note 1 | — | — | — |
| SCLA0 | I/O | Clock input/output for IICA0 | √ | √ Note 1 | √ Note 1 | — | — | — |
| SCL00 | Output | Clock output from simplified $I^2C$ | √ | √ | √ | √ | √ | √ |
| SCL01 | Output | | √ | √ | √ | √ | √ | √ |
| SCL10 | Output | | √ | √ Note 1 | √ Note 1 | — | — | — |
| SCL11 | Output | | √ | √ Note 1 | √ Note 1 | — | — | — |
| SDAA0 | I/O | Serial data input/output for IICA0 | √ | √ Note 1 | √ Note 1 | — | — | — |
| SDA00 | I/O | Serial data input/output for simplified $I^2C$ | √ | √ | √ | √ | √ | √ |
| SDA01 | I/O | | √ | √ | √ | √ | √ | √ |
| SDA10 | I/O | | √ | √ Note 1 | √ Note 1 | — | — | — |
| SDA11 | I/O | | √ | √ Note 1 | √ Note 1 | — | — | — |
| $\overline{SCK00}$ | I/O | Clock input/output for CSI00 | √ | √ | √ | √ | √ | √ |
| $\overline{SCK01}$ | I/O | Clock input/output for CSI01 | √ | √ | √ | √ | √ | √ |

**Note 1.** Provided only in the products with 96 Kbytes or 128 Kbytes of ROM.

**Table 2-4. List of RL78/F13 (LIN incorporated) Pins Other than Port Pins (3/4)**

| Pin Function | I/O | Function | Pin Count | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
| $\overline{SCK10}$ | I/O | Clock input/output for CSI10 | √ | √ Note 1 | √ Note 1 | — | — | — |
| $\overline{SCK11}$ | I/O | Clock input/output for CSI11 | √ | √ Note 1 | √ Note 1 | — | — | — |
| SI00 | Input | Serial data input to CSI00 | √ | √ | √ | √ | √ | √ |
| SI01 | Input | Serial data input to CSI01 | √ | √ | √ | √ | √ | √ |
| SI10 | Input | Serial data input to CSI10 | √ | √ Note 1 | √ Note 1 | — | — | — |
| SI11 | Input | Serial data input to CSI11 | √ | √ Note 1 | √ Note 1 | — | — | — |
| SO00 | Output | Serial data output from CSI00 | √ | √ | √ | √ | √ | √ |
| SO01 | Output | Serial data output from CSI01 | √ | √ | √ | √ | √ | √ |
| SO10 | Output | Serial data output from CSI10 | √ | √ Note 1 | √ Note 1 | — | — | — |
| SO11 | Output | Serial data output from CSI11 | √ | √ Note 1 | √ Note 1 | — | — | — |
| $\overline{SSI00}$ | Input | Slave select input to CSI00 (SPI00) | √ | √ | √ | √ | √ | √ |
| $\overline{SSI01}$ | Input | Slave select input to CSI01 (SPI01) | √ | √ | √ | √ | √ | √ |
| $\overline{SSI10}$ | Input | Slave select input to CSI10 (SPI10) | √ | √ Note 1 | √ Note 1 | — | — | — |
| $\overline{SSI11}$ | Input | Slave select input to CSI11 (SPI11) | √ | √ Note 1 | √ Note 1 | — | — | — |
| LRXD0 | Input | Serial data input to LIN | √ | √ | √ | √ | √ | √ |
| LTXD0 | Output | Serial data output from LIN | √ | √ | √ | √ | √ | √ |
| INTP0 | Input | External interrupt input | √ | √ | √ | √ | √ | √ |
| INTP1 | Input | | √ | √ | √ | √ | √ | √ |
| INTP2 | Input | | √ | √ | √ | √ | √ | √ |
| INTP3 | Input | | √ | √ | √ | √ | √ | √ |
| INTP4 | Input | | √ | √ | √ | √ | √ | √ |
| INTP5 | Input | | √ | √ | √ | √ | √ | — |
| INTP6 | Input | | √ | √ | √ | — | — | — |
| INTP7 | Input | | √ | √ | √ | — | — | — |
| INTP8 | Input | | √ | √ Note 1 | √ Note 1 | — | — | — |
| INTP9 | Input | | √ | √ Note 1 | √ Note 1 | — | — | — |
| INTP10 | Input | | √ | √ Note 1 | — | — | — | — |
| INTP11 | Input | | √ | √ Note 1 | — | — | — | — |
| PCLBUZ0 | Output | Clock output/buzzer output 0 | √ | √ | √ | — | — | — |
| RESOUT | Output | Reset output | √ | √ | √ | — | — | — |
| STOPST | Output | STOP status output | √ | √ | √ | — | — | — |
| SNZOUT0 | Output | SNOOZE status output | √ | √ | √ | √ | √ | √ |
| SNZOUT1 | Output | | √ | √ | √ | √ | √ | √ |
| SNZOUT2 | Output | | √ | √ | √ | √ | √ | — |
| SNZOUT3 | Output | | √ | √ | √ | √ | √ | — |
| SNZOUT4 | Output | | √ | √ | √ | — | — | — |
| SNZOUT5 | Output | | √ | √ | √ | — | — | — |
| SNZOUT6 | Output | | √ | √ | √ | — | — | — |
| SNZOUT7 | Output | | √ | √ | √ | — | — | — |
| RTC1HZ | Output | Real-time clock correction clock (1 Hz) output | √ | √ | √ | √ | √ | √ |

**Note 1.** Provided only in the products with 96 Kbytes or 128 Kbytes of ROM.

**Table 2-4. List of RL78/F13 (LIN incorporated) Pins Other than Port Pins (4/4)**

| Pin Function | I/O | Function | Pin Count | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
| EXCLK | Input | External clock input for main system clock | √ | √ | √ | √ | √ | √ |
| EXCLKS | Input | External clock input for subsystem clock | √ | √ | √ | — | — | — |
| X1 | — | Resonator connection for main system clock | √ | √ | √ | √ | √ | √ |
| X2 | — | | √ | √ | √ | √ | √ | √ |
| XT1Note | — | Resonator connection for subsystem clock | √ | √ | √ | — | — | — |
| XT2Note | — | | √ | √ | √ | — | — | — |
| $\overline{\text{RESET}}$ | Input | External reset input | √ | √ | √ | √ | √ | √ |
| REGC | — | Regulator output stabilization capacitance connection for internal operation. Connect to Vss via the capacitor (0.47 to 1 μF). | √ | √ | √ | √ | √ | √ |
| $V_{DD}$ | — | Positive power supply for the P33, P34, P80 to P87, P90 to P97, P100 to P105, P121 to P124, P137, and $\overline{\text{RESET}}$ pins | √ | √ | √ | √ | √ | √ |
| $EV_{DD0}$ | — | Positive power supply for the pins that are not connected to $V_{DD}$ | √ | √ | — | — | — | — |
| $AV_{REFP}$ | Input | A/D converter reference voltage (+ side) input | √ | √ | √ | √ | √ | √ |
| $AV_{REFM}$ | Input | A/D converter reference voltage (- side) input | √ | √ | √ | √ | √ | √ |
| $V_{SS}$ | — | Ground potential for the P33, P34, P80 to P87, P90 to P97, P100 to P105, P121 to P124, P137, and $\overline{\text{RESET}}$ pins | √ | √ | √ | √ | √ | √ |
| $EV_{SS0}$ | — | Ground potential for the pins that are not connected to $V_{SS}$ | √ | √ | — | — | — | — |
| TOOLRXD | Input | UART reception pin for the external device connection used during flash memory programming | √ | √ | √ | √ | √ | √ |
| TOOLTXD | Output | UART transmission pin for the external device connection used during flash memory programming | √ | √ | √ | √ | √ | √ |
| TOOL0 | I/O | Data input/output for flash memory programmer/debugger | √ | √ | √ | √ | √ | √ |

**Note** Do not use the XT1 and XT2 pin functions in grade-Y products.

## 2.2 Description of Pin Functions

The pins provided depend on the product. See **1.5 Pin Configurations**, for details. This subchapter describes the pin functions of the 100-pin products of RL78/F14 and the 80-pin products of RL78/F13 (CAN and LIN incorporated) as examples.

### 2.2.1 P00 to P03 (Port 0)

P00 to P03 function as an I/O port. These pins also function as external interrupt request input, timer I/O, and real-time clock correction clock output. P03 is provided only in the 100-pin products of RL78/F14.

Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P00 to P03 function as an I/O port. These pins can be set to input or output port in 1-bit units using port mode register 0 (PM0).

**(2) Control mode**

P00 to P03 function as external interrupt request input, real-time clock correction clock output, and timer I/O.

**(a) INTP9**

This is an external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**(b) RTC1HZ**

This is a real-time clock correction clock (1 Hz) output pin.

**(c) TI04 to TI06**

These are pins for inputting an external count clock/capture trigger to 16-bit timers.

**(d) TO04 to TO06**

These are timer output pins of 16-bit timers.

### 2.2.2 P10 to P17 (Port 1)

P10 to P17 function as an I/O port. These pins also function as external interrupt request input, real-time clock correction clock output, serial interface data I/O, clock I/O, timer I/O, programming UART I/O, SNOOZE status output, LIN serial data I/O, and CAN serial data I/O.

Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

For input to the P10, P11, P13, P14, P16, and P17 pins, a CMOS input buffer or a TTL input buffer can be selected using the port input mode register 1 (PIM1).

For output from the P10 to P17 pins, CMOS output or N-ch open-drain output can be selected using the port output mode register 1 (POM1).

For the P10, P11, P13, P14, P16, and P17 pins, the input threshold level can be specified using the port input threshold control register 1 (PITHL1).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P10 to P17 function as an I/O port. These pins can be set to input or output port in 1-bit units using port mode register 1 (PM1).

**(2) Control mode**

P10 to P17 function as external interrupt request input, real-time clock correction clock output, serial interface data I/O, clock I/O, timer I/O, programming UART I/O, SNOOZE status output, LIN serial data I/O, and CAN serial data I/O.

**(a) INTP3, INTP5**

These are external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**(b) RTC1HZ**

This is a real-time clock correction clock (1 Hz) output pin.

**(c) TXD0, TXD1**

These are serial data output pins of the UART0 and UART1 serial interface.

**(d) RXD0, RXD1**

These are serial data input pins of the UART0 and UART1 serial interface.

**(e) $\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$**

These are serial clock I/O pins of the CSI00, CSI01, and CSI10 serial interface.

**(f) SI00, SI01, SI10**

These are serial data input pins of the CSI00, CSI01, and CSI10 serial interface.

**(g) SO00, SO10**

These are serial data output pins of the CSI00 and CSI10 serial interface.

**(h) TI00, TI02, TI04 to TI06, TI11 to TI13**

These are pins for inputting an external count clock/capture trigger to 16-bit timers.

**(i) TO00, TO02, TO04 to TO06, TO11 to TO13**

These are timer output pins of 16-bit timers.

**(j) SDA00, SDA01, SDA10**

These are serial data I/O pins of the simplified $I^2C$ serial interface.

**(k) SCL00, SCL01, SCL10**

These are serial clock I/O pins of the simplified $I^2C$ serial interface.

**(l) TRDIOA0, TRDIOB0, TRDIOC0, TRDIOD0, TRDIOA1, TRDIOB1, TRDIOC1**

These are timer I/O pins of timer RD.

**(m) TRDCLK0**

This is an external clock input pin of timer RD.

**(n) TRJO0**

This is a timer output pin of timer RJ.

**(o) LTXD0, LTXD1**

These are serial data output pins for the LIN. LTXD1 is provided only in the 100-pin products of RL78/F14.

**(p) LRXD0, LRXD1**

These are serial data input pins for the LIN. LRXD1 is provided only in the 100-pin products of RL78/F14.

**(q) CTXD0**

This is a serial data output pins for the CAN.

**(r) CRXD0**

This is a serial data input pins for the CAN.

**(s) TOOLTXD**

This is a UART serial data output pin for the external device connection used during flash memory programming.

**(t) TOOLRXD**

This is a UART serial data input pin for the external device connection used during flash memory programming.

**(u) SNZOUT3**

This is a SNOOZE status output pin.

### 2.2.3 P30 to P34 (Port 3)

P30 to P34 function as an I/O port. These pins also function as A/D converter analog input, A/D converter reference voltage input, external interrupt request input, serial interface slave select input, timer I/O, SNOOZE status output, and STOP status output.

Only for P30 to P32, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

For input to the P30 pin, a CMOS input buffer or a TTL input buffer can be selected using the port input mode register 3 (PIM3).

For the P30 pin, the input threshold level can be specified using the port input threshold control register 3 (PITHL3).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P30 to P34 function as an I/O port. These pins can be set to input or output port in 1-bit units using port mode register 3 (PM3).

**(2) Control mode**

P30 to P34 function as A/D converter analog input, A/D converter reference voltage input, external interrupt request input, serial interface slave select input, timer I/O, SNOOZE status output, and STOP status output.

**(a) ANI0, ANI1**

These are analog input pins of the A/D converter. For details, see **12.10 (5) Analog input (ANIn) pins**.

**(b) AVREFP**

This is a reference voltage (+ side) input pin of the A/D converter.

**(c) AVREFM**

This is a reference voltage (- side) input pin of the A/D converter.

**(d) INTP2, INTP7**

These are external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**(e) $\overline{\text{SSI00}}$**

This is a slave select input pin of the CSI00 (SPI00) serial interface.

**(f) TI01, TI14, TI16**

These are pins for inputting an external count clock/capture trigger to 16-bit timers. TI14 and TI16 are provided only in the 100-pin products of RL78/F14.

**(g) TO01, TO14, TO16**

These are timer output pins of 16-bit timers. TO14 and TO16 are provided only in the 100-pin products of RL78/F14.

**(h) SNZOUT0**

This is a SNOOZE status output pin.

**(i) TRDIOD1**

This is a timer output pin of timer RD.

**(j) STOPST**

This is a STOP status output pin.

### 2.2.4 P40 to P47 (Port 4)

P40 to P47 function as an I/O port. These pins also function as data I/O for a flash memory programmer/debugger, timer I/O, comparator output, external interrupt request input, SNOOZE status output, and LIN serial data I/O.

Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 4 (PU4).

For the P43 pin, the input threshold level can be specified using the port input threshold control register 4 (PITHL4).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P40 to P47 function as an I/O port. These pins can be set to input or output port in 1-bit units using port mode register 4 (PM4).

**(2) Control mode**

P40 to P47 function as data I/O for a flash memory programmer/debugger, timer I/O, comparator output, external interrupt request input, SNOOZE status output, and LIN serial data I/O.

**(a) TOOL0**

This is a data I/O pin for a flash memory programmer/debugger.

Be sure to pull up this pin externally when on-chip debugging is enabled (pulling it down is prohibited).

**(b) TI07, TI10, TI12**

These are pins for inputting an external count clock/capture trigger to 16-bit timers.

**(c) TO07, TO10, TO12**

These are timer output pins of 16-bit timers.

**(d) TRJIO0**

This is a timer I/O pin of timer RJ.

**(e) VCOUT0**

This is a comparator output pin. This pin is provided only in the 100-pin products of RL78/F14.

**(f) INTP13**

This is an external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified. This pin is provided only in the 100-pin products of RL78/F14.

**(g) SNZOUT2**

This is a SNOOZE status output pin.

**(h) LTXD0**

This is a serial data output pin for the LIN.

**(i) LRXD0**

This is a serial data input pin for the LIN.

**Caution After reset release, the relationships between P40/TOOL0 and the operating mode are as follows. For details, see 30. 5 Programming Method.**

**Table 2-5. Relationships Between P40/TOOL0 and Operation Mode After Reset Release**

| P40/TOOL0 | Operating Mode |
|---|---|
| $EV_{DD}$ | Normal operation mode |
| 0 V | Flash memory programming mode |

### 2.2.5 P50 to P57 (Port 5)

P50 to P57 function as an I/O port. These pins also function as external interrupt request input, serial interface slave select input, serial interface data I/O, clock I/O, timer I/O, SNOOZE status output, and STOP status output.

Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 5 (PU5).

For input to the P54 pin, a CMOS input buffer or a TTL input buffer can be selected using the port input mode register 5 (PIM5).

For the P50 and P52 to P54 pins, the input threshold level can be specified using the port input threshold control register 5 (PITHL5).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P50 to P57 function as an I/O port. These pins can be set to input or output port in 1-bit units using port mode register 5 (PM5).

**(2) Control mode**

P50 to P57 function as external interrupt request input, serial interface slave select input, serial interface data I/O, clock I/O, timer I/O, SNOOZE status output, and STOP status output.

**(a) INTP3, INTP10, INTP11**

These are external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**(b) SSI01**

This is a slave select pin of the CSI01 (SPI01) serial interface.

**(c) $\overline{\text{SSI10}}$**

This is a slave select pin of the CSI10 (SPI10) serial interface.

**(d) SCK01**

This is a serial clock I/O pin of the CSI01 serial interface.

**(e) SI01**

This is a serial data input pin of the CSI01 serial interface.

**(f) SO01**

This is a serial data output pin of the CSI01 serial interface.

**(g) TI11, TI13, TI15, TI17**

These are pins for inputting an external count clock/capture trigger to 16-bit timers.

**(h) TO11, TO13, TO15, TO17**

These are timer output pins of 16-bit timers. TO15 and TO17 are provided only in 100-pin products of RL78/F14.

**(i) SNZOUT0, SNZOUT1**

These are SNOOZE status output pins.

**(j) STOPST**

This is a STOP status output pin.

### 2.2.6 P60 to P67 (Port 6)

P60 to P67 function as an I/O port. These pins also function as serial interface data I/O, clock I/O, slave select input, timer I/O, and SNOOZE status output.

Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 6 (PU6).

For input to the P62 and P63 pins, a CMOS input buffer or a TTL input buffer can be selected using the port input mode register 6 (PIM6).

For output from the P60 to P63 pins, CMOS output or N-ch open-drain output can be selected using the port output mode register 6 (POM6).

For the P60 to P63 pins, the input threshold level can be specified using the port input threshold control register 6 (PITHL6).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P60 to P67 function as an I/O port. These pins can be set to input or output port in 1-bit units using port mode register 6 (PM6).

**(2) Control mode**

P60 to P67 function as serial interface data I/O, clock I/O, slave select input, timer I/O, and SNOOZE status output.

**(a) SCLA0**

This is a serial clock I/O pin of the IICA0 serial interface.

**(b) SDAA0**

This is a serial data I/O pin of the IICA0 serial interface.

**(C) SSI00**

This is a slave select input pin of the CSI00 (SPI00) serial interface.

**(d) SCK00**

This is a serial clock I/O pin of the CSI00 serial interface.

**(e) SI00**

This is a serial data input pin of the CSI00 serial interface.

**(f) SO00**

This is a serial data output pin of the CSI00 serial interface.

**(g) TXD0**

This is a serial data output pin of the UART0 serial interface.

**(h) RXD0**

This is a serial data input pin of the UART0 serial interface.

**(i) SCL00**

This is a serial clock I/O pin of the simplified $I^2C$ serial interface.

**(j) SDA00**

This is a serial data I/O pin of the simplified $I^2C$ serial interface.

**(k) TI00, TI02, TI14, TI16**

These are pins for inputting an external count clock/capture trigger to 16-bit timers. TI14 and TI16 are provided only in 100-pin products of RL78/F14.

**(l) TO00, TO02, TO14, TO16**

These are timer output pins of 16-bit timers. TO14 and TO16 are provided only in 100-pin products of RL78/F14.

**(m) SNZOUT2, SNZOUT3**

These are SNOOZE status output pins.

### 2.2.7 P70 to P77 (Port 7)

P70 to P77 function as an I/O port. These pins also function as A/D converter analog input, external interrupt request input, key interrupt input, serial interface slave select input, data I/O, clock I/O, timer I/O, SNOOZE status output, and CAN serial data I/O.

Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

For input to the P70, P71, and P73 pins, a CMOS input buffer or a TTL input buffer can be selected using the port input mode register 7 (PIM7).

For output from the P70 to P72 pins, CMOS output or N-ch open-drain output can be selected using the port output mode register 7 (POM7).

For the P70, P71, P73, and P75 to P77 pins, the input threshold level can be specified using the port input threshold control register 7 (PITHL7).

Input to the P70 to P74 pins can be specified as digital input or analog input in 1-bit units, using the port mode control register 7 (PMC7).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P70 to P77 function as an I/O port. These pins can be set to input or output port in 1-bit units using port mode register 7 (PM7).

**(2) Control mode**

P70 to P77 function as A/D converter analog input, external interrupt request input, key interrupt input, serial interface slave select input, data I/O, clock I/O, timer I/O, SNOOZE status output, and CAN serial data I/O.

**(a) ANI26 to ANI30**

These are analog input pins of the A/D converter. These pins are provided only in the 100-pin products of RL78/F14. For details, see **12.10 (5) Analog input (ANIn) pins**.

**(b) INTP6, INTP8, INTP12**

These are external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified. INTP12 is provided only in the 100-pin products of RL78/F14.

**(c) KR0 to KR7**

These are key interrupt input pins.

**(d) SSI10, SSI11**

These are slave select input pins of the CSI10 (SPI10) and CSI11 (SPI11) serial interface.

**(e) SI10, SI11**

These are serial data input pins of the CSI10 and CSI11 serial interface.

**(f) SO10, SO11**

These are serial data output pins of the CSI10 and CSI11 serial interface.

**(g) TXD1**

This is a serial data output pin of the UART1 serial interface.

**(h) RXD1**

This is a serial data input pin of the UART1 serial interface.

**(i) SCK10, SCK11**

These are serial clock I/O pins of the CSI10 and CSI11 serial interface.

**(j) SCL11**

This is a serial clock I/O pin of the simplified $I^2C$ serial interface.

**(k) SDA11**

This is a serial data I/O pin of the simplified $I^2C$ serial interface.

**(l) TI15, TI17**

These are pins for inputting an external count clock/capture trigger to 16-bit timers. They are provided only in the 100-pin products of RL78/F14.

**(m) TO15, TO17**

These are timer output pins of 16-bit timers. They are provided only in the 100-pin products of RL78/F14.

**(n) CTXD0**

This is a serial data output pin for the CAN.

**(o) CRXD0**

This is a serial data input pin for the CAN.

**(p) SNZOUT4 to SNZOUT7**

These are SNOOZE status output pins.

### 2.2.8 P80 to P87 (Port 8)

P80 to P87 function as an I/O port. These pins also function as A/D converter analog input, D/A converter output, comparator reference voltage input, and comparator analog voltage input.

**(1) Port mode**

P80 to P87 function as an I/O port. These pins can be set to input or output port in 1-bit units using port mode register 8 (PM8).

**(2) Control mode**

P80 to P87 function as A/D converter analog input, D/A converter output, comparator reference voltage input, and comparator analog voltage input.

**(a) ANI2 to ANI9**

These are analog input pins of the A/D converter. For details, see **12.10 (5) Analog input (ANIn) pins**.

**(b) ANO0**

This is a D/A converter output pin. It is provided only in the 100-pin products of RL78/F14.

**(c) IVCMP00 to IVCMP03**

These are analog voltage input pins of the comparator. It is provided only in the 100-pin products of RL78/F14.

**(d) IVREF0**

This is a reference voltage input pin of the comparator. It is provided only in the 100-pin products of RL78/F14.

### 2.2.9 P90 to P97 (Port 9)

P90 to P97 function as an I/O port. These pins also function as A/D converter analog input.

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P90 to P97 function as an I/O port. These pins can be set to input or output port in 1-bit units using port mode register 9 (PM9).

**(2) Control mode**

P90 to P97 function as A/D converter analog input.

**(a) ANI10 to ANI17 (100-pin products of RL78/F14)**

These are analog input pins of the A/D converter. For details, see **12.10 (5) Analog input (ANIn) pins**.

**(b) ANI10 to ANI15, ANI26, ANI27 (80-pin products of RL78/F13 (CAN and LIN incorporated))**

These are analog input pins of the A/D converter. For details, see **12.10 (5) Analog input (ANIn) pins**.

### 2.2.10 P100 to P107 (Port 10)

P100 to P107 function as an I/O port. These pins are provided only in the 100-pin products of RL78/F14. These pins also function as A/D converter analog input and LIN serial data I/O.

For P106 and P107, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 10 (PU10).

For the P107 pin, the input threshold level can be specified using the port input threshold control register 10 (PITHL10).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P100 to P107 function as an I/O port. These pins can be set to input or output port in 1-bit units using port mode register 10 (PM10).

**(2) Control mode**

P100 to P107 function as A/D converter analog input and LIN serial data I/O.

**(a) ANI18 to ANI23**

These are analog input pins of the A/D converter. For details, see **12.10 (5) Analog input (ANIn) pins**.

**(b) LTXD1**

This is a serial data output pin for the LIN. LTXD1 is provided only in 100-pin products of RL78/F14.

**(c) LRXD1**

This is a serial data output pin for the LIN. LRXD1 is provided only in 100-pin products of RL78/F14.

### 2.2.11 P120 to P127 (Port 12)

The 100-pin products of RL78/F14 has P120 and P125 to P127 I/O port pins, and the 80-pin products of RL78/F13 (CAN and LIN incorporated) has P120, P125, and P126 I/O port pins. P121 to P124 are input port pins and provided in both products. These pins also function as A/D converter analog input, external interrupt request input, resonator connection for the main system clock, resonator connection for the subsystem clock, external clock input for the main system clock, external clock input for the subsystem clock, serial interface slave select input, serial interface data output, timer I/O, and SNOOZE status output.

Only for the P120 and P125 to P127 pins, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

For input to the P125 pin, a CMOS input buffer or a TTL input buffer can be selected using the port input mode register 12 (PIM12).

For output from the P120 pin, CMOS output or N-ch open-drain output can be selected using the port output mode register 12 (POM12).

For the P125 pin, the input threshold level can be specified using the port input threshold control register 12 (PITHL12).

Input to the P120 and P125 pins can be specified as digital I/O or analog input in 1-bit units, using port mode control register 12 (PMC12).

The following operation modes can be specified in 1-bit units.

#### (1) Port mode

P120 and P125 to P127 function as I/O port pins. These pins can be set to input or output port using port mode register 12 (PM12).

P121 to P124 function as input port pins.

#### (2) Control mode

P120 to P127 function as A/D converter analog input, external interrupt request input, resonator connection for the main system clock, resonator connection for the subsystem clock, external clock input for the main system clock, external clock input for the subsystem clock, serial interface slave select input, serial interface data output, timer I/O, and SNOOZE status output.

**(a) ANI24, ANI25**

These are analog input pins of the A/D converter. For details, see **12.10 (5) Analog input (ANIn) pins**.

**(b) INTP1, INTP4**

These are external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**(c) X1, X2**

These are resonator connection pins for the main system clock.

**(d) EXCLK**

This is an external clock input pin for the main system clock.

**(e) XT1, XT2**

These are resonator connection pins for the subsystem clock.

**(f) EXCLKS**

This is an external clock input pin for the subsystem clock.

**(g) $\overline{\text{SSI01}}$**

This is a slave select input pin of the CSI01 (SPI01) serial interface.

**(h) SO01**

This is a serial data output pin of the CSI01 serial interface.

**(i) TI01, TI03, TI07**

These are pins for inputting an external count clock/capture trigger to 16-bit timers.

**(j) TO01, TO03, TO07**

These are timer output pins of 16-bit timers.

**(k) SNZOUT1**

This is a SNOOZE status output pin.

**(l) TRDIOB0, TRDIOD0**

These are timer I/O pins of timer RD.

### 2.2.12 P130, P137 (Port 13)

P130 functions as an output port. P137 functions as an input port. These pins also function as external interrupt request input and reset output.

**(1) Port mode**

P130 functions as an output port. P137 functions as an input port.

**(2) Control mode**

P130 and P137 function as external interrupt request input and reset output.

**(a) INTP0**

This is an external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**(b) RESOUT**

This is a reset output pin.

### 2.2.13 P140 (Port 14)

P140 functions as an I/O port. This pin also functions as clock/buzzer output.

Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 14 (PU14).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P140 functions as an I/O port. This pin can be set to input or output port in 1-bit units using port mode register 14 (PM14).

**(2) Control mode**

P140 functions as clock/buzzer output.

**(a) PCLBUZ0**

This is a clock/buzzer output pin.

### 2.2.14 P150 to P157 (Port 15)

P150 to P157 function as an I/O port. These pins also function as serial interface data I/O, clock I/O, timer I/O, slave select input, and SNOOZE status output. These pins are provided only in the 100-pin products of RL78/F14.

Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 15 (PU15).

For the P150, P152, and P153 pins, the input threshold level can be specified using the port input threshold control register 15 (PITHL15).

**(1) Port mode**

P150 to P157 function as an I/O port. These pins can be set to input or output port in 1-bit units using port mode register 15 (PM15).

**(2) Control mode**

P150 to P157 function as serial interface slave select input, data I/O, clock I/O, and SNOOZE status output.

**(a) SSI11**

This is a slave select input pin of the CSI11 (SPI11) serial interface.

**(b) SI11**

This is a serial data input pin of the CSI11 serial interface.

**(c) SO11**

This is a serial data output pin of the CSI11 serial interface.

**(d) SCK11**

This is a serial clock I/O pin of the CSI11 serial interface.

**(e) SNZOUT4 to SNZOUT7**

These are SNOOZE status output pins.

### 2.2.15 $V_{DD}$, $EV_{DD0}$, $EV_{DD1}$, $V_{SS}$, $EV_{SS0}$, $EV_{SS1}$

**(1) $V_{DD}$, $EV_{DD0}$, $EV_{DD1}$**

$V_{DD}$ is a positive power supply pin for the P33, P34, P80 to P87, P90 to P97 Note, P100 to P105, P121 to P124, P137 and the pins other than ports.

$EV_{DD0}$ and $EV_{DD1}$ are positive power supply pins for ports other than P33, P34, P80 to P87, P90 to P97 Note, P100 to P105, P121 to P124, and P137. $EV_{DD1}$ is provided only in the 100-pin products of RL78/F14.

**Note** In products with 48 or fewer pins, $V_{DD}$ is the positive power supply pin for all port pins.
In products with 64 or more pins, $V_{DD}$ and $EV_{DD0}$ are the positive power supply pins for P92 to P95 of products of groups B to E and of group A, respectively.
In products with 64 or more pins, $V_{DD}$ and $EV_{DD0}$ are the positive power supply pins for P96 and P97 of products of group E and of groups A to D, respectively.

**(2) $V_{SS}$, $EV_{SS0}$, $EV_{SS1}$**

$V_{SS}$ is a ground potential pin for P33, P34, P80 to P87, P90 to P97 Note, P100 to P105, P121 to P124, P137 and the pins other than ports.

$EV_{SS0}$ and $EV_{SS1}$ are ground potential pins for ports other than P33, P34, P80 to P87, P90 to P97 Note, P100 to P105, P121 to P124, and P137. $EV_{SS1}$ is provided only in the 100-pin products of RL78/F14.

**Note** In products with 48 or fewer pins, $V_{SS}$ is the ground potential pin for all port pins.
In products with 64 or more pins, Vss and $EV_{SS0}$ are the ground potential pins for P92 to P95 of products of groups B to E and of group A, respectively.
In products with 64 or more pins, Vss and $EV_{SS0}$ are the ground potential pins for P96 and P97 of products of group E and of groups A to D, respectively.

**Remark** Use bypass capacitors (about 0.1 μF) as noise and latch-up countermeasures with relatively thick wires at the shortest distance to $V_{DD}$ to $V_{SS}$, $EV_{DD0}$ to $EV_{SS0}$ and $EV_{DD1}$ to $EV_{SS1}$ lines.

### 2.2.16 $\overline{\text{RESET}}$

This is an active-low system reset input pin.

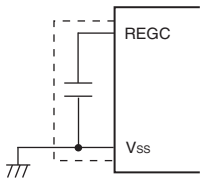
When the external reset pin is not used, connect this pin directly or via a resistor to $V_{DD}$.

When the external reset pin is used, design the circuit based on $V_{DD}$.

### 2.2.17 REGC

This is the pin for connecting regulator output stabilization capacitance for internal operation. Connect this pin to $V_{SS}$ via a capacitor (0.47 to 1 μF).

Use a capacitor with good characteristics because it is used to stabilize internal voltage.




**Caution Keep the wiring length as short as possible for the broken-line part in the above figure.**

## 2.3 Recommended Connection of Unused Pins

Tables 2-6 and 2-7 show the recommended connections of unused pins taking the 100-pin products of RL78/F14 and the 80-pin products of RL78/F13 (CAN and LIN incorporated) as examples.

**Table 2-6. Connection of Unused Pins (100-Pin Products of RL78/F14) (1/3)**

| Pin Name | I/O | Recommended Connection of Unused Pins |
|---|---|---|
| P00/(TI05)/(TO05)/INTP9 | I/O | Input: Independently connect to $EV_{DD0}$ and $EV_{DD1}$, or $EV_{SS0}$ and $EV_{SS1}$ via a resistor.<br>Output: Leave open. |
| P01/(TI04)/(TO04) | | |
| P02/(TI06)/(TO06) | | |
| P03/(RTC1HZ) | | |
| P10/TI13/TO13/TRJO0/$\overline{\text{SCK10}}$/SCL10/ LTXD1/CTXD0 | | |
| P11/TI12/TO12/(TRDIOB0)/SI10/SDA10/ RXD1/LRXD1/CRXD0 | | |
| P12/TI11/TO11/(TRDIOD0)/INTP5/SO10/ TXD1/SNZOUT3 | | |
| P13/TI04/TO04/TRDIOA0/TRDCLK0/SI01/ SDA01/LTXD0 | | |
| P14/TI06/TO06/TRDIOC0/$\overline{\text{SCK01}}$/SCL01/ LRXD0 | | |
| P15/TI05/TO05/TRDIOA1/(TRDIOA0)/ (TRDCLK0)/SO00/TXD0/TOOLTXD/ RTC1HZ | | |
| P16/TI02/TO02/TRDIOC1/SI00/SDA00/ RXD0/TOOLRXD | | |
| P17/TI00/TO00/TRDIOB1/$\overline{\text{SCK00}}$/SCL00/ INTP3 | | |
| P30/TI01/TO01/TRDIOD1/$\overline{\text{SSI00}}$/INTP2/ SNZOUT0 | | |
| P31/TI14/TO14/STOPST/(INTP2) | | |
| P32/TI16/TO16/INTP7 | | |
| P33/$AV_{REFP}$/ANI0 | | Input: Independently connect to $V_{DD}$ or $V_{SS}$ via a resistor.<br>Output: Leave open. |
| P34/$AV_{REFM}$/ANI1 | | |
| P40/TOOL0 | | Input: Independently connect to $EV_{DD0}$ and $EV_{DD1}$, or $EV_{SS0}$ and $EV_{SS1}$ via a resistor.<br>Output: Leave open. |
| P41/TI10/TO10/TRJIO0/VCOUT0/ SNZOUT2 | | |
| P42/(LTXD0) | | |
| P43/(LRXD0) | | |
| P44/(TI07)/(TO07) | | |
| P45/(TI10)/(TO10) | | |
| P46/(TI12)/(TO12) | | |
| P47/INTP13 | | |

**Remark** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIOR).

**Table 2-6. Connection of Unused Pins (100-Pin Products of RL78/F14) (2/3)**

| Pin Name | I/O | Recommended Connection of Unused Pins |
|---|---|---|
| P50/($\overline{\text{SSI01}}$)/(INTP3) | I/O | Input: Independently connect to $EV_{DD0}$ and $EV_{DD1}$, or $EV_{SS0}$ and $EV_{SS1}$ via a resistor.<br>Output: Leave open. |
| P51/(SO01)/INTP11 | | |
| P52/($\overline{\text{SCK01}}$)/(STOPST) | | |
| P53/(SI01)/INTP10 | | |
| P54/(TI11)/(TO11)/$\overline{\text{SSI10}}$ | | |
| P55/(TI13)/(TO13) | | |
| P56/(TI15)/(TO15)/(SNZOUT1) | | |
| P57/(TI17)/(TO17)/(SNZOUT0) | | |
| P60/($\overline{\text{SCK00}}$)/(SCL00) | | |
| P61/(SI00)/(SDA00)/(RXD0) | | |
| P62/(SO00)/(TXD0)/SCLA0 | | |
| P63/($\overline{\text{SSI00}}$)/SDAA0 | | |
| P64/(TI14)/(TO14)/(SNZOUT3) | | |
| P65/(TI16)/(TO16)/(SNZOUT2) | | |
| P66/(TI00)/(TO00) | | |
| P67/(TI02)/(TO02) | | |
| P70/ANI26/KR0/TI15/TO15/INTP8/ SI11/SDA11/SNZOUT4 | | |
| P71/ANI27/KR1/TI17/TO17/INTP6/ $\overline{\text{SCK11}}$/SCL11/SNZOUT5 | | |
| P72/ANI28/KR2/(CTXD0)/SO11/SNZOUT6 | | |
| P73/ANI29/KR3/(CRXD0)/$\overline{\text{SSI11}}$/ SNZOUT7 | | |
| P74/ANI30/KR4/(SO10)/(TXD1) | | |
| P75/KR5/(SI10)/(RXD1) | | |
| P76/KR6/($\overline{\text{SCK10}}$) | | |
| P77/KR7/($\overline{\text{SSI10}}$)/INTP12 | | |
| P80/ANI2/ANO0 | | Input: Independently connect to $V_{DD}$ or $V_{SS}$ via a resistor.<br>Output: Leave open. |
| P81/ANI3/IVCMP00 | | |
| P82/ANI4/IVCMP01 | | |
| P83/ANI5/IVCMP02 | | |
| P84/ANI6/IVCMP03 | | |
| P85/ANI7/IVREF0 | | |
| P86/ANI8 | | |
| P87/ANI9 | | |
| P90/ANI10 | | |
| P91/ANI11 | | |
| P92/ANI12 | | |
| P93/ANI13 | | |
| P94/ANI14 | | |
| P95/ANI15 | | |
| P96/ANI16 | | |
| P97/ANI17 | | |

**Remark** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIOR).

**Table 2-6. Connection of Unused Pins (100-Pin Products of RL78/F14) (3/3)**

| Pin Name | I/O | Recommended Connection of Unused Pins |
|---|---|---|
| P100/ANI18 | I/O | Input: Independently connect to $V_{DD}$ or $V_{SS}$ via a resistor.<br>Output: Leave open. |
| P101/ANI19 | | |
| P102/ANI20 | | |
| P103/ANI21 | | |
| P104/ANI22 | | |
| P105/ANI23 | | |
| P106/(LTXD1) | | Input: Independently connect to $EV_{DD0}$ and $EV_{DD1}$, or $EV_{SS0}$ and $EV_{SS1}$ via a resistor.<br>Output: Leave open. |
| P107/(LRXD1) | | |
| P120/ANI25/TI07/TO07/TRDIOD0/SO01/INTP4 | | |
| P121/X1 | Input | Independently connect to $V_{DD}$ or $V_{SS}$ via a resistor. |
| P122/X2/EXCLK | | |
| P123/XT1 | | |
| P124/XT2/EXCLKS | | |
| P125/ANI24/TI03/TO03/TRDIOB0/$\overline{\text{SSI01}}$/INTP1/SNZOUT1 | I/O | Input: Independently connect to $EV_{DD0}$ and $EV_{DD1}$, or $EV_{SS0}$ and $EV_{SS1}$ via a resistor.<br>Output: Leave open. |
| P126/(TI01)/(TO01) | | |
| P127/(TI03)/(TO03) | | |
| P130/RESOUT | Output | Leave open. |
| P137/INTP0 | Input | Independently connect to $V_{DD}$ or $V_{SS}$ via a resistor. |
| P140/PCLBUZ0 | I/O | Input: Independently connect to $EV_{DD0}$ and $EV_{DD1}$, or $EV_{SS0}$ and $EV_{SS1}$ via a resistor.<br>Output: Leave open. |
| P150/($\overline{\text{SSI11}}$) | | |
| P151/(SO11) | | |
| P152/(SI11) | | |
| P153/($\overline{\text{SCK11}}$) | | |
| P154/(SNZOUT7) | | |
| P155/(SNZOUT6) | | |
| P156/(SNZOUT5) | | |
| P157/(SNZOUT4) | | |
| $\overline{\text{RESET}}$ | Input | Connect to $V_{DD}$ directly or via a resistor. |
| REGC | — | Connect to $V_{SS}$ via a capacitor (0.47 to 1 μF). |

**Remark** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIOR).

**Table 2-7. Connection of Unused Pins (80-Pin Products of RL78/F13 (CAN and LIN incorporated)) (1/3)**

| Pin Name | I/O | Recommended Connection of Unused Pins |
|---|---|---|
| P00/(TI05)/(TO05)/INTP9 | I/O | Input: Independently connect to $EV_{DD0}$ or $EV_{SS0}$ via a resistor.<br>Output: Leave open. |
| P01/(TI04)/(TO04) | | |
| P02/(TI06)/(TO06) | | |
| P10/TI13/TO13/TRJO0/$\overline{SCK10}$/SCL10/<br>CTXD0 | | |
| P11/TI12/TO12/(TRDIOB0)/SI10/SDA10/<br>RXD1/CRXD0 | | |
| P12/TI11/TO11/(TRDIOD0)/INTP5/SO10/<br>TXD1/SNZOUT3 | | |
| P13/TI04/TO04/TRDIOA0/TRDCLK0/SI01/<br>SDA01/LTXD0 | | |
| P14/TI06/TO06/TRDIOC0/$\overline{SCK01}$/SCL01/<br>LRXD0 | | |
| P15/TI05/TO05/TRDIOA1/(TRDIOA0)/<br>(TRDCLK0)/SO00/TXD0/TOOLTXD/<br>RTC1HZ | | |
| P16/TI02/TO02/TRDIOC1/SI00/SDA00/<br>RXD0/TOOLRXD | | |
| P17/TI00/TO00/TRDIOB1/$\overline{SCK00}$/SCL00/<br>INTP3 | | |
| P30/TI01/TO01/TRDIOD1/$\overline{SSI00}$/INTP2/<br>SNZOUT0 | | |
| P31/STOPST/(INTP2) | | |
| P32/INTP7 | | |
| P33/$AV_{REFP}$/ANI0 | | Input: Independently connect to $V_{DD}$ or $V_{SS}$ via a resistor.<br>Output: Leave open. |
| P34/$AV_{REFM}$/ANI1 | | |
| P40/TOOL0 | | Input: Independently connect to $EV_{DD0}$ or $EV_{SS0}$ via a resistor.<br>Output: Leave open. |
| P41/TI10/TO10/TRJIO0/SNZOUT2 | | |
| P42/(LTXD0) | | |
| P43/(LRXD0) | | |
| P44/(TI07)/(TO07) | | |
| P45/(TI10)/(TO10) | | |
| P46/(TI12)/(TO12) | | |
| P47 | | |

**Remark** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIOR).

**Table 2-7. Connection of Unused Pins (80-Pin Products of RL78/F13 (CAN and LIN incorporated)) (2/3)**

| Pin Name | I/O | Recommended Connection of Unused Pins |
|---|---|---|
| P50/($\overline{\text{SSI01}}$)/(INTP3) | I/O | Input: Independently connect to $EV_{DD0}$ or $EV_{SS0}$ via a resistor.<br>Output: Leave open. |
| P51/(SO01)/INTP11 | | |
| P52/($\overline{\text{SCK01}}$)/(STOPST) | | |
| P53/(SI01)/INTP10 | | |
| P54/(TI11)/(TO11)/$\overline{\text{SSI10}}$ | | |
| P55/(TI13)/(TO13) | | |
| P56/(SNZOUT1) | | |
| P57/(SNZOUT0) | | |
| P60/($\overline{\text{SCK00}}$)/(SCL00) | | |
| P61/(SI00)/(SDA00)/(RXD0) | | |
| P62/(SO00)/(TXD0)/SCLA0 | | |
| P63/($\overline{\text{SSI00}}$)/SDAA0 | | |
| P64/(SNZOUT3) | | |
| P65/(SNZOUT2) | | |
| P66/(TI00)/(TO00) | | |
| P67/(TI02)/(TO02) | | |
| P70/KR0/INTP8/SI11/SDA11/SNZOUT4 | | |
| P71/KR1/INTP6/$\overline{\text{SCK11}}$/SCL11/SNZOUT5 | | |
| P72/KR2/(CTXD0)/SO11/SNZOUT6 | | |
| P73/KR3/(CRXD0)/$\overline{\text{SSI11}}$/SNZOUT7 | | |
| P74/KR4/(SO10)/(TXD1) | | |
| P75/KR5/(SI10)/(RXD1) | | |
| P76/KR6/($\overline{\text{SCK10}}$) | | |
| P77/KR7/($\overline{\text{SSI10}}$) | | |
| P80/ANI2 | | Input: Independently connect to $V_{DD}$ or $V_{SS}$ via a resistor.<br>Output: Leave open. |
| P81/ANI3 | | |
| P82/ANI4 | | |
| P83/ANI5 | | |
| P84/ANI6 | | |
| P85/ANI7 | | |
| P86/ANI8 | | |
| P87/ANI9 | | |
| P90/ANI10 | | |
| P91/ANI11 | | |
| P92/ANI12 | | |
| P93/ANI13 | | |
| P94/ANI14 | | |
| P95/ANI15 | | |
| P96/ANI26 | | Input: Independently connect to $EV_{DD0}$ or $EV_{SS0}$ via a resistor.<br>Output: Leave open. |
| P97/ANI27 | | |
| P120/ANI25/TI07/TO07/TRDIOD0/SO01/INTP4 | | |

**Remark** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIOR).

**Table 2-7. Connection of Unused Pins (80-Pin Products of RL78/F13 (CAN and LIN incorporated)) (3/3)**

| Pin Name | I/O | Recommended Connection of Unused Pins |
|---|---|---|
| P121/X1 | Input | Independently connect to $V_{DD}$ or $V_{SS}$ via a resistor. |
| P122/X2/EXCLK | | |
| P123/XT1 | | |
| P124/XT2/EXCLKS | | |
| P125/ANI24/TI03/TO03/TRDIOB0/$\overline{\text{SSI01}}$/ INTP1/SNZOUT1 | I/O | Input: Independently connect to $EV_{DD0}$ or $EV_{SS0}$ via a resistor.<br>Output: Leave open. |
| P126/(TI01)/(TO01) | | |
| P130/RESOUT | Output | Leave open. |
| P137/INTP0 | Input | Independently connect to $V_{DD}$ or $V_{SS}$ via a resistor. |
| P140/PCLBUZ0 | I/O | Input: Independently connect to $EV_{DD0}$ or $EV_{SS0}$ via a resistor.<br>Output: Leave open. |
| $\overline{\text{RESET}}$ | Input | Connect to $V_{DD}$ directly or via a resistor. |
| REGC | — | Connect to $V_{SS}$ via a capacitor (0.47 to 1 μF). |

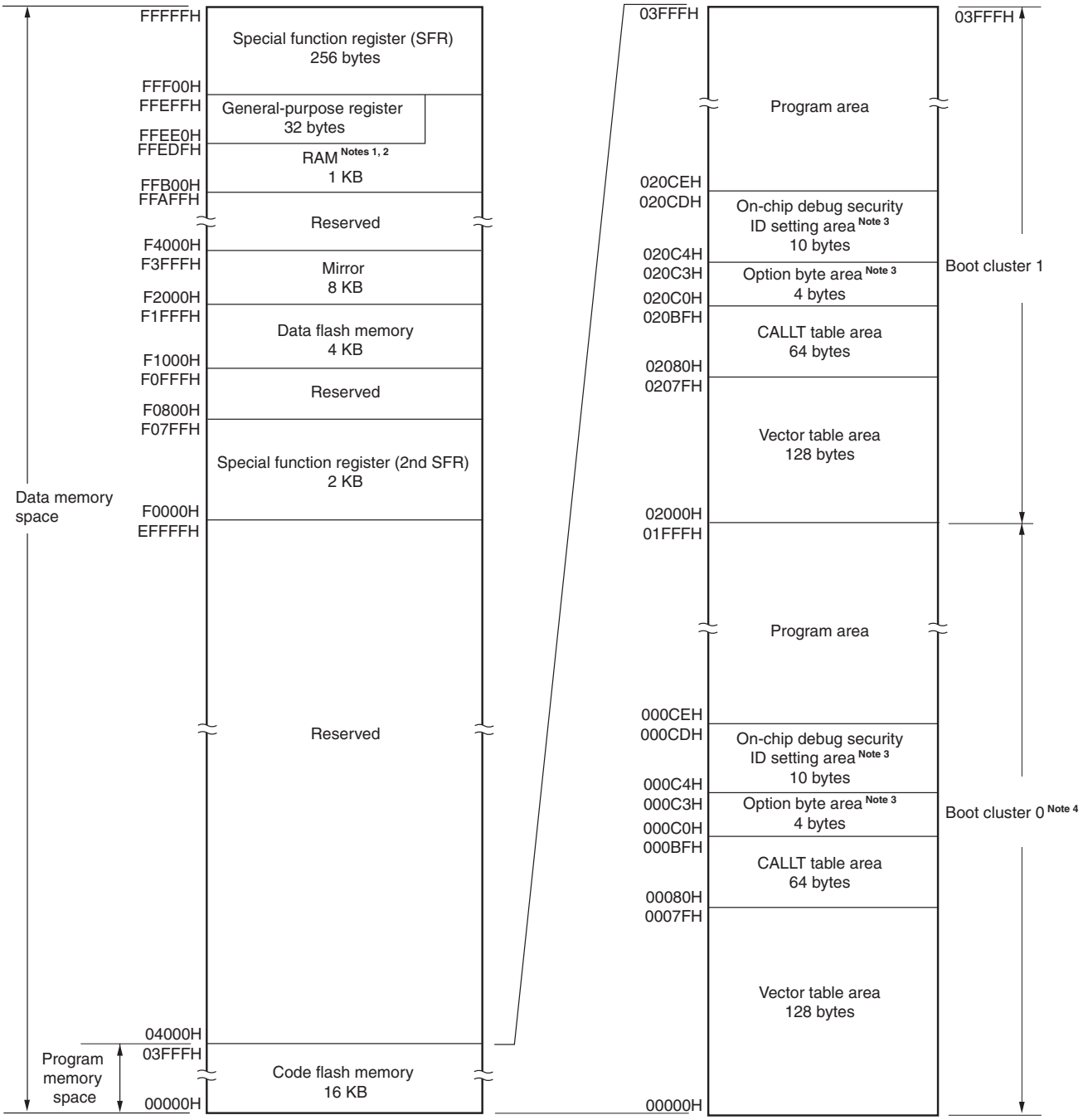
**Remark** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIOR).

# CHAPTER 3 CPU ARCHITECTURE

## 3.1 Memory Space

Products in the RL78/F13 and RL78/F14 can access a 1 MB memory space. Figures 3-1 to 3-17 show the memory maps.

**Figure 3-1. Memory Map (R5F10AmA (m = 6, A, B, G))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

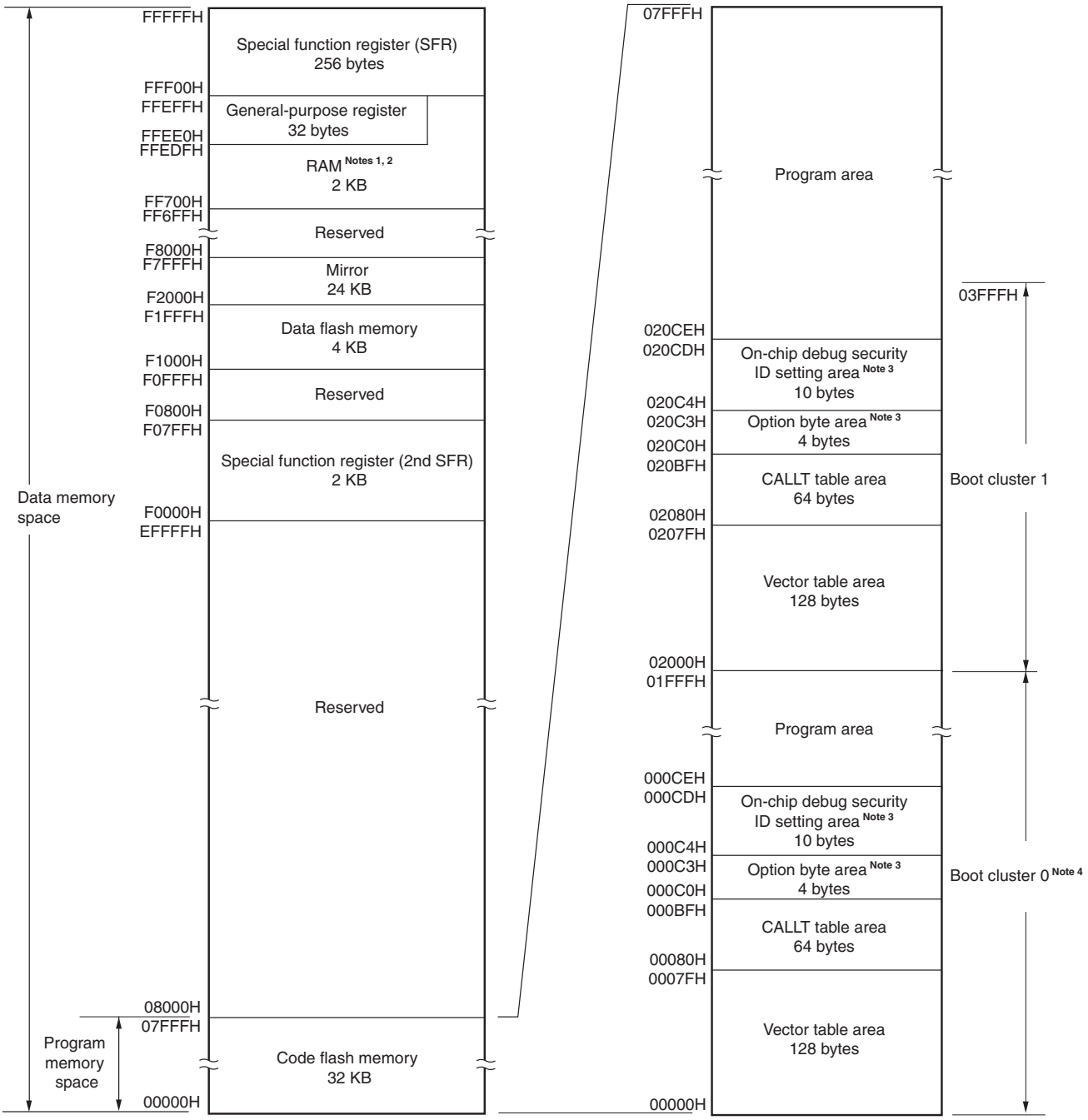
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-2. Memory Map (R5F10AmC (m = 6, A, B, G, L))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

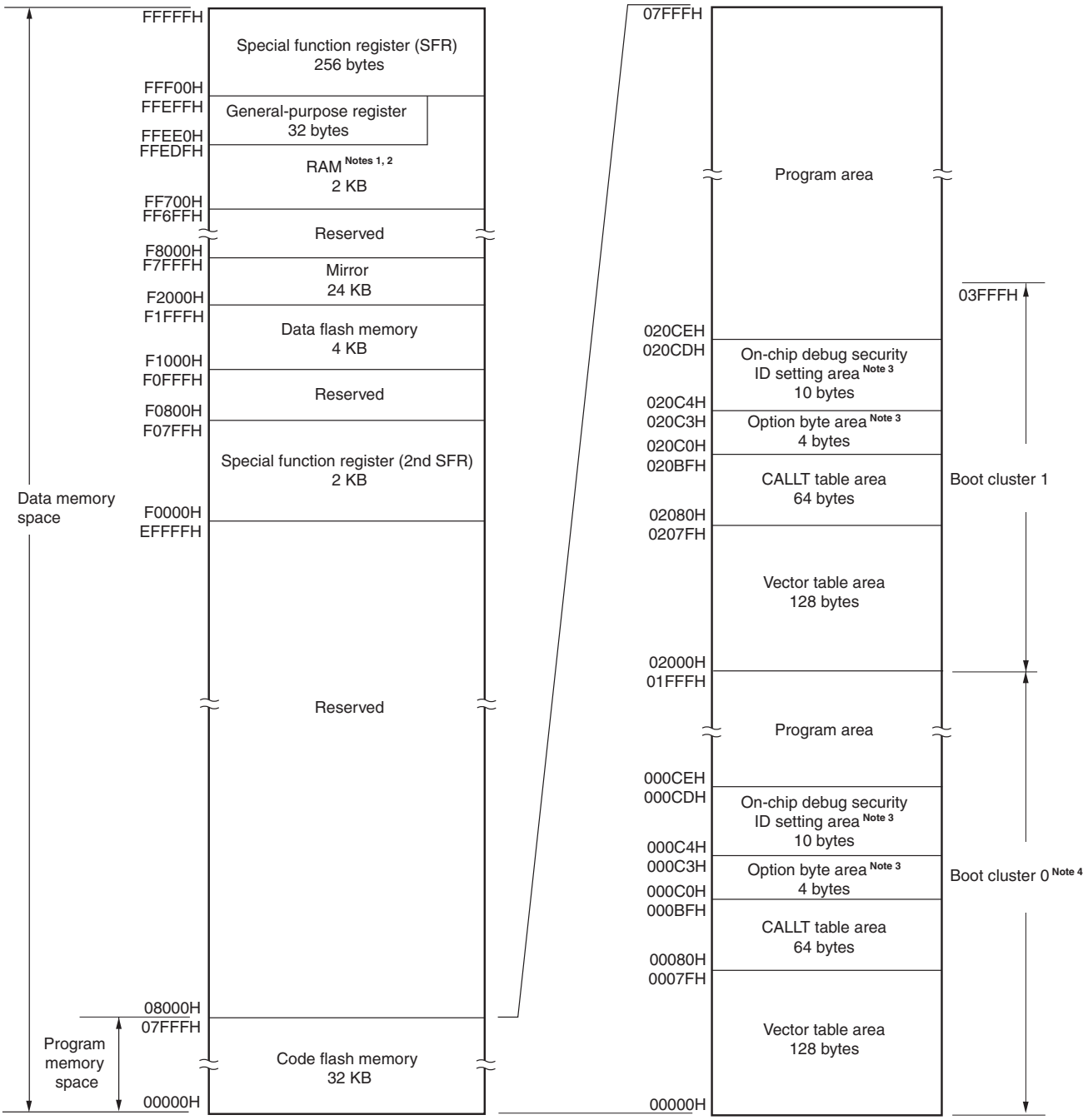
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-3. Memory Map (R5F10BmC (m = A, B, G, L))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

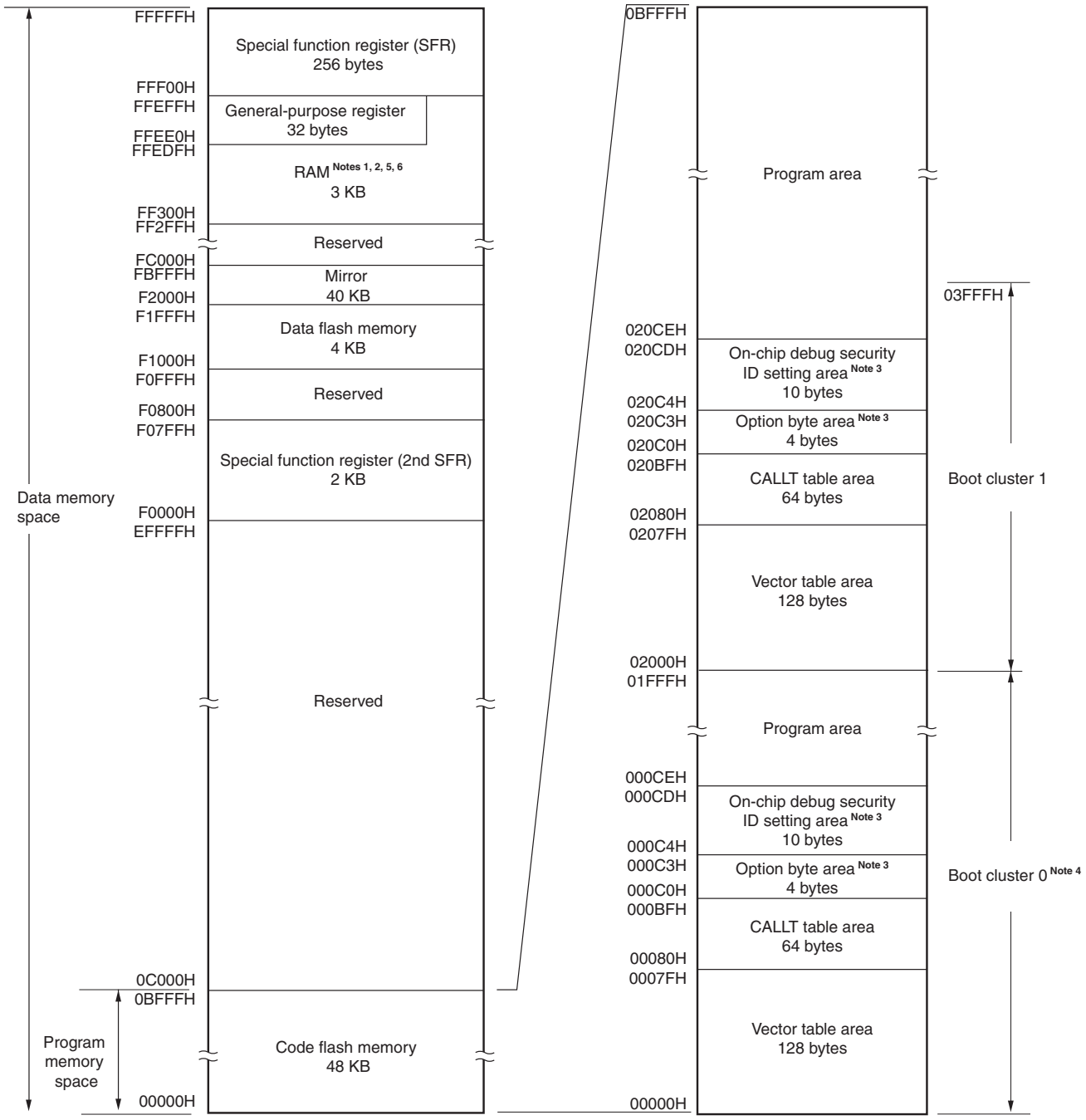
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (**see 30.6 Security Settings**).

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-4. Memory Map (R5F10AmD (m = 6, A, B, G, L))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

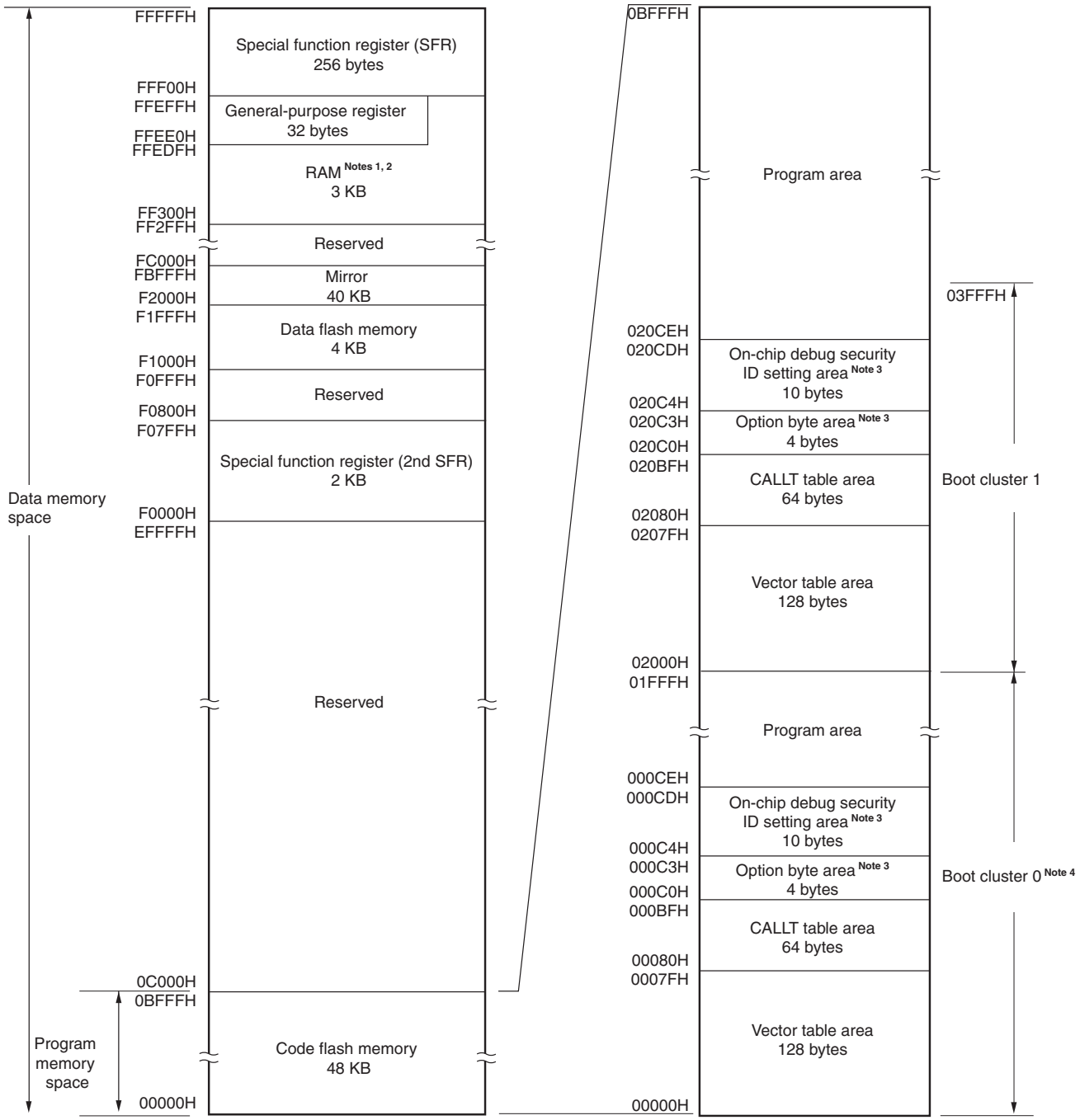
**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

**5.** The debugger uses the area FF300H to FF37FH to store the result of tracing when the tracing function for on-chip debugging is in use. Accordingly, use of this area is prohibited while the tracing function is in use.

**6.** The debugger uses the area FF400H to FF42FH as a working area when the hot plug-in function is in use or when the DTC is in use for the real-time RAM monitor (RRM) or dynamic memory modification (DMM) function. Accordingly, use of this area is prohibited while the hot plug-in function is in use or the DTC is in use for the RRM or DMM function.

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-5. Memory Map (R5F10BmD (m = A, B, G, L))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

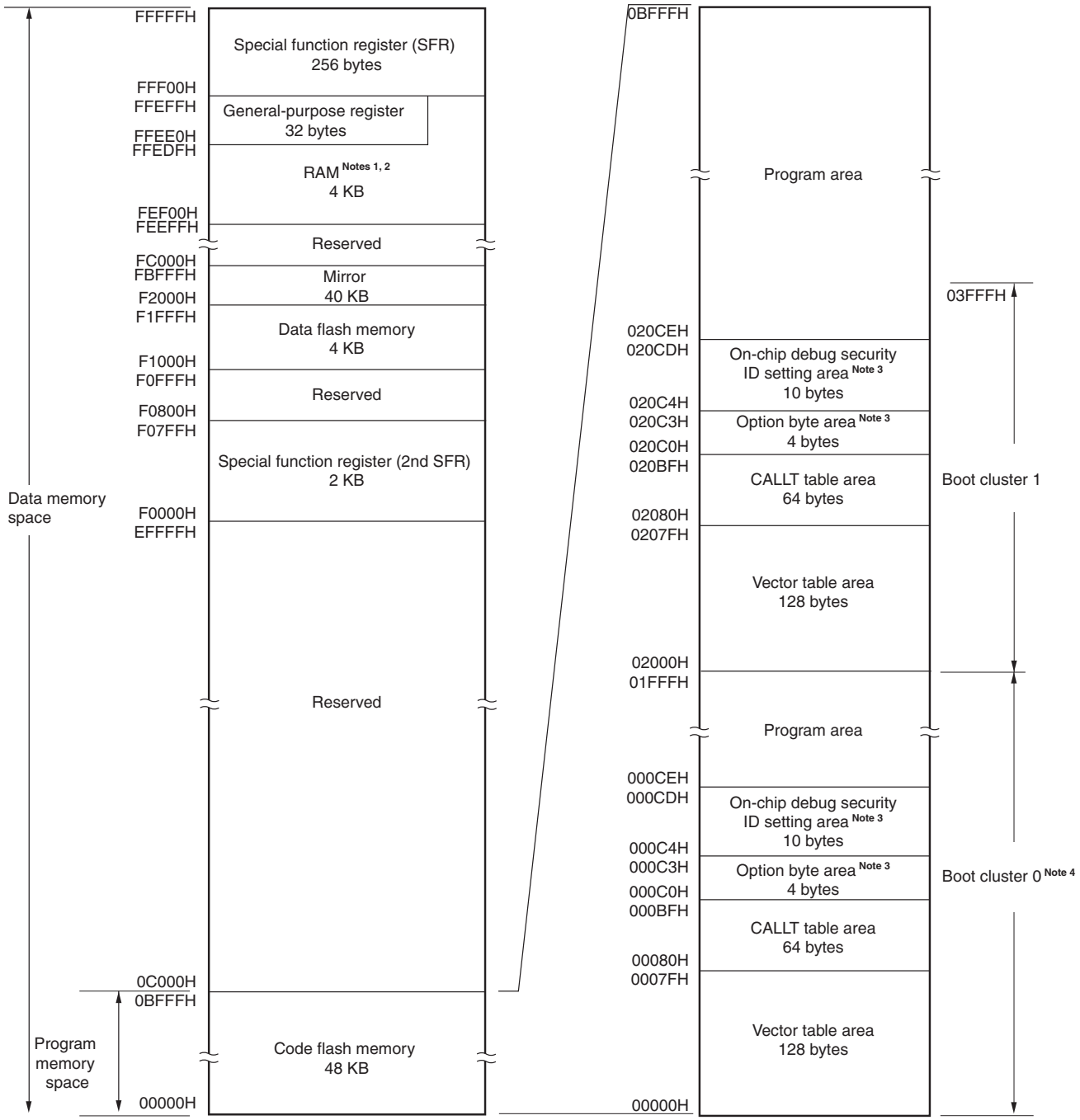
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-6. Memory Map (R5F10PmD (m = A, B, G))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

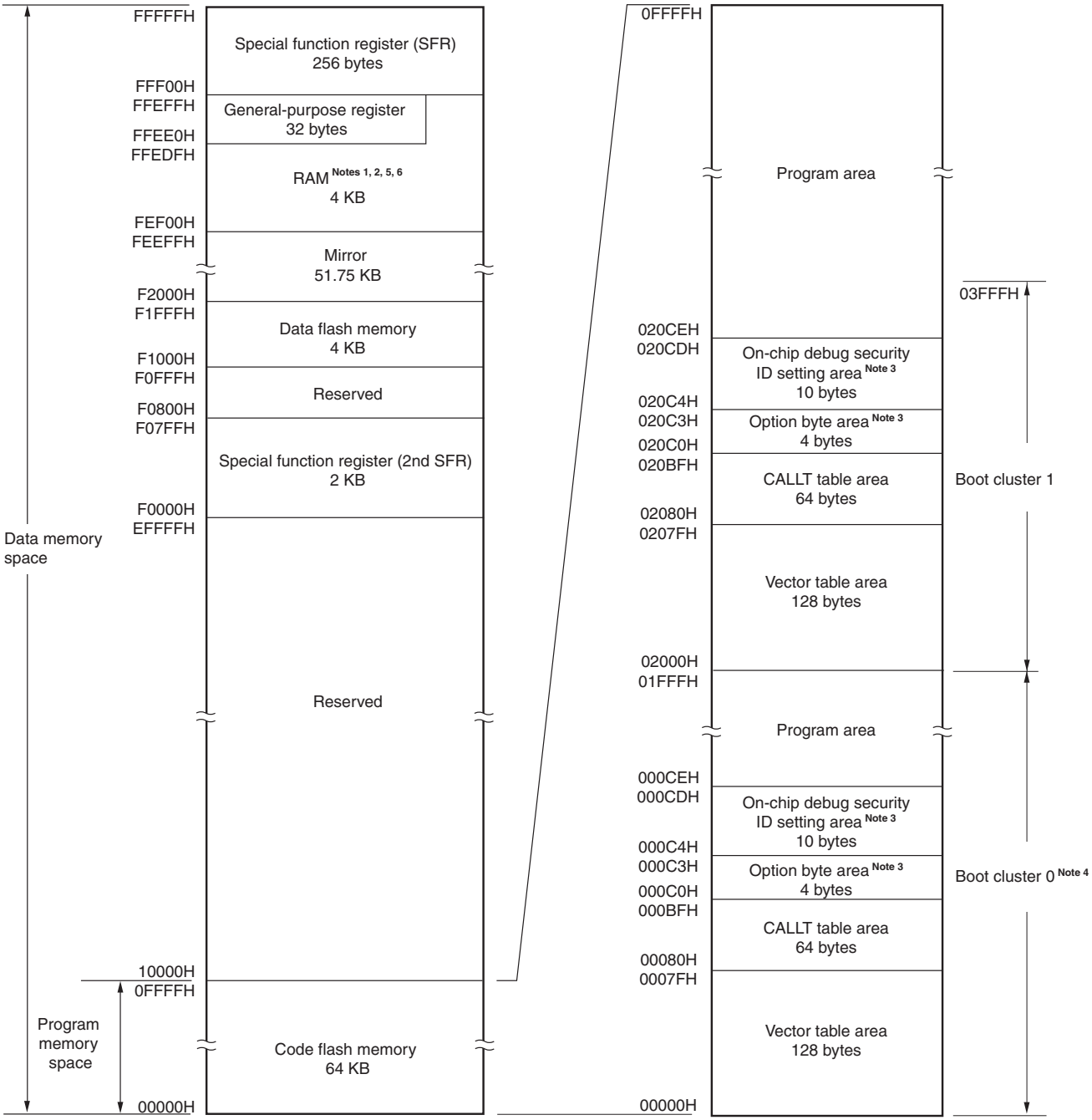
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-7. Memory Map (R5F10AmE (m = 6, A, B, G, L))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory. Also, use of the area FEF00H to FF2FFH is prohibited, because this area is used for each library. However, the area to which this prohibition applies may vary with the version of the library. For details, refer to the manual for the individual library.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

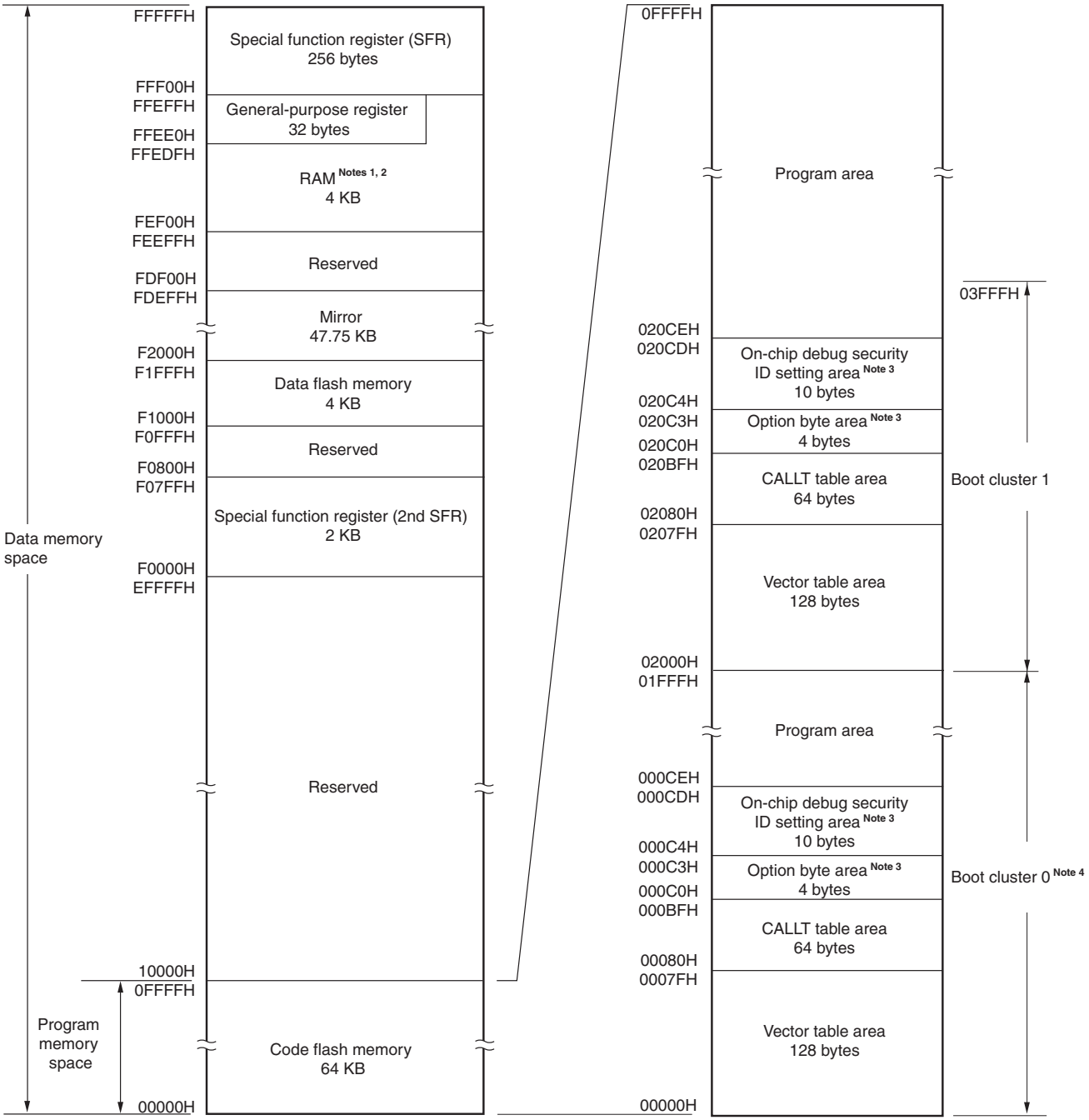
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).
5. The debugger uses the area FF300H to FF37FH to store the result of tracing when the tracing function for on-chip debugging is in use. Accordingly, use of this area is prohibited while the tracing function is in use.
6. The debugger uses the area FF400H to FF42FH as a working area when the hot plug-in function is in use or when the DTC is in use for the real-time RAM monitor (RRM) or dynamic memory modification (DMM) function. Accordingly, use of this area is prohibited while the hot plug-in function is in use or the DTC is in use for the RRM or DMM function.

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-8. Memory Map (R5F10AME, R5F10BmE (m = A, B, G, L, M))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

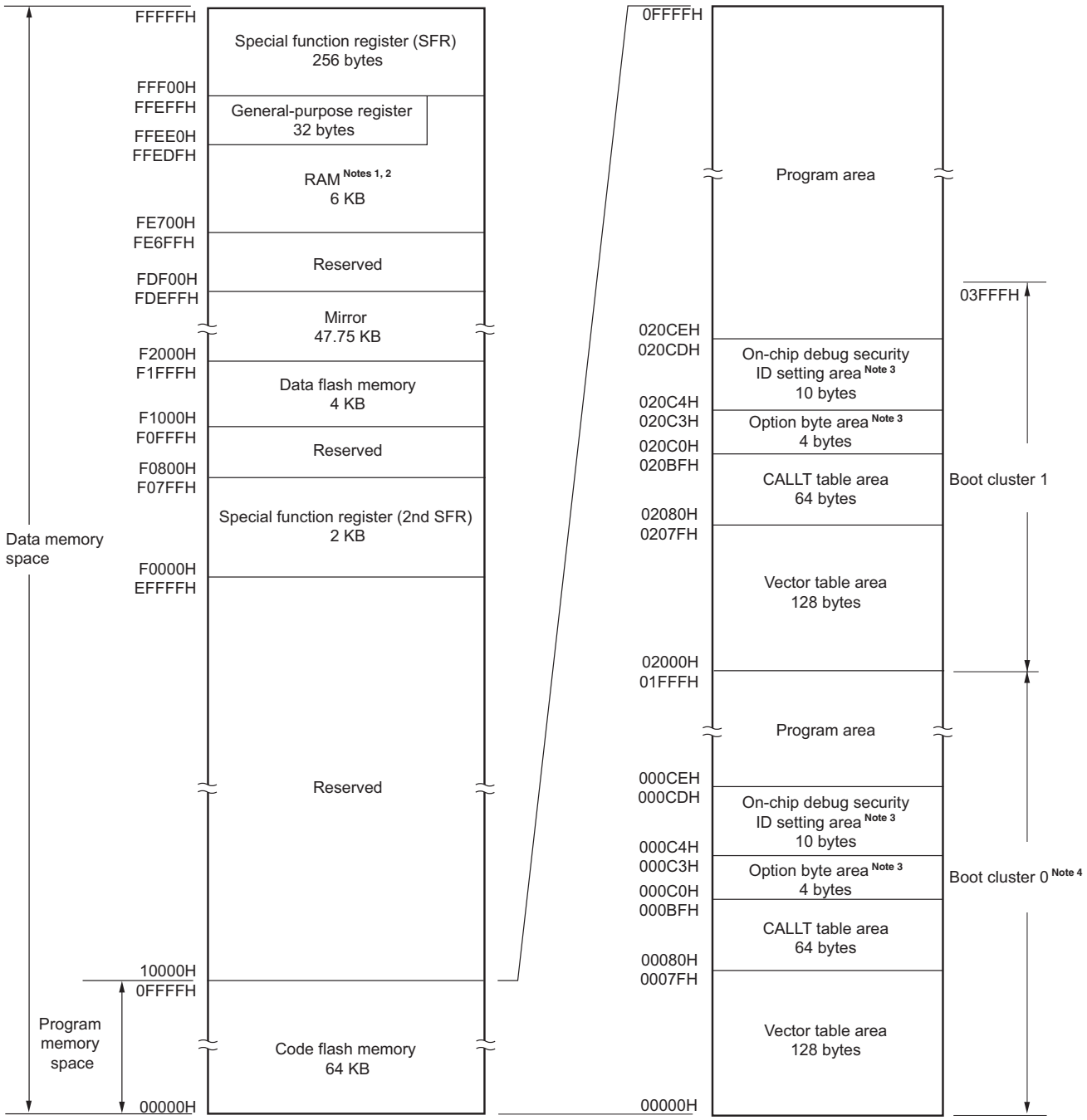
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-9. Memory Map (R5F10PmE (m = A, B, G, L, M))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

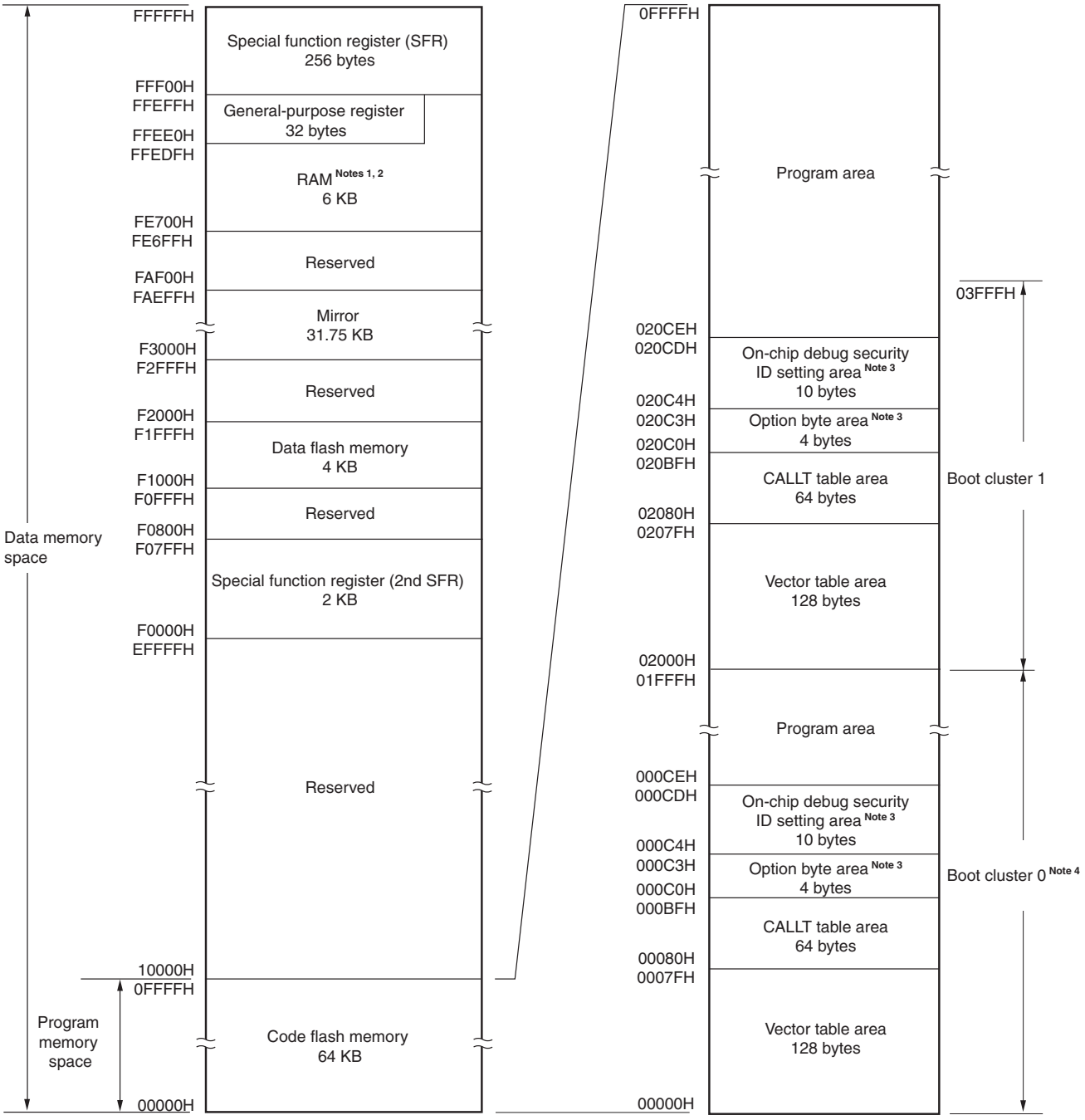
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-10. Memory Map (R5F10PPE)**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

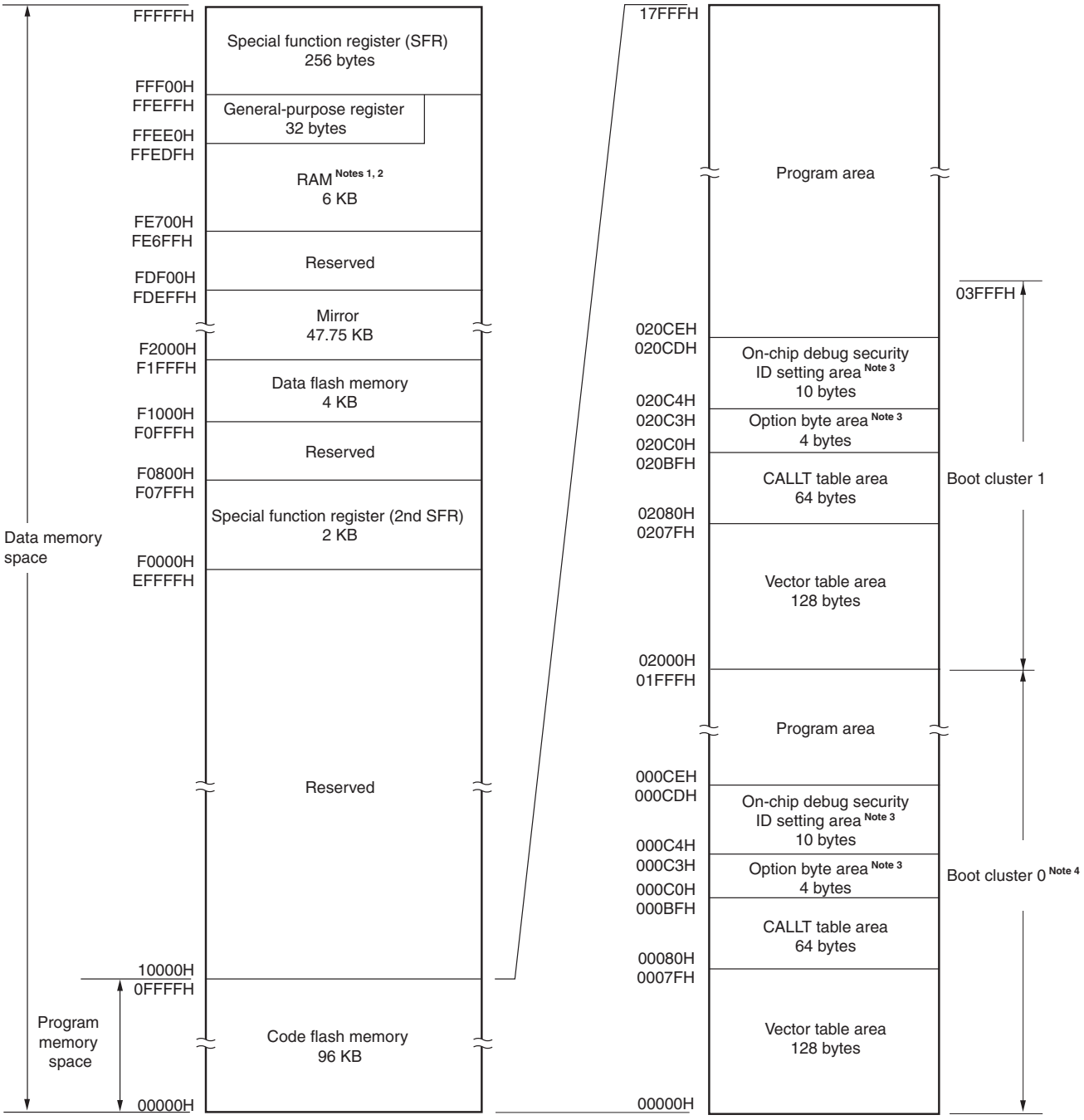
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-11. Memory Map (R5F10AmF (m = G, L, M), R5F10BmF (m = A, B, G, L, M))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

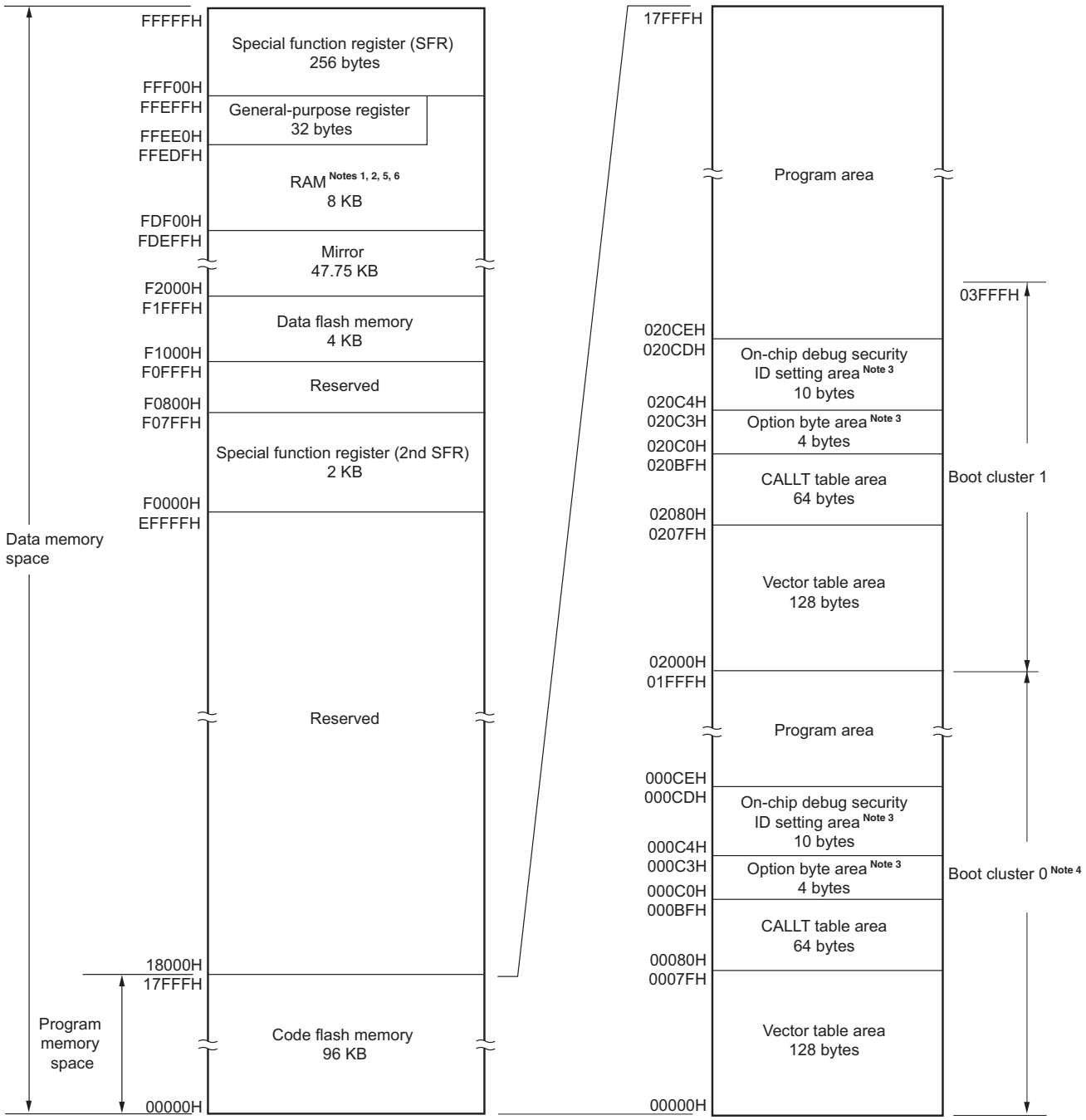
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-12. Memory Map (R5F10PmF (m = G, L, M))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory. Also, use of the area FDF00H to FE2FFH is prohibited, because this area is used for each library. However, the area to which this prohibition applies may vary with the version of the library. For details, refer to the manual for the individual library.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

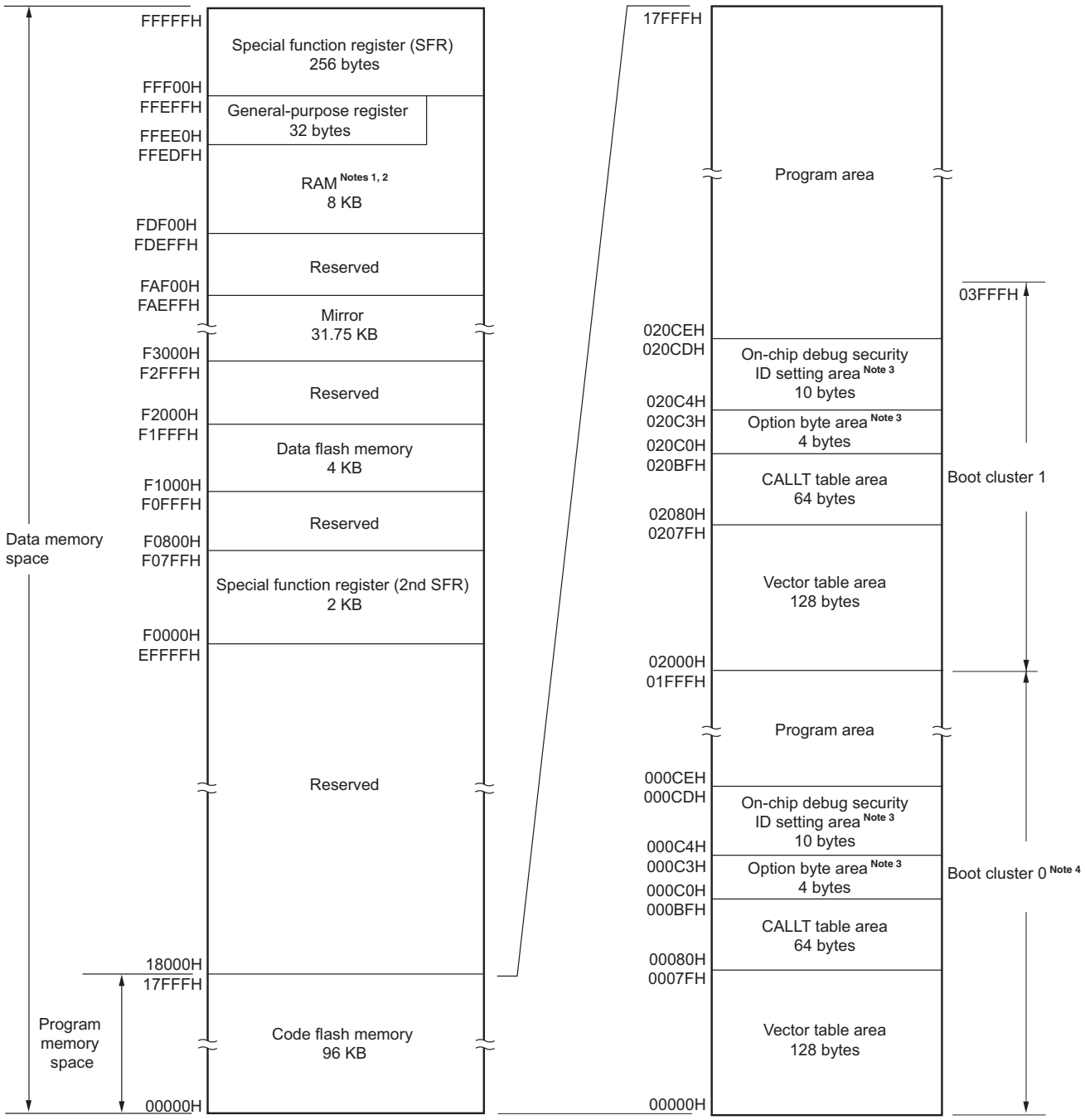
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

5. The debugger uses the area FE300H to FE4FFH to store the result of tracing when the tracing function for on-chip debugging is in use. Accordingly, use of this area is prohibited while the tracing function is in use.
6. The debugger uses the area FE500H to FE52FH as a working area when the hot plug-in function is in use or when the DTC is in use for the real-time RAM monitor (RRM) or dynamic memory modification (DMM) function. Accordingly, use of this area is prohibited while the hot plug-in function is in use or the DTC is in use for the RRM or DMM function.

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-13. Memory Map (R5F10PPF)**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

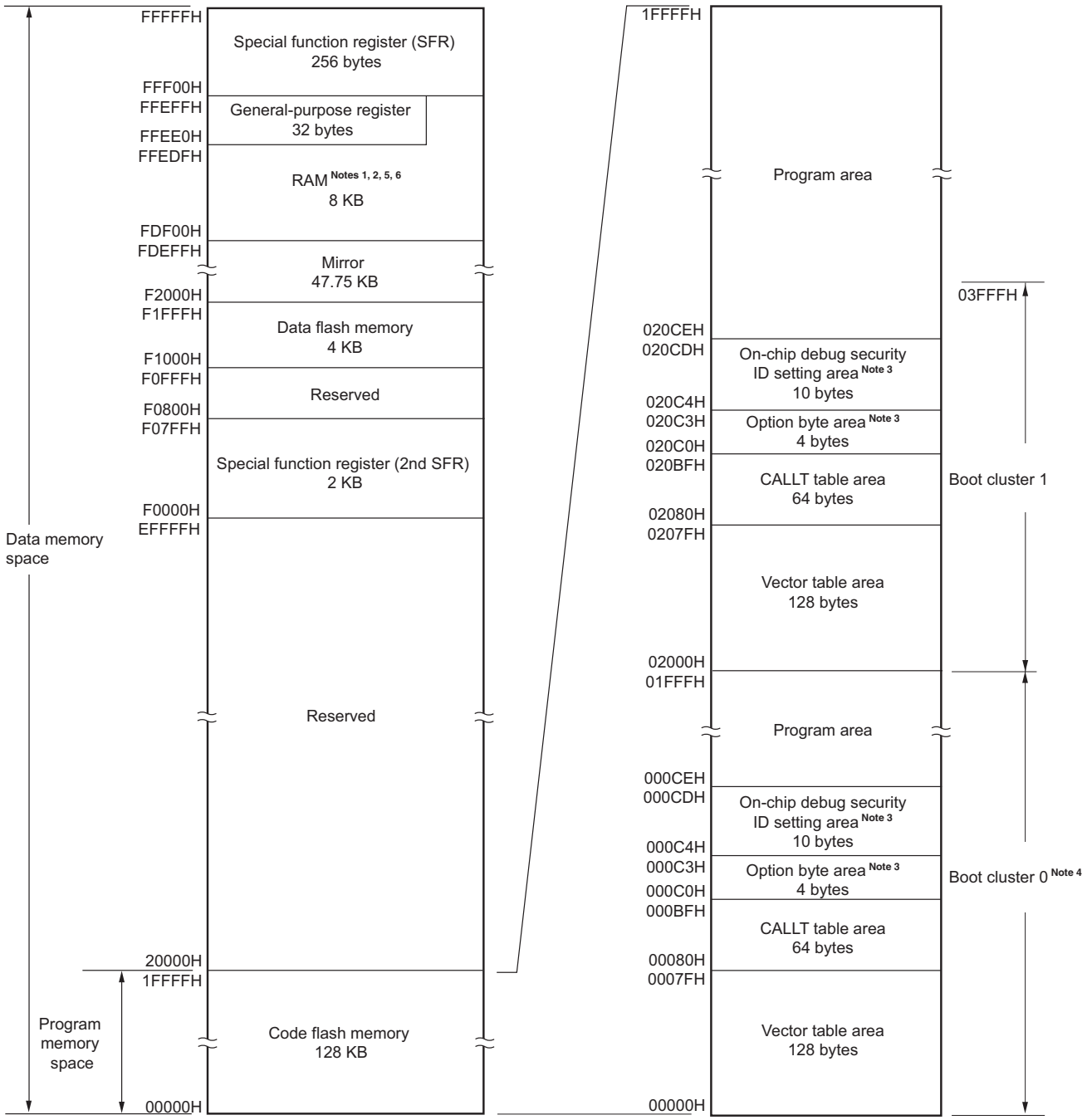
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-14. Memory Map (R5F10AmG (m = G, L, M), R5F10BnG (n = A, B, G, L, M))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory. Also, use of the area FDF00H to FE2FFH is prohibited, because this area is used for each library. However, the area to which this prohibition applies may vary with the version of the library. For details, refer to the manual for the individual library.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

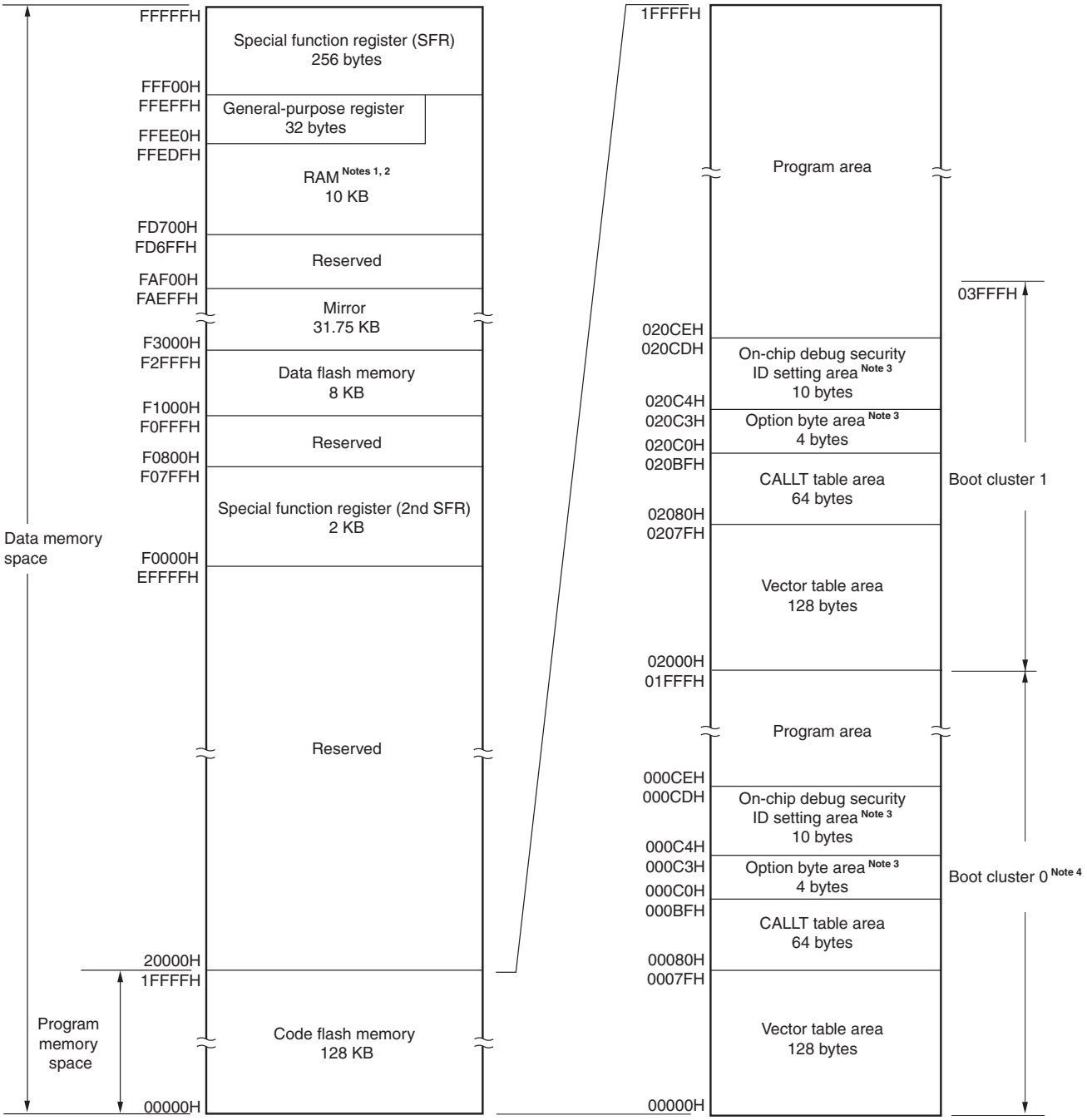
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

5. The debugger uses the area FE300H to FE4FFH to store the result of tracing when the tracing function for on-chip debugging is in use. Accordingly, use of this area is prohibited while the tracing function is in use.
6. The debugger uses the area FE500H to FE52FH as a working area when the hot plug-in function is in use or when the DTC is in use for the real-time RAM monitor (RRM) or dynamic memory modification (DMM) function. Accordingly, use of this area is prohibited while the hot plug-in function is in use or the DTC is in use for the RRM or DMM function.

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-15. Memory Map (R5F10PmG (m = G, L, M, P))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

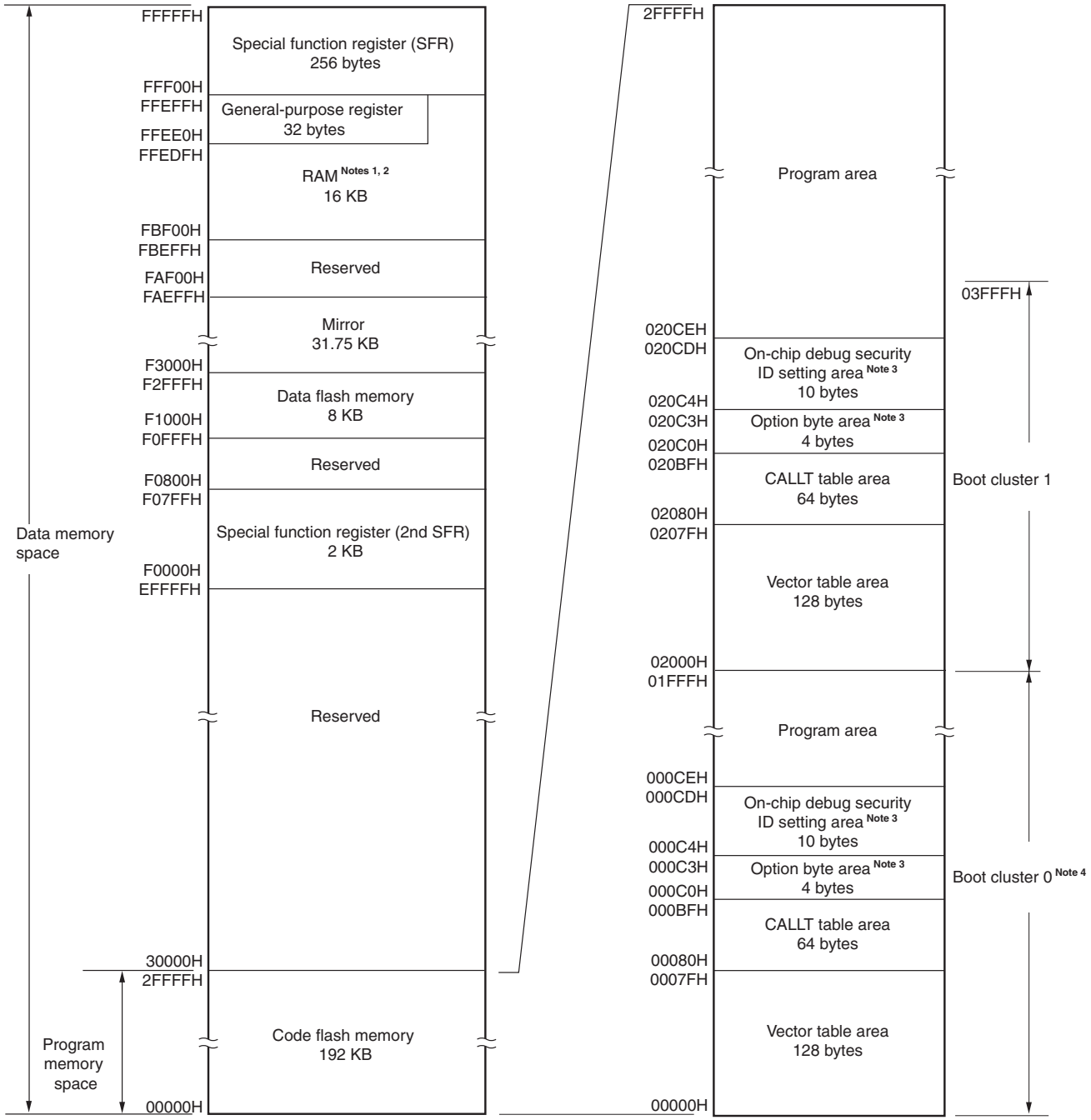
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-16. Memory Map (R5F10PmH (m = G, L, M, P))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

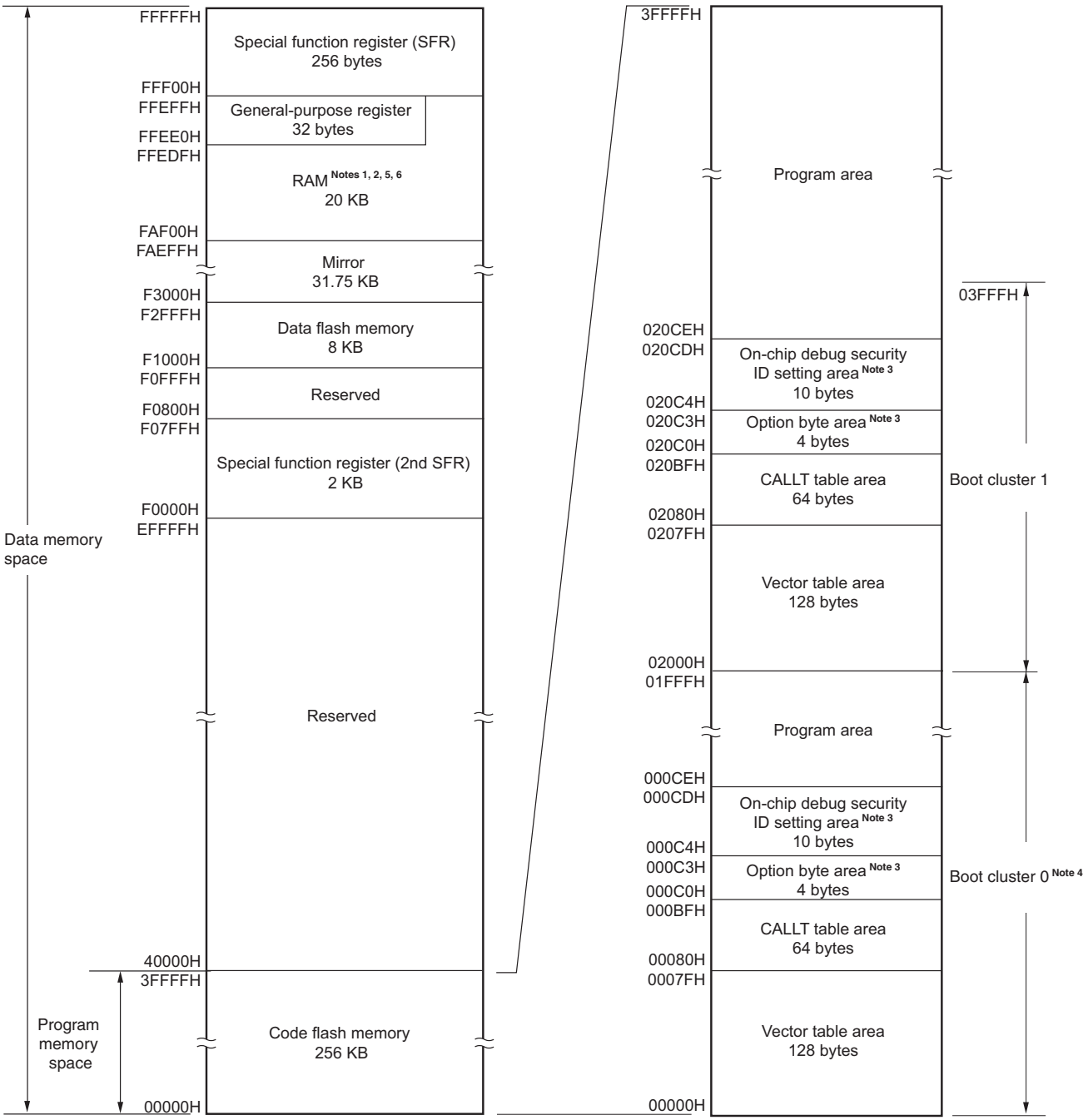
**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-17. Memory Map (R5F10PmJ (m = G, L, M, P))**




**Notes 1.** Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory. Also, use of the area FAF00H to FB2FFH is prohibited, because this area is used for each library. However, the area to which this prohibition applies may vary with the version of the library. For details, refer to the manual for the individual library.

**2.** Instructions can be executed from the RAM area excluding the general-purpose register area.

**3.** When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

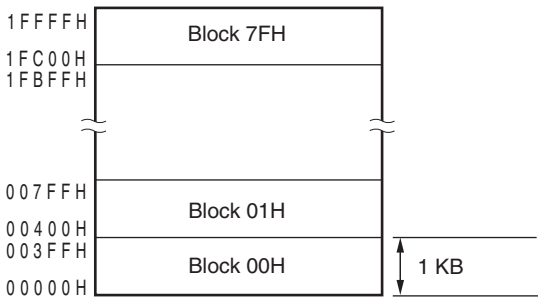
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 020C0H to 020C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 020C4H to 020CDH.

**4.** Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.6 Security Settings**).

5. The debugger uses the area FB300H to FB4FFH to store the result of tracing when the tracing function for on-chip debugging is in use. Accordingly, use of this area is prohibited while the tracing function is in use.
6. The debugger uses the area FB500H to FB52FH as a working area when the hot plug-in function is in use or when the DTC is in use for the real-time RAM monitor (RRM) or dynamic memory modification (DMM) function. Accordingly, use of this area is prohibited while the hot plug-in function is in use or the DTC is in use for the RRM or DMM function.

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Remark** The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-1 Correspondence between Address Values and Block Numbers in Flash Memory**.




(R5F10BMG)

Correspondence between the address values and block numbers in the flash memory are shown below.

**Table 3-1. Correspondence between Address Values and Block Numbers in Flash Memory (1/2)**

| Address Value | Block Number | Address Value | Block Number | Address Value | Block Number | Address Value | Block Number |
|---|---|---|---|---|---|---|---|
| 00000H to 003FFH | 00H | 08000H to 083FFH | 20H | 10000H to 103FFH | 40H | 18000H to 183FFH | 60H |
| 00400H to 007FFH | 01H | 08400H to 087FFH | 21H | 10400H to 107FFH | 41H | 18400H to 187FFH | 61H |
| 00800H to 00BFFH | 02H | 08800H to 08BFFH | 22H | 10800H to 10BFFH | 42H | 18800H to 18BFFH | 62H |
| 00C00H to 00FFFH | 03H | 08C00H to 08FFFH | 23H | 10C00H to 10FFFH | 43H | 18C00H to 18FFFH | 63H |
| 01000H to 013FFH | 04H | 09000H to 093FFH | 24H | 11000H to 113FFH | 44H | 19000H to 193FFH | 64H |
| 01400H to 017FFH | 05H | 09400H to 097FFH | 25H | 11400H to 117FFH | 45H | 19400H to 197FFH | 65H |
| 01800H to 01BFFH | 06H | 09800H to 09BFFH | 26H | 11800H to 11BFFH | 46H | 19800H to 19BFFH | 66H |
| 01C00H to 01FFFH | 07H | 09C00H to 09FFFH | 27H | 11C00H to 11FFFH | 47H | 19C00H to 19FFFH | 67H |
| 02000H to 023FFH | 08H | 0A000H to 0A3FFH | 28H | 12000H to 123FFH | 48H | 1A000H to 1A3FFH | 68H |
| 02400H to 027FFH | 09H | 0A400H to 0A7FFH | 29H | 12400H to 127FFH | 49H | 1A400H to 1A7FFH | 69H |
| 02800H to 02BFFH | 0AH | 0A800H to 0ABFFH | 2AH | 12800H to 12BFFH | 4AH | 1A800H to 1ABFFH | 6AH |
| 02C00H to 02FFFH | 0BH | 0AC00H to 0AFFFH | 2BH | 12C00H to 12FFFH | 4BH | 1AC00H to 1AFFFH | 6BH |
| 03000H to 033FFH | 0CH | 0B000H to 0B3FFH | 2CH | 13000H to 133FFH | 4CH | 1B000H to 1B3FFH | 6CH |
| 03400H to 037FFH | 0DH | 0B400H to 0B7FFH | 2DH | 13400H to 137FFH | 4DH | 1B400H to 1B7FFH | 6DH |
| 03800H to 03BFFH | 0EH | 0B800H to 0BBFFH | 2EH | 13800H to 13BFFH | 4EH | 1B800H to 1BBFFH | 6EH |
| 03C00H to 03FFFH | 0FH | 0BC00H to 0BFFFH | 2FH | 13C00H to 13FFFH | 4FH | 1BC00H to 1BFFFH | 6FH |
| 04000H to 043FFH | 10H | 0C000H to 0C3FFH | 30H | 14000H to 143FFH | 50H | 1C000H to 1C3FFH | 70H |
| 04400H to 047FFH | 11H | 0C400H to 0C7FFH | 31H | 14400H to 147FFH | 51H | 1C400H to 1C7FFH | 71H |
| 04800H to 04BFFH | 12H | 0C800H to 0CBFFH | 32H | 14800H to 14BFFH | 52H | 1C800H to 1CBFFH | 72H |
| 04C00H to 04FFFH | 13H | 0CC00H to 0CFFFH | 33H | 14C00H to 14FFFH | 53H | 1CC00H to 1CFFFH | 73H |
| 05000H to 053FFH | 14H | 0D000H to 0D3FFH | 34H | 15000H to 153FFH | 54H | 1D000H to 1D3FFH | 74H |
| 05400H to 057FFH | 15H | 0D400H to 0D7FFH | 35H | 15400H to 157FFH | 55H | 1D400H to 1D7FFH | 75H |
| 05800H to 05BFFH | 16H | 0D800H to 0DBFFH | 36H | 15800H to 15BFFH | 56H | 1D800H to 1DBFFH | 76H |
| 05C00H to 05FFFH | 17H | 0DC00H to 0DFFFH | 37H | 15C00H to 15FFFH | 57H | 1DC00H to 1DFFFH | 77H |
| 06000H to 063FFH | 18H | 0E000H to 0E3FFH | 38H | 16000H to 163FFH | 58H | 1E000H to 1E3FFH | 78H |
| 06400H to 067FFH | 19H | 0E400H to 0E7FFH | 39H | 16400H to 167FFH | 59H | 1E400H to 1E7FFH | 79H |
| 06800H to 06BFFH | 1AH | 0E800H to 0EBFFH | 3AH | 16800H to 16BFFH | 5AH | 1E800H to 1EBFFH | 7AH |
| 06C00H to 06FFFH | 1BH | 0EC00H to 0EFFFH | 3BH | 16C00H to 16FFFH | 5BH | 1EC00H to 1EFFFH | 7BH |
| 07000H to 073FFH | 1CH | 0F000H to 0F3FFH | 3CH | 17000H to 173FFH | 5CH | 1F000H to 1F3FFH | 7CH |
| 07400H to 077FFH | 1DH | 0F400H to 0F7FFH | 3DH | 17400H to 177FFH | 5DH | 1F400H to 1F7FFH | 7DH |
| 07800H to 07BFFH | 1EH | 0F800H to 0FBFFH | 3EH | 17800H to 17BFFH | 5EH | 1F800H to 1FBFFH | 7EH |
| 07C00H to 07FFFH | 1FH | 0FC00H to 0FFFFH | 3FH | 17C00H to 17FFFH | 5FH | 1FC00H to 1FFFFH | 7FH |

**Table 3-1. Correspondence between Address Values and Block Numbers in Flash Memory (2/2)**

| Address Value | Block Number | Address Value | Block Number | Address Value | Block Number | Address Value | Block Number |
|---|---|---|---|---|---|---|---|
| 20000H to 203FFH | 80H | 28000H to 283FFH | A0H | 30000H to 303FFH | C0H | 38000H to 383FFH | E0H |
| 20400H to 207FFH | 81H | 28400H to 287FFH | A1H | 30400H to 307FFH | C1H | 38400H to 387FFH | E1H |
| 20800H to 20BFFH | 82H | 28800H to 28BFFH | A2H | 30800H to 30BFFH | C2H | 38800H to 38BFFH | E2H |
| 20C00H to 20FFFH | 83H | 28C00H to 28FFFH | A3H | 30C00H to 30FFFH | C3H | 38C00H to 38FFFH | E3H |
| 21000H to 213FFH | 84H | 29000H to 293FFH | A4H | 31000H to 313FFH | C4H | 39000H to 393FFH | E4H |
| 21400H to 217FFH | 85H | 29400H to 297FFH | A5H | 31400H to 317FFH | C5H | 39400H to 397FFH | E5H |
| 21800H to 21BFFH | 86H | 29800H to 29BFFH | A6H | 31800H to 31BFFH | C6H | 39800H to 39BFFH | E6H |
| 21C00H to 21FFFH | 87H | 29C00H to 29FFFH | A7H | 31C00H to 31FFFH | C7H | 39C00H to 39FFFH | E7H |
| 22000H to 223FFH | 88H | 2A000H to 2A3FFH | A8H | 32000H to 323FFH | C8H | 3A000H to 3A3FFH | E8H |
| 22400H to 227FFH | 89H | 2A400H to 2A7FFH | A9H | 32400H to 327FFH | C9H | 3A400H to 3A7FFH | E9H |
| 22800H to 22BFFH | 8AH | 2A800H to 2ABFFH | AAH | 32800H to 32BFFH | CAH | 3A800H to 3ABFFH | EAH |
| 22C00H to 22FFFH | 8BH | 2AC00H to 2AFFFH | ABH | 32C00H to 32FFFH | CBH | 3AC00H to 3AFFFH | EBH |
| 23000H to 233FFH | 8CH | 2B000H to 2B3FFH | ACH | 33000H to 333FFH | CCH | 3B000H to 3B3FFH | ECH |
| 23400H to 237FFH | 8DH | 2B400H to 2B7FFH | ADH | 33400H to 337FFH | CDH | 3B400H to 3B7FFH | EDH |
| 23800H to 23BFFH | 8EH | 2B800H to 2BBFFH | AEH | 33800H to 33BFFH | CEH | 3B800H to 3BBFFH | EEH |
| 23C00H to 23FFFH | 8FH | 2BC00H to 2BFFFH | AFH | 33C00H to 33FFFH | CFH | 3BC00H to 3BFFFH | EFH |
| 24000H to 243FFH | 90H | 2C000H to 2C3FFH | B0H | 34000H to 343FFH | D0H | 3C000H to 3C3FFH | F0H |
| 24400H to 247FFH | 91H | 2C400H to 2C7FFH | B1H | 34400H to 347FFH | D1H | 3C400H to 3C7FFH | F1H |
| 24800H to 24BFFH | 92H | 2C800H to 2CBFFH | B2H | 34800H to 34BFFH | D2H | 3C800H to 3CBFFH | F2H |
| 24C00H to 24FFFH | 93H | 2CC00H to 2CFFFH | B3H | 34C00H to 34FFFH | D3H | 3CC00H to 3CFFFH | F3H |
| 25000H to 253FFH | 94H | 2D000H to 2D3FFH | B4H | 35000H to 353FFH | D4H | 3D000H to 3D3FFH | F4H |
| 25400H to 257FFH | 95H | 2D400H to 2D7FFH | B5H | 35400H to 357FFH | D5H | 3D400H to 3D7FFH | F5H |
| 25800H to 25BFFH | 96H | 2D800H to 2DBFFH | B6H | 35800H to 35BFFH | D6H | 3D800H to 3DBFFH | F6H |
| 25C00H to 25FFFH | 97H | 2DC00H to 2DFFFH | B7H | 35C00H to 35FFFH | D7H | 3DC00H to 3DFFFH | F7H |
| 26000H to 263FFH | 98H | 2E000H to 2E3FFH | B8H | 36000H to 363FFH | D8H | 3E000H to 3E3FFH | F8H |
| 26400H to 267FFH | 99H | 2E400H to 2E7FFH | B9H | 36400H to 367FFH | D9H | 3E400H to 3E7FFH | F9H |
| 26800H to 26BFFH | 9AH | 2E800H to 2EBFFH | BAH | 36800H to 36BFFH | DAH | 3E800H to 3EBFFH | FAH |
| 26C00H to 26FFFH | 9BH | 2EC00H to 2EFFFH | BBH | 36C00H to 36FFFH | DBH | 3EC00H to 3EFFFH | FBH |
| 27000H to 273FFH | 9CH | 2F000H to 2F3FFH | BCH | 37000H to 373FFH | DCH | 3F000H to 3F3FFH | FCH |
| 27400H to 277FFH | 9DH | 2F400H to 2F7FFH | BDH | 37400H to 377FFH | DDH | 3F400H to 3F7FFH | FDH |
| 27800H to 27BFFH | 9EH | 2F800H to 2FBFFH | BEH | 37800H to 37BFFH | DEH | 3F800H to 3FBFFH | FEH |
| 27C00H to 27FFFH | 9FH | 2FC00H to 2FFFFH | BFH | 37C00H to 37FFFH | DFH | 3FC00H to 3FFFFH | FFH |

### 3.1.1 Internal program memory space

The internal program memory space stores the program and table data.

The RL78/F13 and RL78/F14 products incorporate internal ROM (flash memory), as shown below.

**Table 3-2. Internal ROM Capacity**

| Part Number | Internal ROM | |
|---|---|---|
| | Structure | Capacity |
| R5F10AmA (m = 6, A, B, G) | Flash memory | 16 Kbytes (00000H-03FFFH) |
| R5F10AmC (m = 6, A, B, G, L) | | 32 Kbytes (00000H-07FFFH) |
| R5F10BmC (m = A, B, G, L) | | |
| R5F10AmD (m = 6, A, B, G, L) | | 48 Kbytes (00000H-0BFFFH) |
| R5F10BmD (m = A, B, G, L) | | |
| R5F10PmD (m = A, B,G) | | |
| R5F10AmE (m = 6, A, B, G, L, M) | | 64 Kbytes (00000H-0FFFFH) |
| R5F10BmE (m = A, B, G, L, M) | | |
| R5F10PmE (m = A, B, G, L, M, P) | | |
| R5F10AmF (m = G, L, M) | | 96 Kbytes (00000H-17FFFH) |
| R5F10BmF (m = A, B, G, L, M) | | |
| R5F10PmF (m = G, L, M, P) | | |
| R5F10AmG (m = G, L, M) | | 128 Kbytes (00000H-1FFFFH) |
| R5F10BmG (m = A, B, G, L, M) | | |
| R5F10PmG (m = G, L, M, P) | | |
| R5F10PmH (m = G, L, M, P) | | 192 Kbytes (00000H-2FFFFH) |
| R5F10PmJ (m = G, L, M, P) | | 256 Kbytes (00000H-3FFFFH) |

The internal program memory space is divided into the following areas.

**(1) Vector table area**

The 128-byte area 00000H to 0007FH is reserved as a vector table area. The program start addresses for branch upon reset or generation of each interrupt request are stored in the vector table area. Furthermore, the interrupt jump address is a 64 K address of 00000H to 0FFFFH, because the vector code is assumed to be 2 bytes.

Of the 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

To use the boot swap function, set a vector table also at 02000H to 0207FH.

Table 3-3 lists the vector table. "√" indicates an interrupt source which is supported. "–" indicates an interrupt source which is not supported.

**Table 3-3. Vector Table (1/2)**

| Vector Table Address | Interrupt Source | 100-pin | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
|---|---|---|---|---|---|---|---|---|
| 0000H | RESET, POR, LVD, WDT, TRAP, IAW, CLM | √ | √ | √ | √ | √ | √ | √ |
| 0004H | INTWDTI | √ | √ | √ | √ | √ | √ | √ |
| 0006H | INTLVI | √ | √ | √ | √ | √ | √ | √ |
| 0008H | INTP0 | √ | √ | √ | √ | √ | √ | √ |
| 000AH | INTP1 | √ | √ | √ | √ | √ | √ | √ |
| 000CH | INTP2 | √ | √ | √ | √ | √ | √ | √ |
| 000EH | INTP3 | √ | √ | √ | √ | √ | √ | √ |
| 0010H | INTP4/INTSPM | √ | √ | √ | √ | √ | √ | √ |
| 0012H | INTP5/INTCMP0 | √ | √ | √ | √ | √ | √ | – |
| 0014H | INTP13 | √ | √ | – | – | – | – | – |
| | INTCLM | √ | √ | √ | √ | √ | √ | √ |
| 0016H | INTST0 | √ | √ | √ | √ | √ | √ | √ |
| | INTCSI00 | √ | √ | √ | √ | √ | √ | √ |
| | INTIIC00 | √ | √ | √ | √ | √ | √ | √ |
| 0018H | INTSR0 | √ | √ | √ | √ | √ | √ | √ |
| | INTCSI01 | √ | √ | √ | √ | √ | √ | √ |
| | INTIIC01 | √ | √ | √ | √ | √ | √ | √ |
| 001AH | INTTRD0 | √ | √ | √ | √ | √ | √ | √ |
| 001CH | INTTRD1 | √ | √ | √ | √ | √ | √ | √ |
| 001EH | INTTRJ0 | √ | √ | √ | √ | √ | √ | √ |
| 0020H | INTRAM | √ | √ | √ | √ | √ | √ | √ |
| 0022H | INTLIN0TRM | √ | √ | √ | √ | √ | √ | √ |
| 0024H | INTLIN0RVC | √ | √ | √ | √ | √ | √ | √ |
| 0026H | INTLIN0STA/INTLIN0 | √ | √ | √ | √ | √ | √ | √ |
| 0028H | INTIICA0 | √ | √ | √ | √ | √ | – | – |
| 002AH | INTP8 | √ | √ | √ | √ | – | – | – |
| | INTRTC | √ | √ | √ | √ | √ | √ | √ |
| 002CH | INTTM00 | √ | √ | √ | √ | √ | √ | √ |
| 002EH | INTTM01 | √ | √ | √ | √ | √ | √ | √ |
| 0030H | INTTM02 | √ | √ | √ | √ | √ | √ | √ |
| 0032H | INTTM03 | √ | √ | √ | √ | √ | √ | √ |
| 0034H | INTAD | √ | √ | √ | √ | √ | √ | √ |
| 0036H | INTP6 | √ | √ | √ | √ | – | – | – |
| | INTTM11H | √ | √ | √ | √ | √ | √ | – |
| 0038H | INTP7 | √ | √ | √ | √ | – | – | – |
| | INTTM13H | √ | √ | √ | √ | √ | √ | – |
| 003AH | INTP9 | √ | √ | √ | √ | – | – | – |
| | INTTM01H | √ | √ | √ | √ | √ | √ | √ |
| 003CH | INTP10 | √ | √ | √ | – | – | – | – |
| | INTTM03H | √ | √ | √ | √ | √ | √ | √ |
| 003EH | INTST1 | √ | √ | √ | √ | √ | √ | – |
| | INTCSI10 | √ | √ | √ | √ | √ | √ | – |
| | INTIIC10 | √ | √ | √ | √ | √ | √ | – |

**Table 3-3. Vector Table (2/2)**

| Vector Table Address | Interrupt Source | 100-pin | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
|---|---|---|---|---|---|---|---|---|
| 0040H | INTSR1 | √ | √ | √ | √ | √ | √ | – |
| | INTCSI11 | √ | √ | √ | √ | – | – | – |
| | INTIIC11 | √ | √ | √ | √ | – | – | – |
| 0042H | INTTM04 | √ | √ | √ | √ | √ | √ | √ |
| 0044H | INTTM05 | √ | √ | √ | √ | √ | √ | √ |
| 0046H | INTTM06 | √ | √ | √ | √ | √ | √ | √ |
| 0048H | INTTM07 | √ | √ | √ | √ | √ | √ | √ |
| 004AH | INTP11 | √ | √ | √ | – | – | – | – |
| | INTLIN0WUP | √ | √ | √ | √ | √ | √ | √ |
| 004CH | INTKR | √ | √ | √ | √ | √ | √ | √ |
| 004EH | INTCAN0ERR | √ | √ | √ | √ | √ | √ | – |
| 0050H | INTCAN0WUP | √ | √ | √ | √ | √ | √ | – |
| 0052H | INTCAN0CFR | √ | √ | √ | √ | √ | √ | – |
| 0054H | INTCAN0TRM | √ | √ | √ | √ | √ | √ | – |
| 0056H | INTCANGRFR | √ | √ | √ | √ | √ | √ | – |
| 0058H | INTCANGERR | √ | √ | √ | √ | √ | √ | – |
| 005AH | INTTM10 | √ | √ | √ | √ | √ | √ | – |
| 005CH | INTTM11 | √ | √ | √ | √ | √ | √ | – |
| 005EH | INTTM12 | √ | √ | √ | √ | √ | √ | – |
| 0060H | INTTM13 | √ | √ | √ | √ | √ | √ | – |
| 0062H | INTFL | √ | √ | √ | √ | √ | √ | √ |
| 0064H | INTP12 | √ | √ | √ | – | – | – | – |
| | INTLIN1WUP | √ | √ | √ | √ | – | – | – |
| 0066H | INTLIN1TRM | √ | √ | √ | √ | – | – | – |
| 0068H | INTLIN1RVC | √ | √ | √ | √ | – | – | – |
| 006AH | INTLIN1STA/INTLIN1 | √ | √ | √ | √ | – | – | – |
| 006CH | INTTM14 | √ | √ | √ | √ | – | – | – |
| 006EH | INTTM15 | √ | √ | √ | √ | – | – | – |
| 0070H | INTTM16 | √ | √ | √ | √ | – | – | – |
| 0072H | INTTM17 | √ | √ | √ | √ | – | – | – |
| 007EH | BRK | √ | √ | √ | √ | √ | √ | √ |

**(2) CALLT instruction table area**

The 64-byte area 00080H to 000BFH can store the subroutine entry address of a 2-byte call instruction (CALLT). Set the subroutine entry address to a value in a range of 00000H to 0FFFFH (because an address code is 2 bytes).

To use the boot swap function, set a CALLT instruction table also at 02080H to 020BFH.

**(3) Option byte area**

A 4-byte area of 000C0H to 000C3H can be used as an option byte area. Set the option byte at 020C0H to 020C3H when the boot swap is used. For details, see **CHAPTER 29 OPTION BYTE**.

**(4) On-chip debug security ID setting area**

A 10-byte area of 000C4H to 000CDH and 020C4H to 020CDH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 000C4H to 000CDH when the boot swap is not used and at 000C4H to 000CDH and at 020C4H to 020CDH when the boot swap is used. For details, see **CHAPTER 31 ON-CHIP DEBUG FUNCTION**.

### 3.1.2 Mirror area

The RL78/F13 and RL78/F14 mirror the code flash area of 00000H to 0FFFFH, to F0000H to FFFFFH. The products with 96 KB or more flash memory mirror the code flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, to F0000H to FFFFFH (the code flash area to be mirrored is set by the processor mode control register (PMC)).

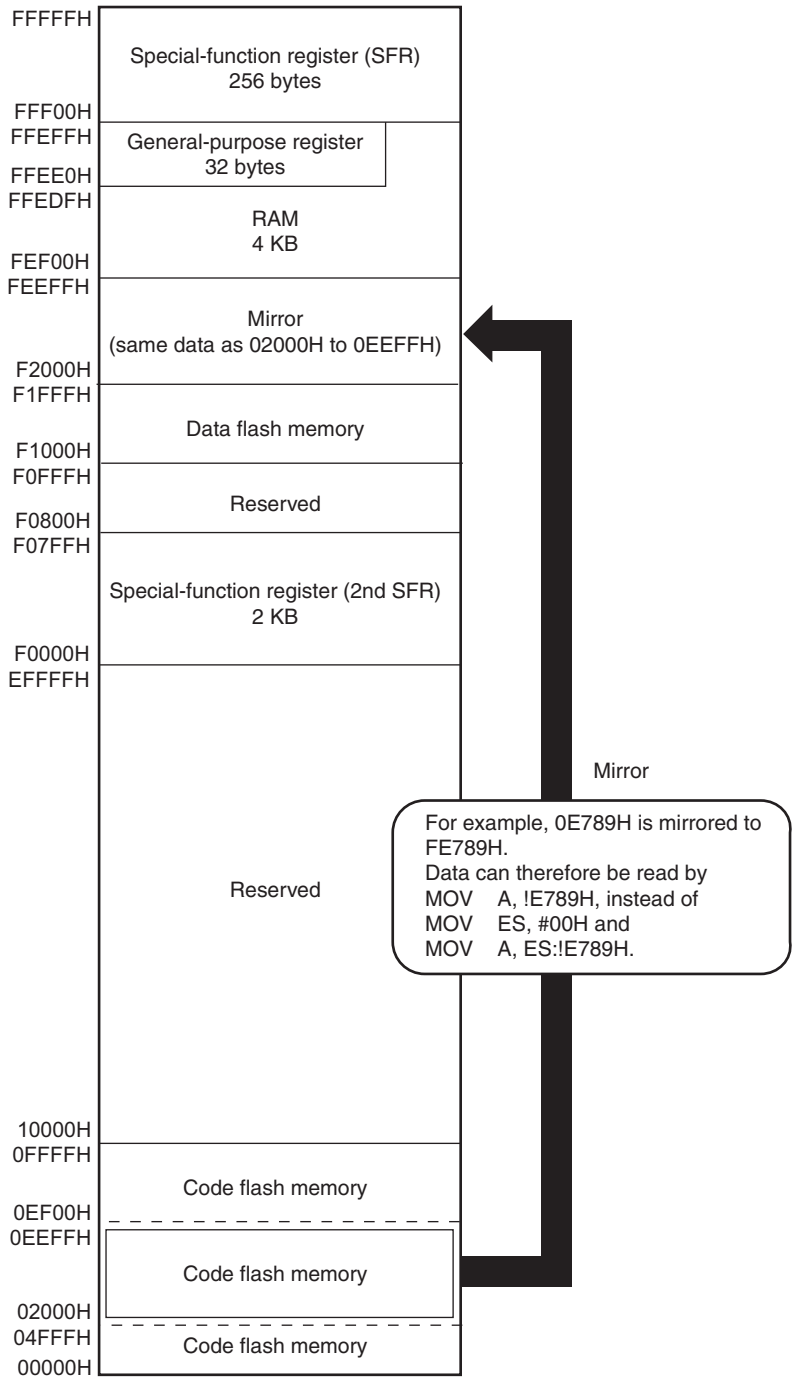
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the SFR, extended SFR, RAM, data flash memory, and use prohibited areas.

See **3.1 Memory Space** for the mirror area of each product.

The mirror area can only be read and no instruction can be fetched from this area.

The following show examples.

**Example R5F10AmE (m = 6, A, B, G, L) (Flash memory: 64 KB, RAM: 4 KB)**



The PMC register is described below.

- **Processor mode control register (PMC)**

This register sets the flash memory space for mirroring to area from F0000H to FFFFFH.

The PMC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

**Figure 3-18. Format of Configuration of Processor Mode Control Register (PMC)**

Address: FFFFEH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PMC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | MAA |

| MAA | Selection of flash memory space for mirroring to area from F0000H to FFFFFH |
|---|---|
| 0 | 00000H to 0FFFFH is mirrored to F0000H to FFFFFH |
| 1 | 10000H to 1FFFFH is mirrored to F0000H to FFFFFH Note |

**Note** This setting is prohibited in products with 64 KB or less flash memory

**Cautions 1. In products with 64 KB or less flash memory, be sure to clear bit 0 (MAA) of this register to 0 (default value).**

**2. Set the PMC register only once during the initial settings prior to operating the data transfer controller (DTC). Rewriting the PMC register other than during the initial settings is prohibited.**

**3. After setting the PMC register, wait for at least one instruction and access the mirror area.**

### 3.1.3 Internal data memory space

The RL78/F13 and RL78/F14 products incorporate the following RAMs.

**Table 3-4. Internal RAM Capacity**

| Part Number | Internal RAM |
|---|---|
| R5F10AmA (m = 6, A, B, G) | 1 Kbyte (FFB00H to FFEFFH) |
| R5F10AmC (m = 6, A, B, G, L) | 2 Kbytes (FF700H to FFEFFH) |
| R5F10BmC (m = A, B, G, L) | |
| R5F10AmD (m = 6, A, B, G, L) | 3 Kbytes (FF300H to FFEFFH) |
| R5F10BmD (m = A, B, G, L) | |
| R5F10PmD (m = A, B, G) | 4 Kbytes (FEF00H to FFEFFH) |
| R5F10AmE (m = 6, A, B, G, L, M) | |
| R5F10BmE (m = A, B, G, L, M) | |
| R5F10PmE (m = A, B, G, L, M, P) | 6 Kbytes (FE700H to FFEFFH) |
| R5F10AmF (m = G, L, M) | |
| R5F10BmF (m = A, B, G, L, M) | |
| R5F10PmF (m = G, L, M, P) | 8 Kbytes (FDF00H to FFEFFH) |
| R5F10AmG (m = G, L, M) | |
| R5F10BmG (m = A, B, G, L, M) | |
| R5F10PmG (m = G, L, M, P) | 10 Kbytes (FD700H to FFEFFH) |
| R5F10PmH (m = G, L, M, P) | 16 Kbytes (FBF00H to FFEFFH) |
| R5F10PmJ (m = G, L, M, P) | 20 Kbytes (FAF00H to FFEFFH) |

The internal RAM can be used as a data area and a program area where instructions are written and executed. Four general-purpose register banks consisting of eight 8-bit registers per bank are assigned to the 32-byte area of FFEE0H to FFEFFH of the internal RAM area. However, instructions cannot be executed by using the general-purpose registers.

The internal RAM is used as a stack memory.

**Cautions 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.**

**2. Do not allocate RAM addresses which are used as stack area, data buffers used by the libraries, branch destinations for vectored interrupt servicing, or DTC transfer destinations/transfer sources to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.**

**3. Using the listed RAM areas in the corresponding products below is prohibited since the respective libraries use them in self-programming and overwriting of the data flash memory. However, the area to which this prohibition applies may vary with the version of the library. For details, refer to the manuals for the individual libraries.**

**R5F10AmE (m = 6, A, B, G, L): FEF00H to FF2FFH**
**R5F10PmF (m = G, L, M): FDF00H to FE2FFH**
**R5F10AmG (m = G, L, M), R5F10BnG (n = A, B, G, L, M): FDF00H to FE2FFH**
**R5F10PmJ (m = G, L, M, P): FAF00H to FB2FFH**

### 3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FFF00H to FFFFFH (see **Table 3-5 SFR List** in **3.2.4 Special function registers (SFRs)**).

**Caution Do not access addresses to which SFRs are not assigned.**

### 3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area

On-chip peripheral hardware special function registers (2nd SFRs) are allocated in the area F0000H to F07FFH (see **Table 3-6 Extended SFR (2nd SFR) List** in **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**).

SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

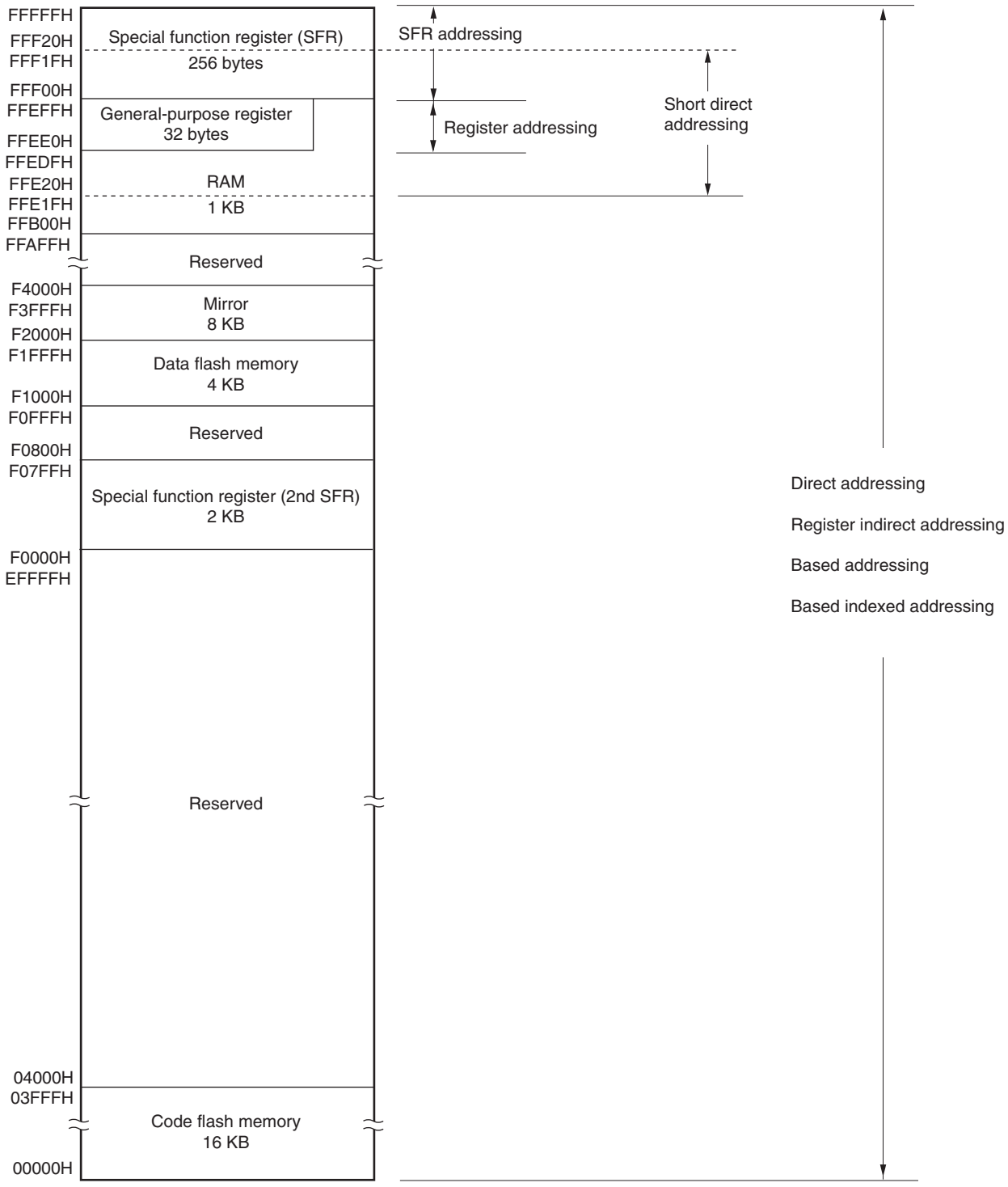
**Caution Do not access addresses to which extended SFRs are not assigned.**

### 3.1.6 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

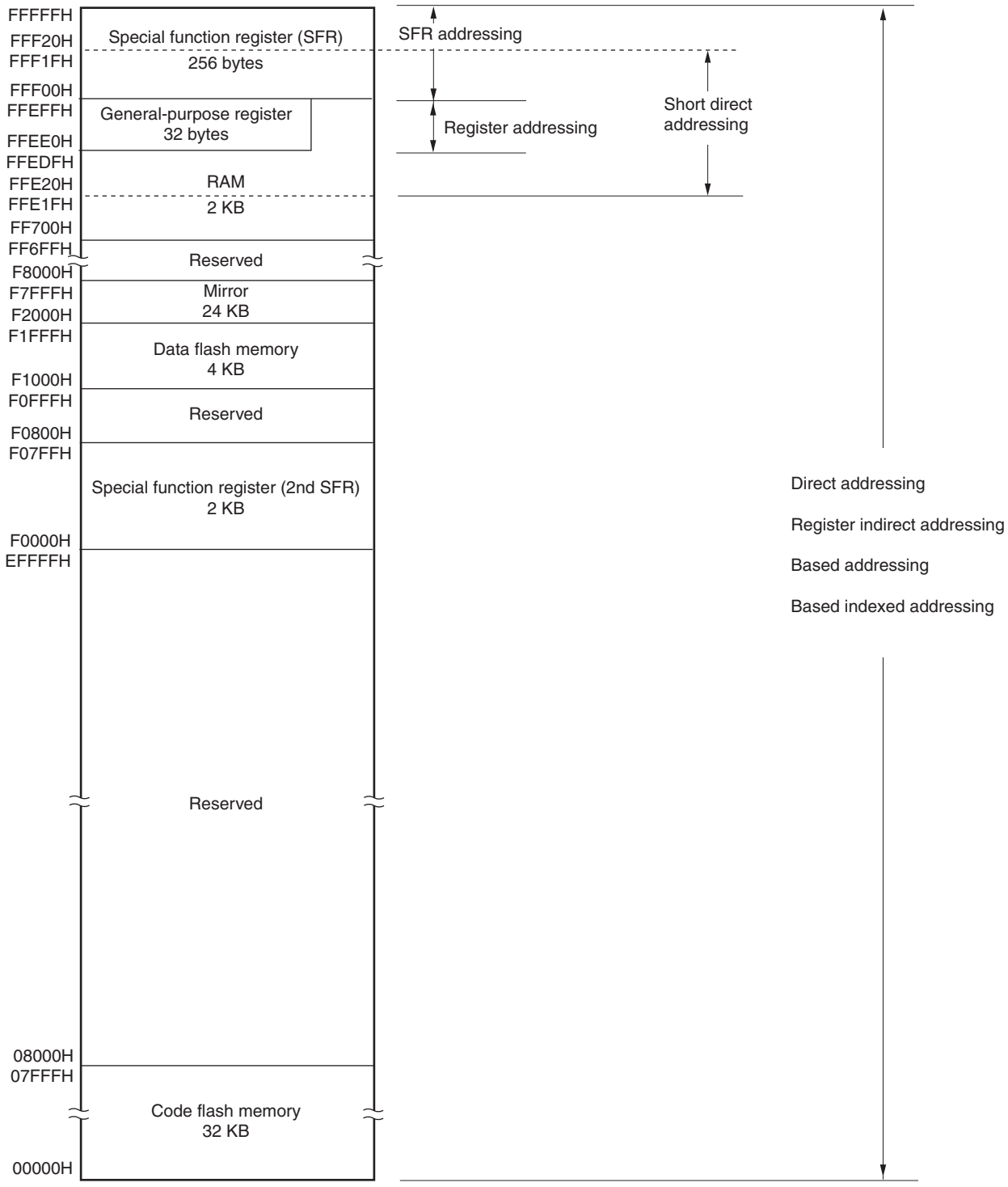
Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the RL78/F13 and RL78/F14, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figures 3-19 to 3-35 show correspondence between data memory and addressing. For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

**Figure 3-19. Correspondence between Data Memory and Addressing (R5F10AmA (m = 6, A, B, G))**



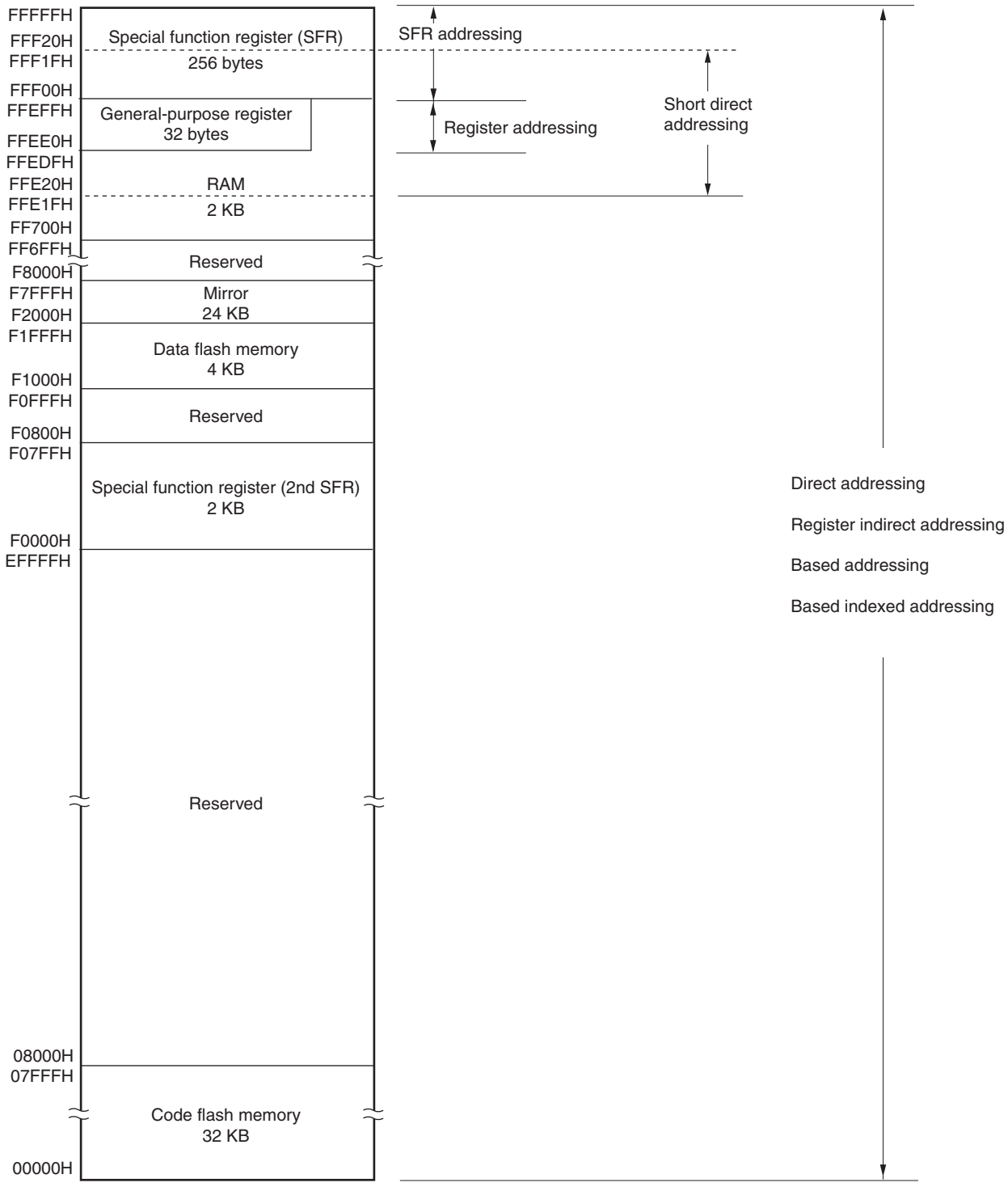

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-20. Correspondence between Data Memory and Addressing (R5F10AmC (m = 6, A, B, G, L))**



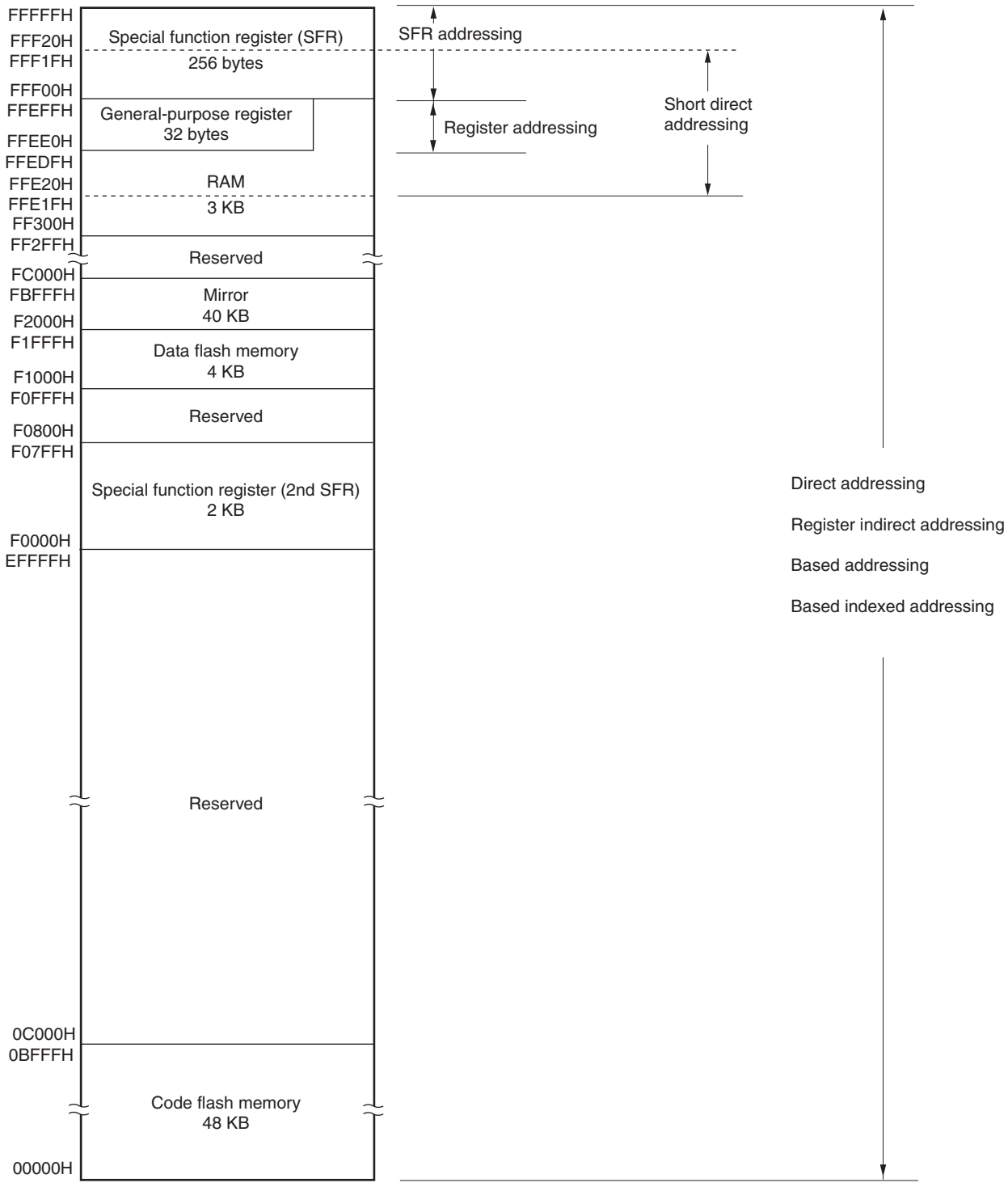

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-21. Correspondence between Data Memory and Addressing (R5F10BmC (m = A, B, G, L))**



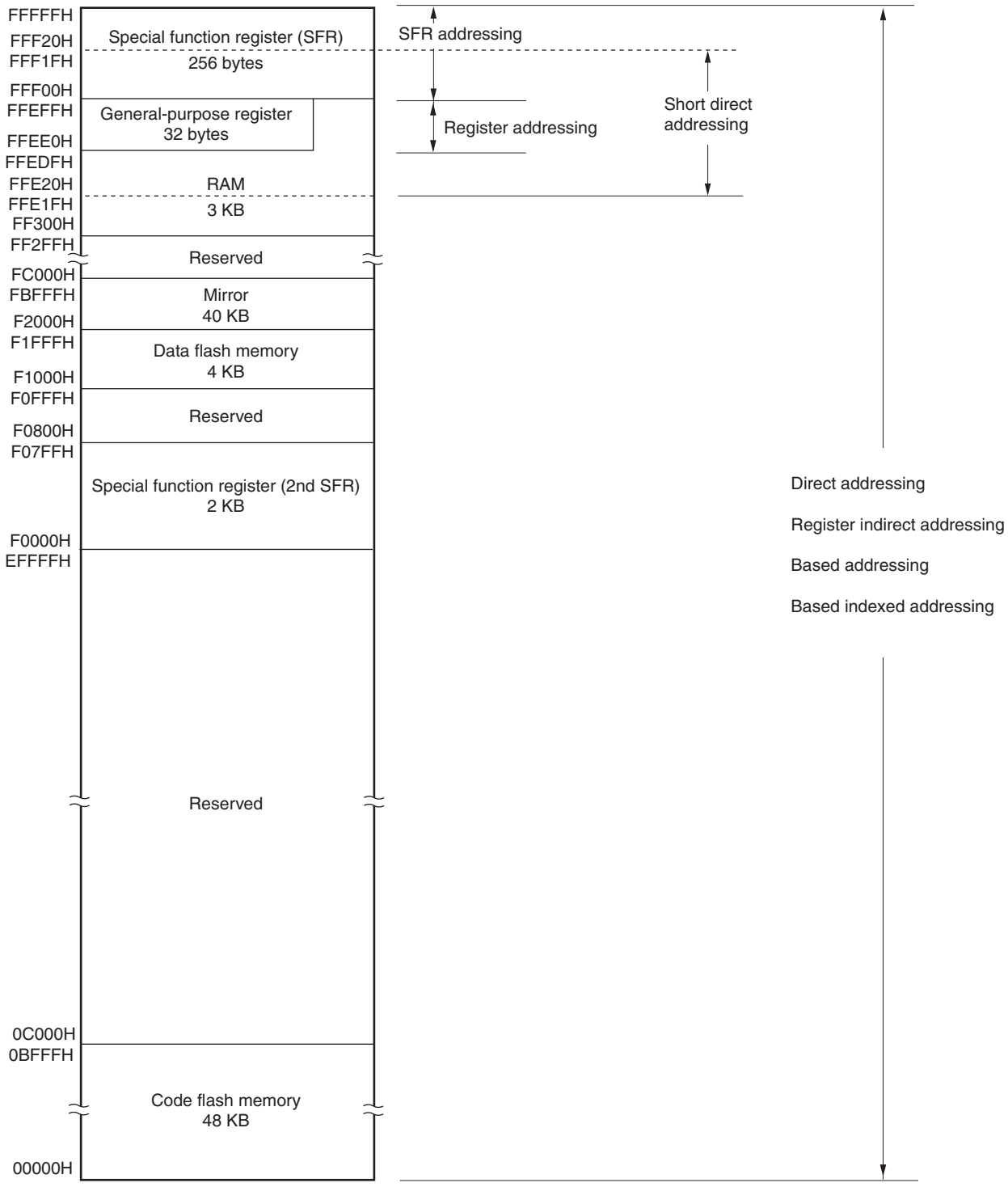

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-22. Correspondence between Data Memory and Addressing (R5F10AmD (m = 6, A, B, G, L))**



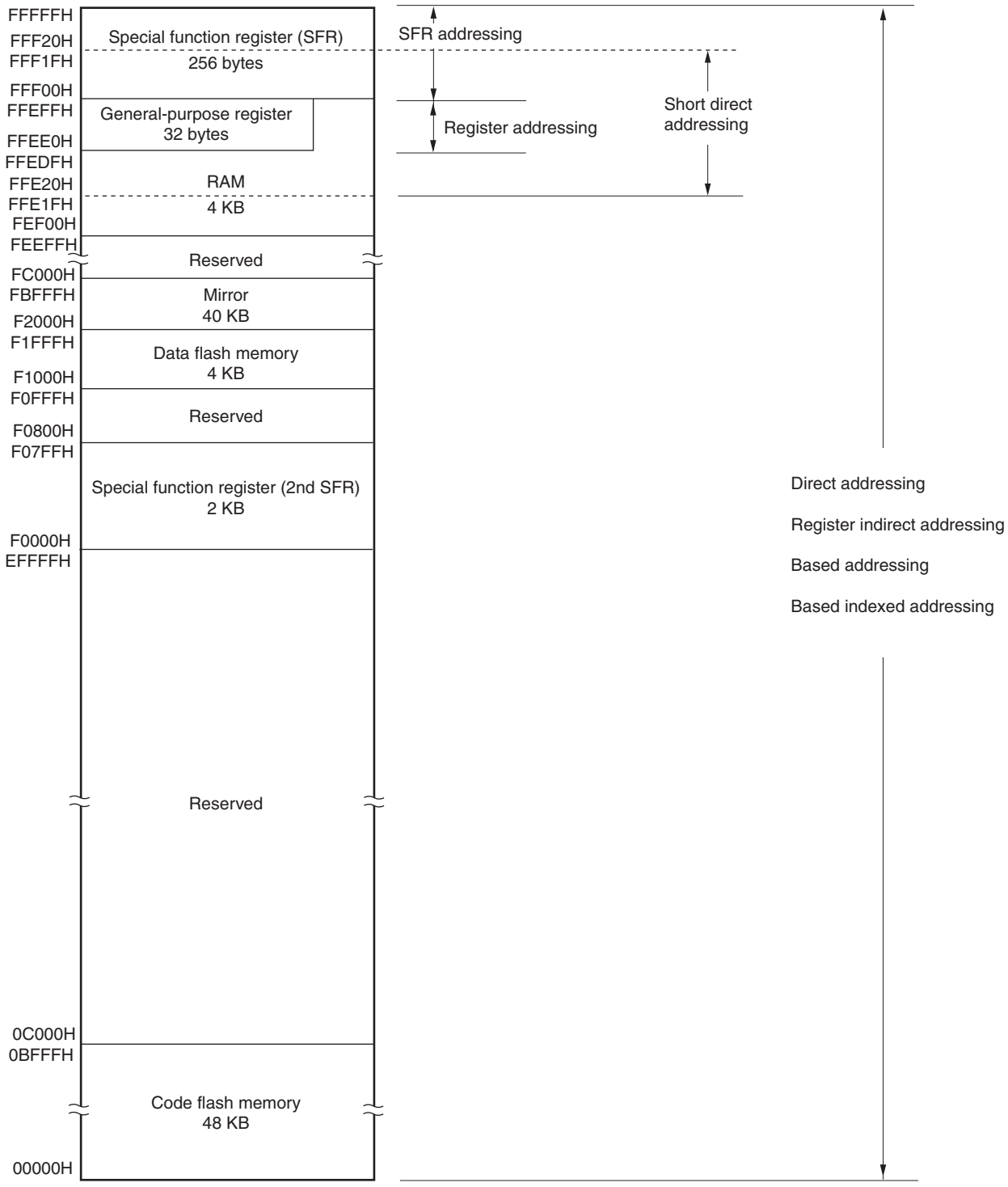

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-23. Correspondence between Data Memory and Addressing (R5F10BmD (m = A, B, G, L))**



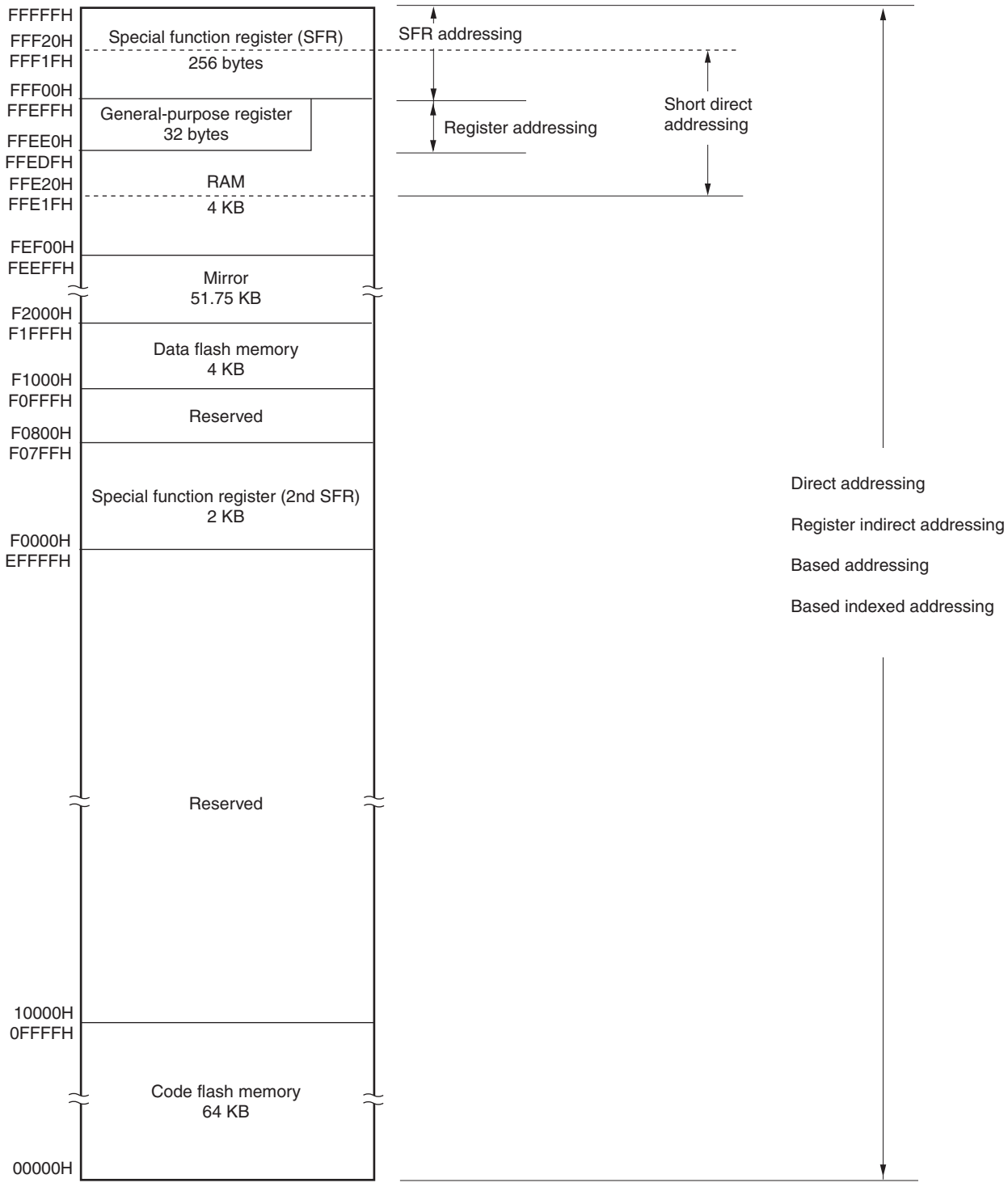

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-24. Correspondence between Data Memory and Addressing (R5F10PmD (m = A, B, G))**



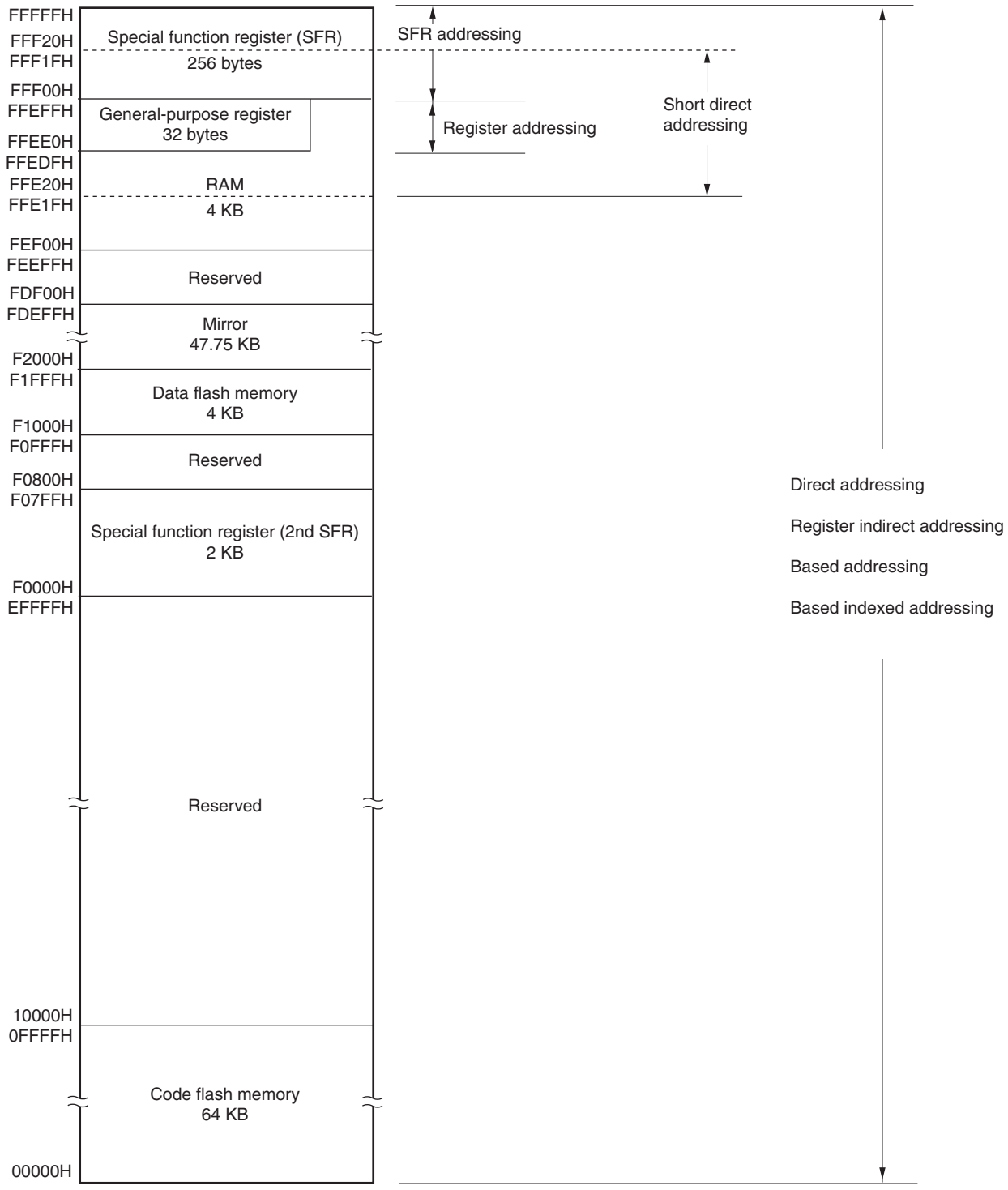

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-25. Correspondence between Data Memory and Addressing (R5F10AmE (m = 6, A, B, G, L))**



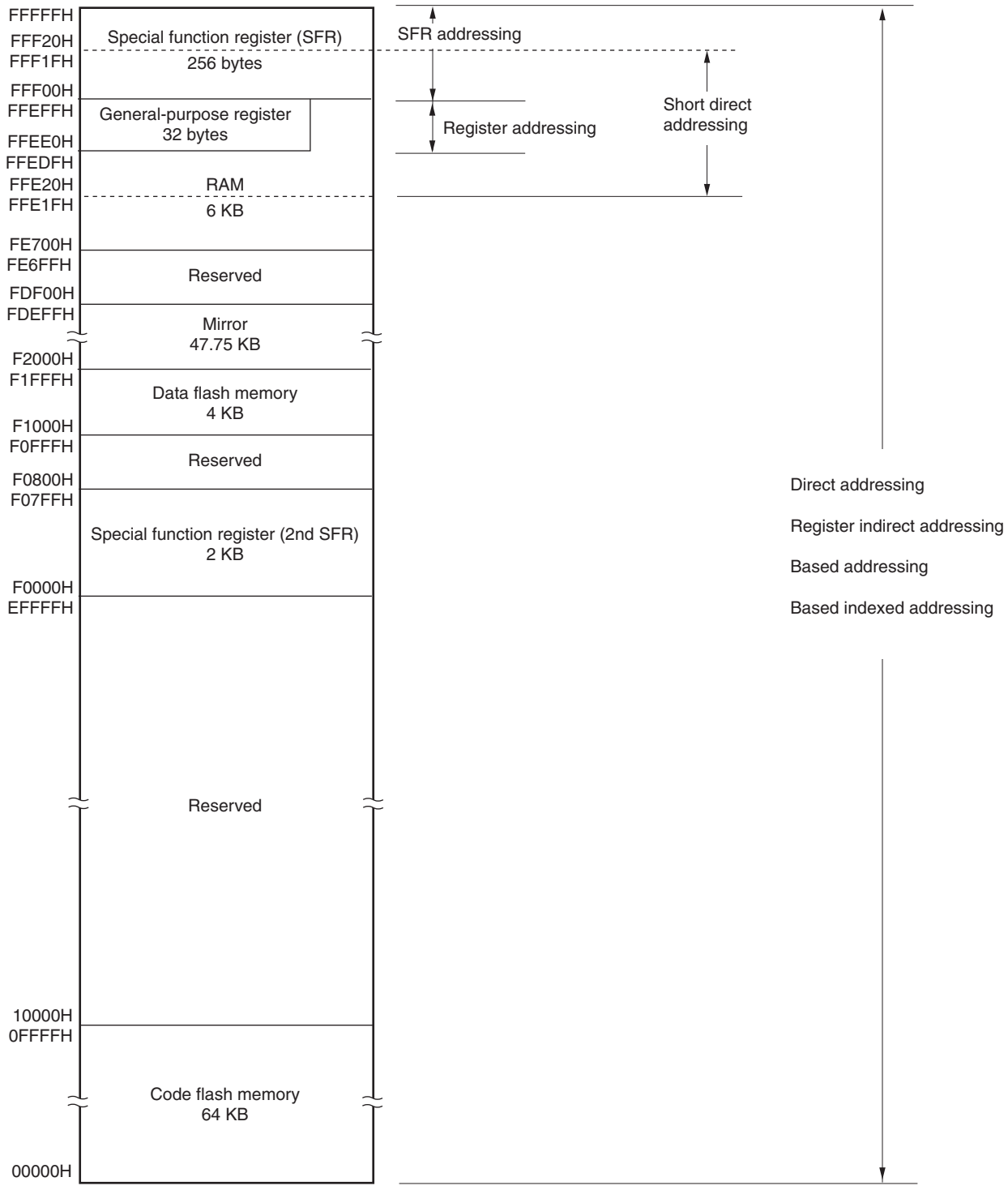

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-26. Correspondence between Data Memory and Addressing (R5F10AmE, R5F10BmE (m = A, B, G, L, M))**



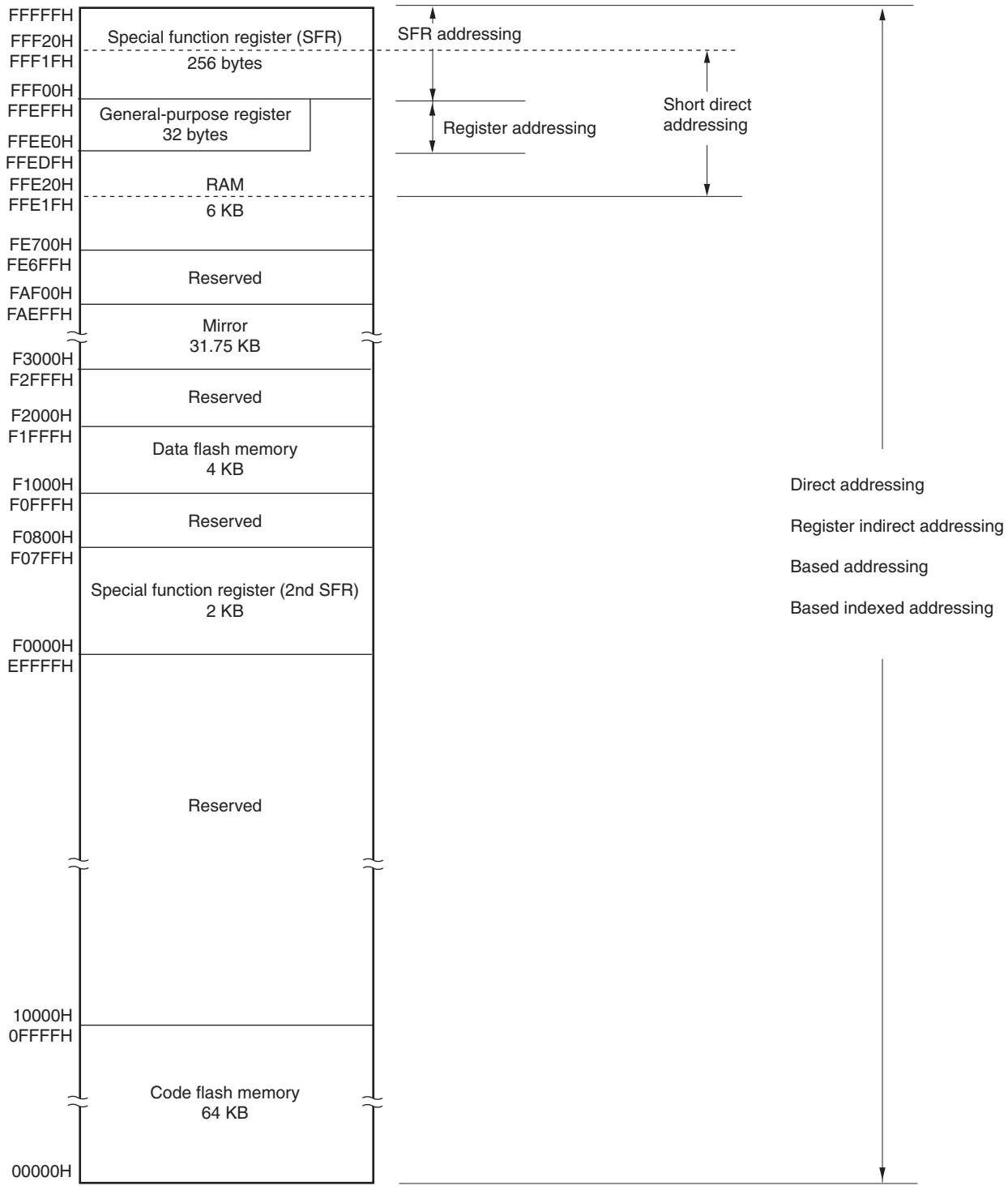

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-27. Correspondence between Data Memory and Addressing (R5F10PmE (m = A, B, G, L, M))**



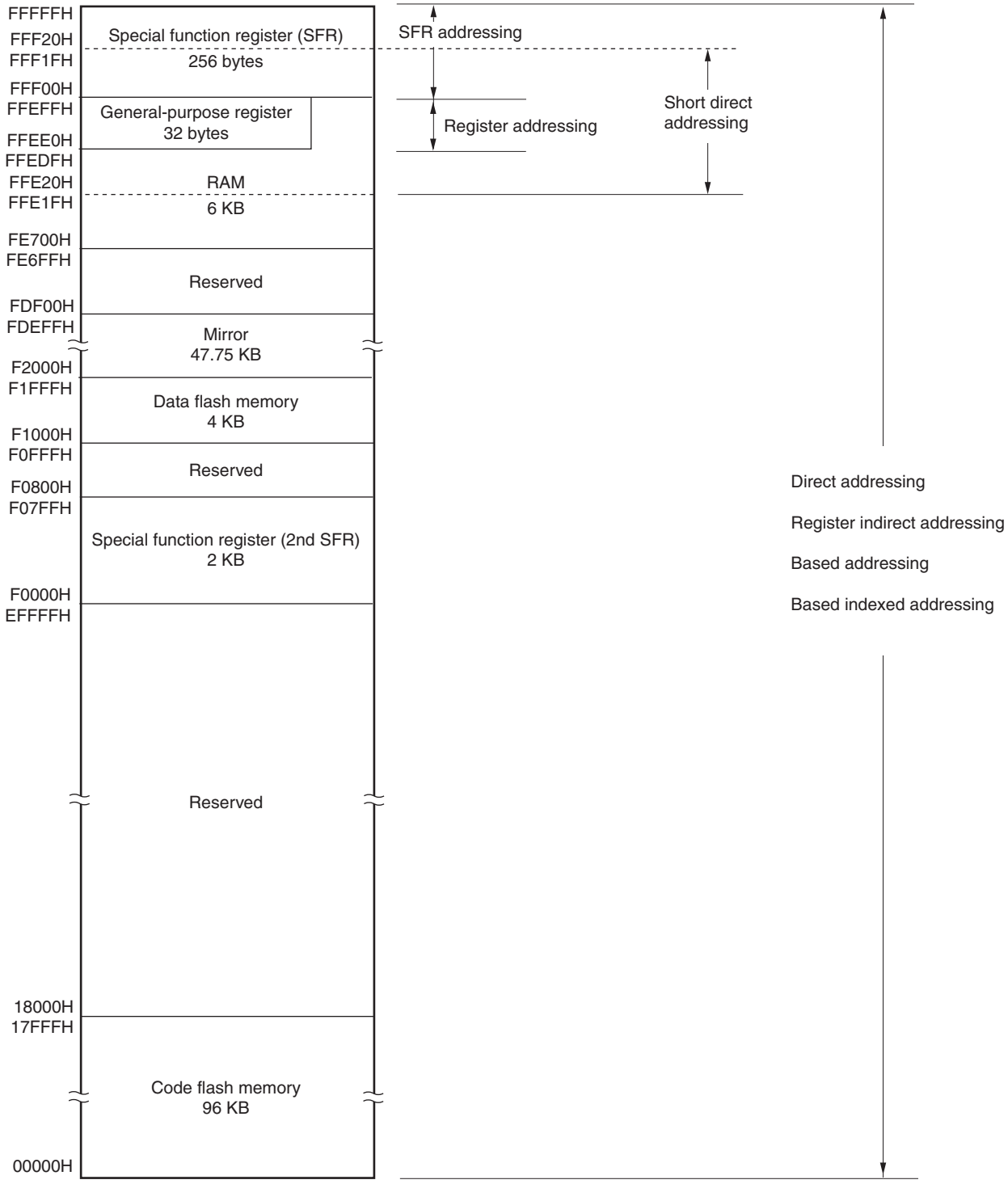

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-28. Correspondence between Data Memory and Addressing (R5F10PPE)**



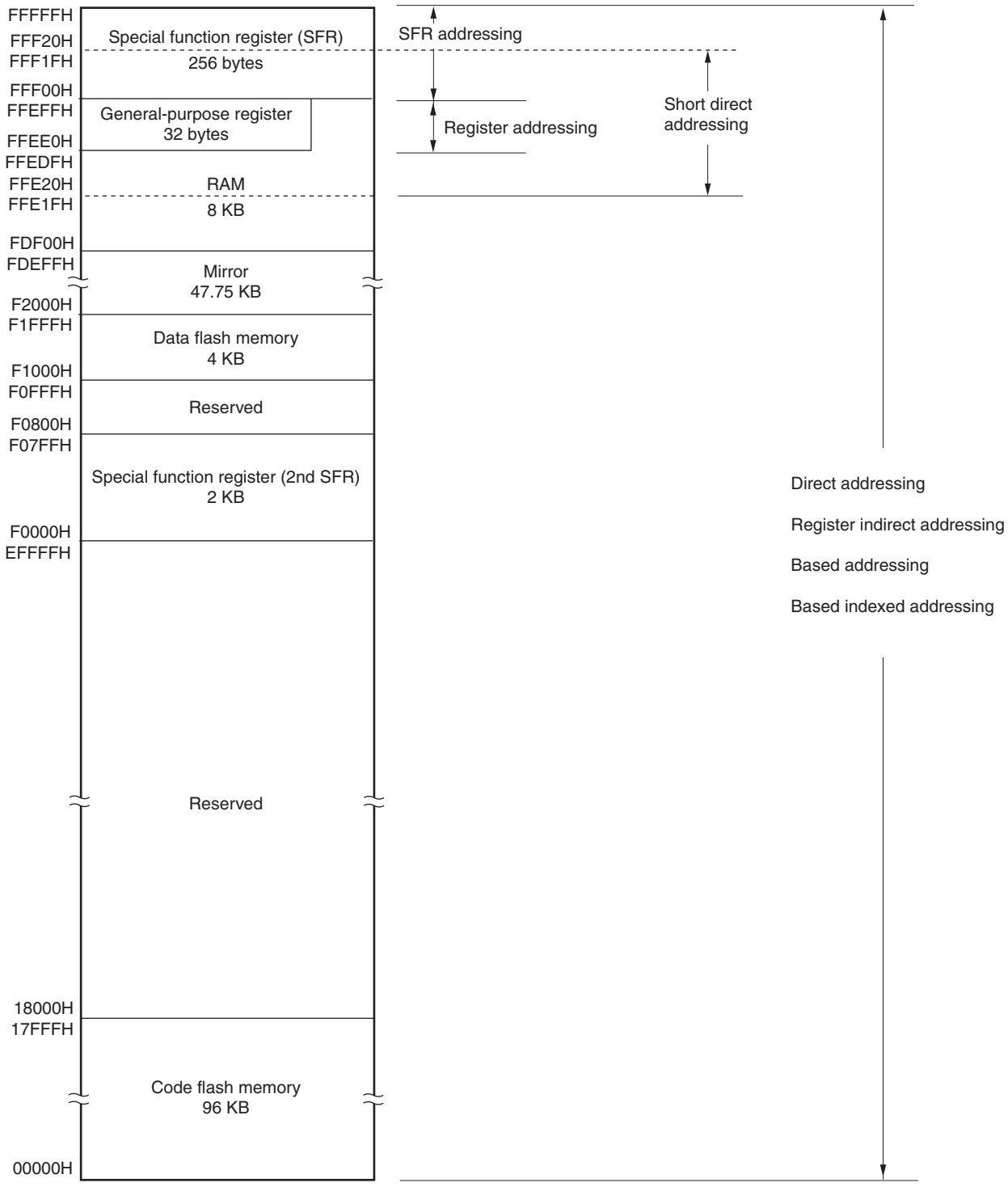

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-29. Correspondence between Data Memory and Addressing (R5F10AmF (m = G, L, M), R5F10BnF (n = A, B, G, L, M))**



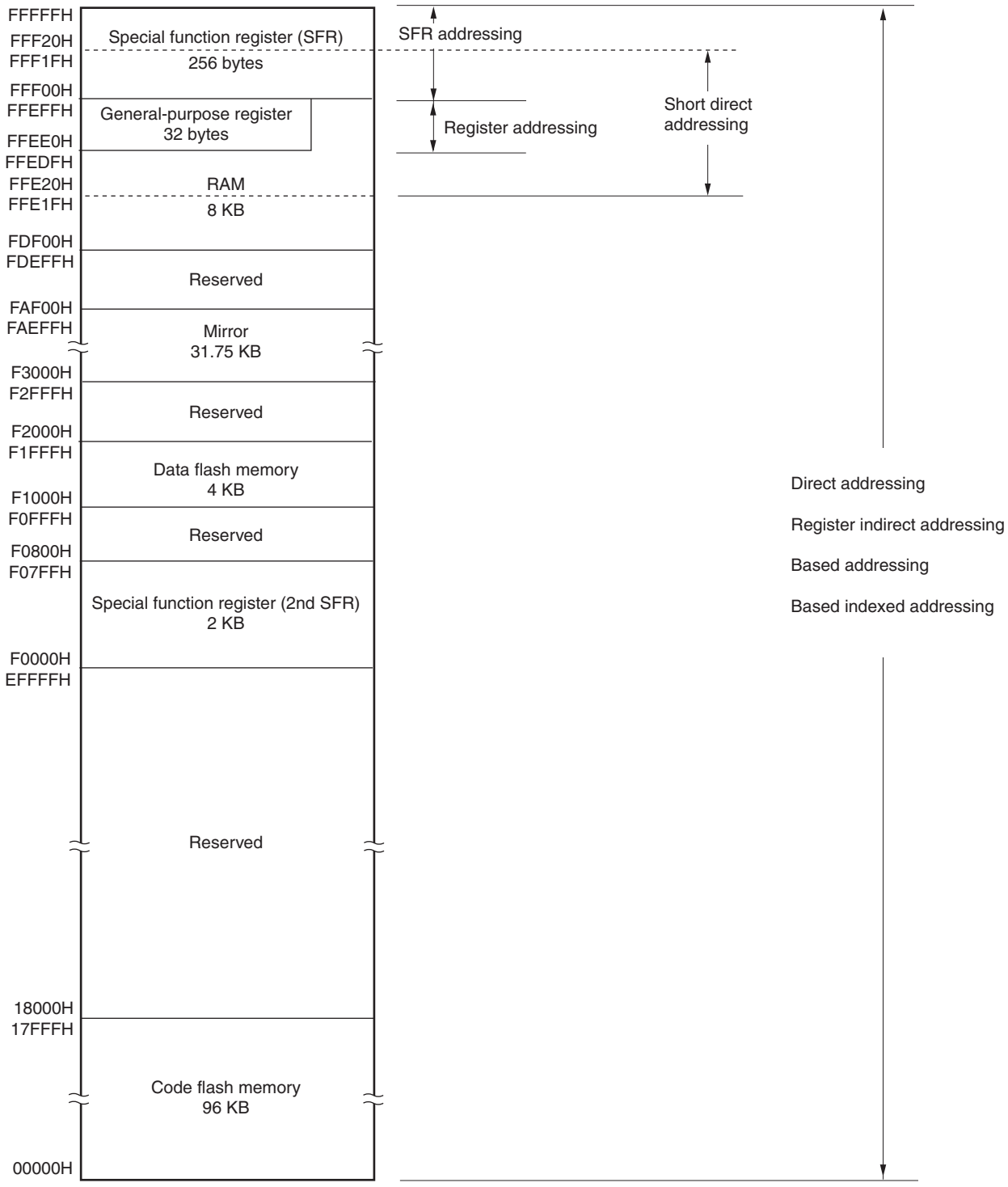

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-30. Correspondence between Data Memory and Addressing (R5F10PmF (m = G, L, M))**



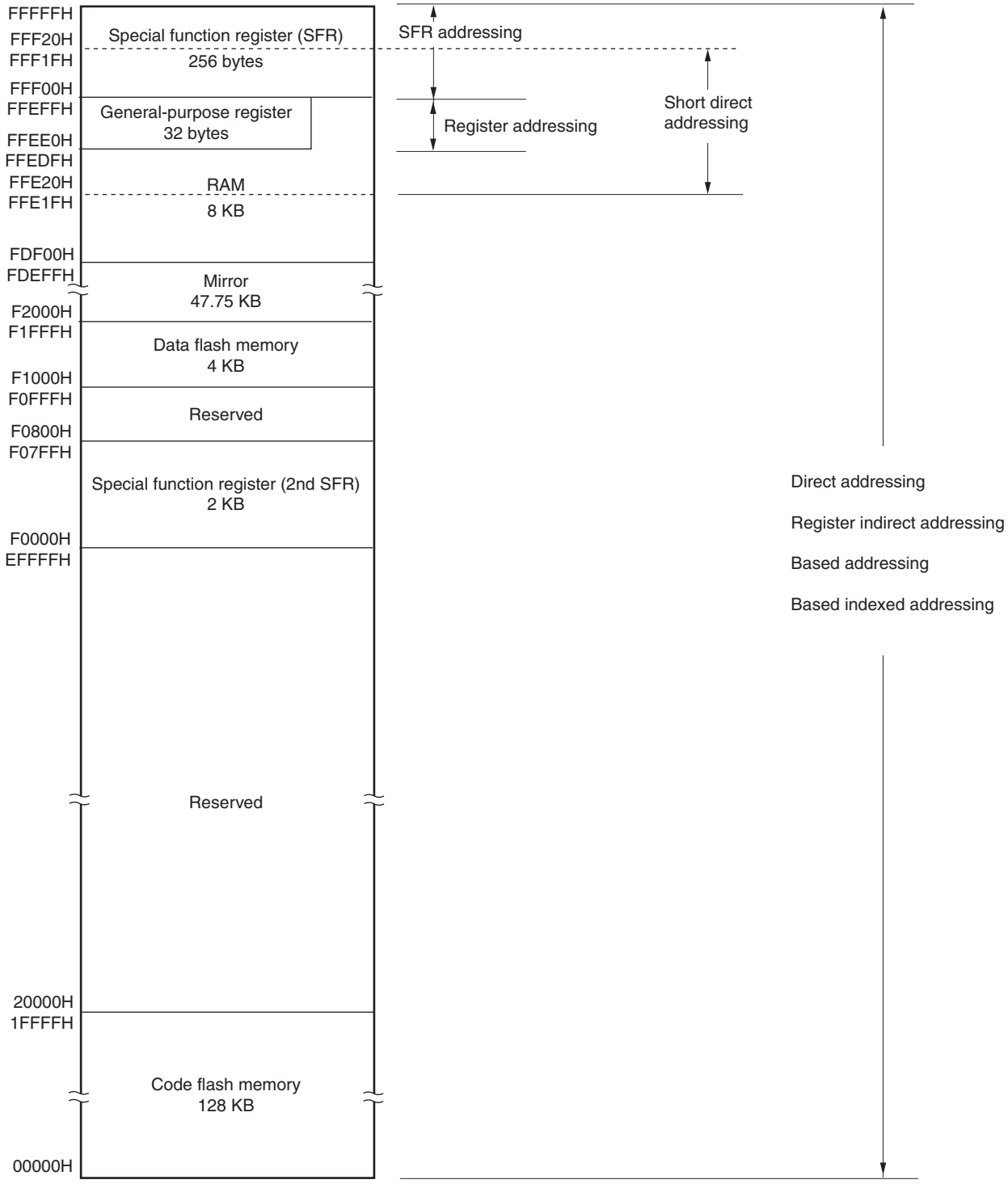

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-31. Correspondence between Data Memory and Addressing (R5F10PPF)**



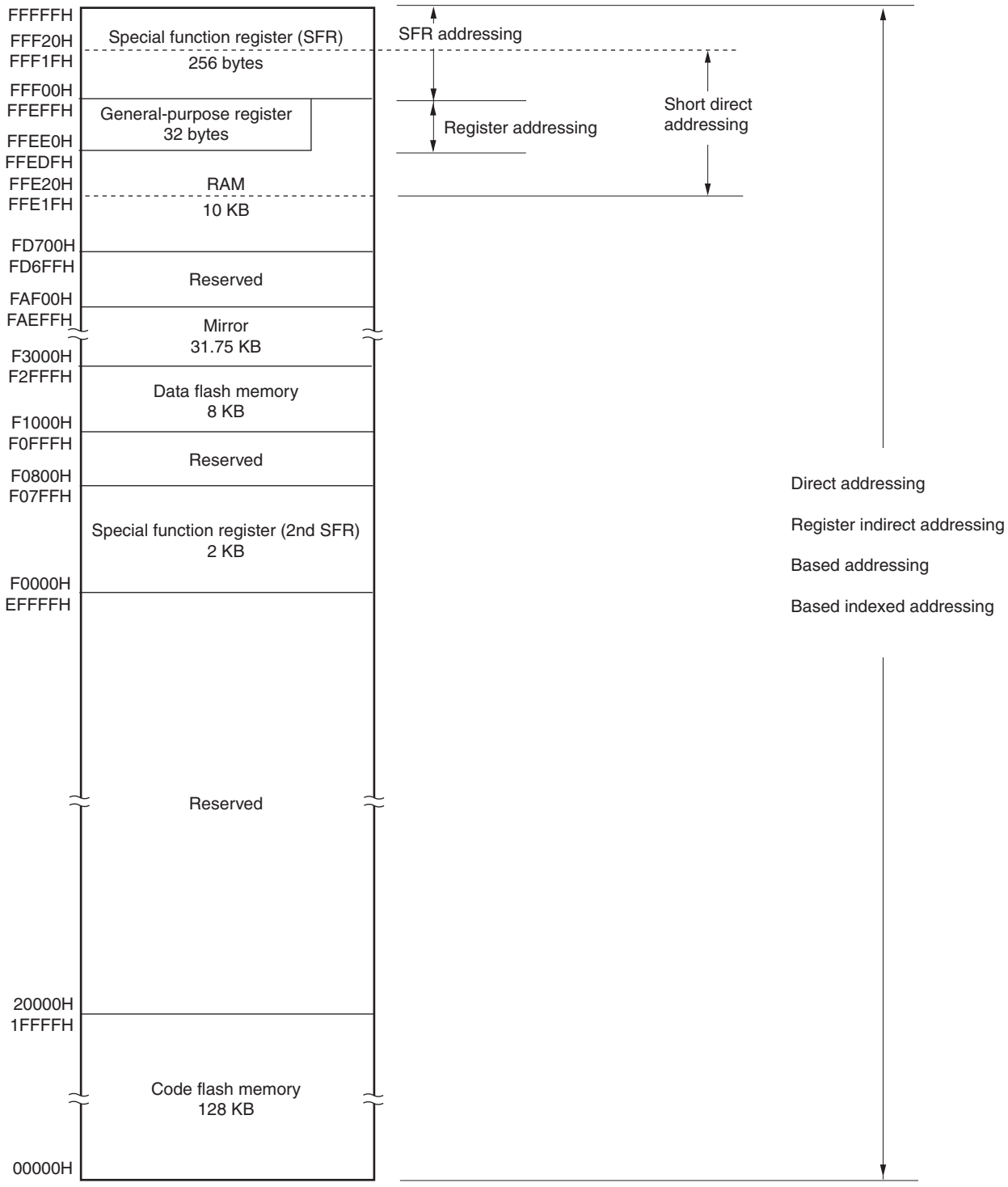

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-32. Correspondence between Data Memory and Addressing (R5F10AmG (m = G, L, M), R5F10BnG (n = A, B, G, L, M))**



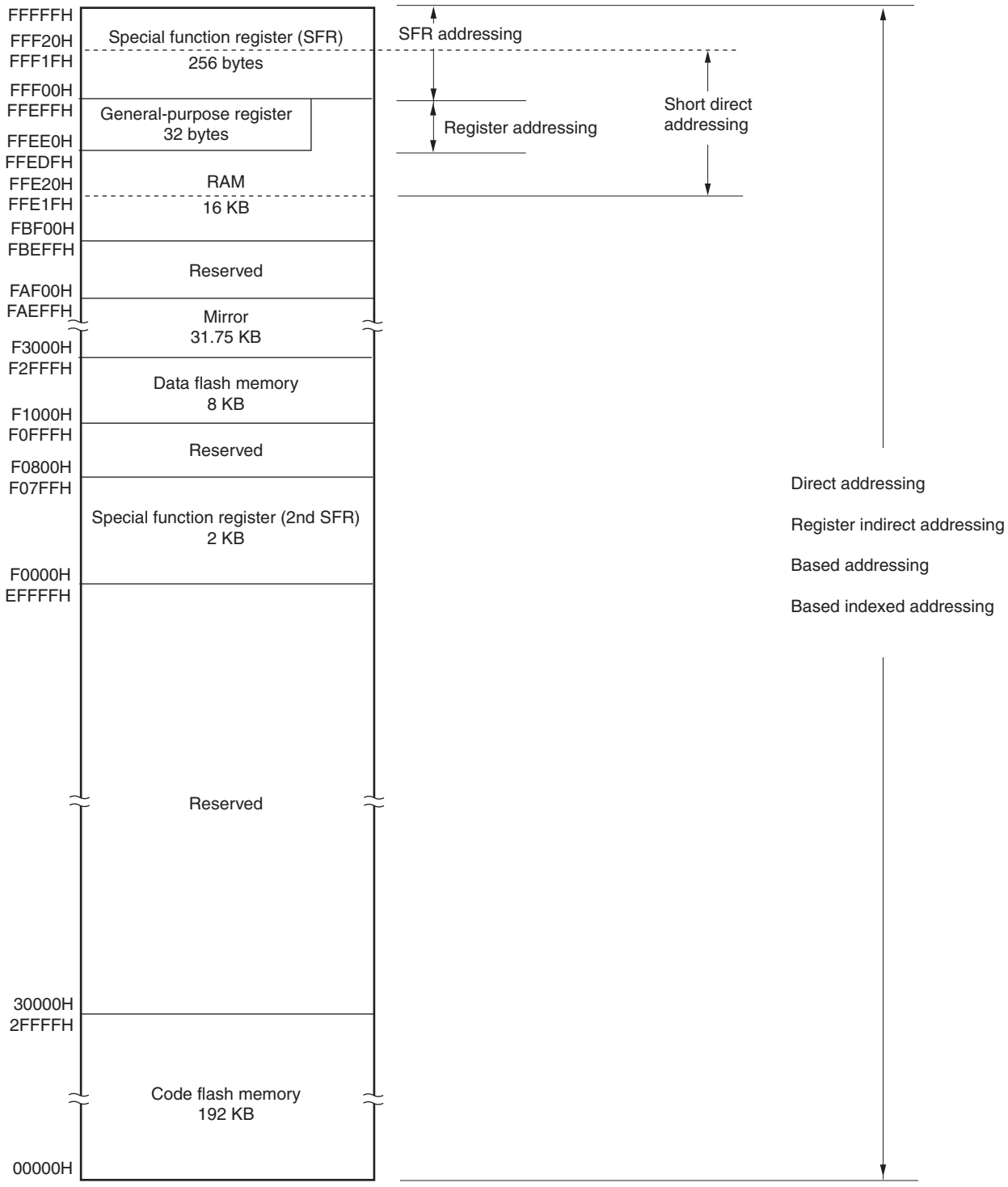

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-33. Correspondence between Data Memory and Addressing (R5F10PmG (m = G, L, M, P))**



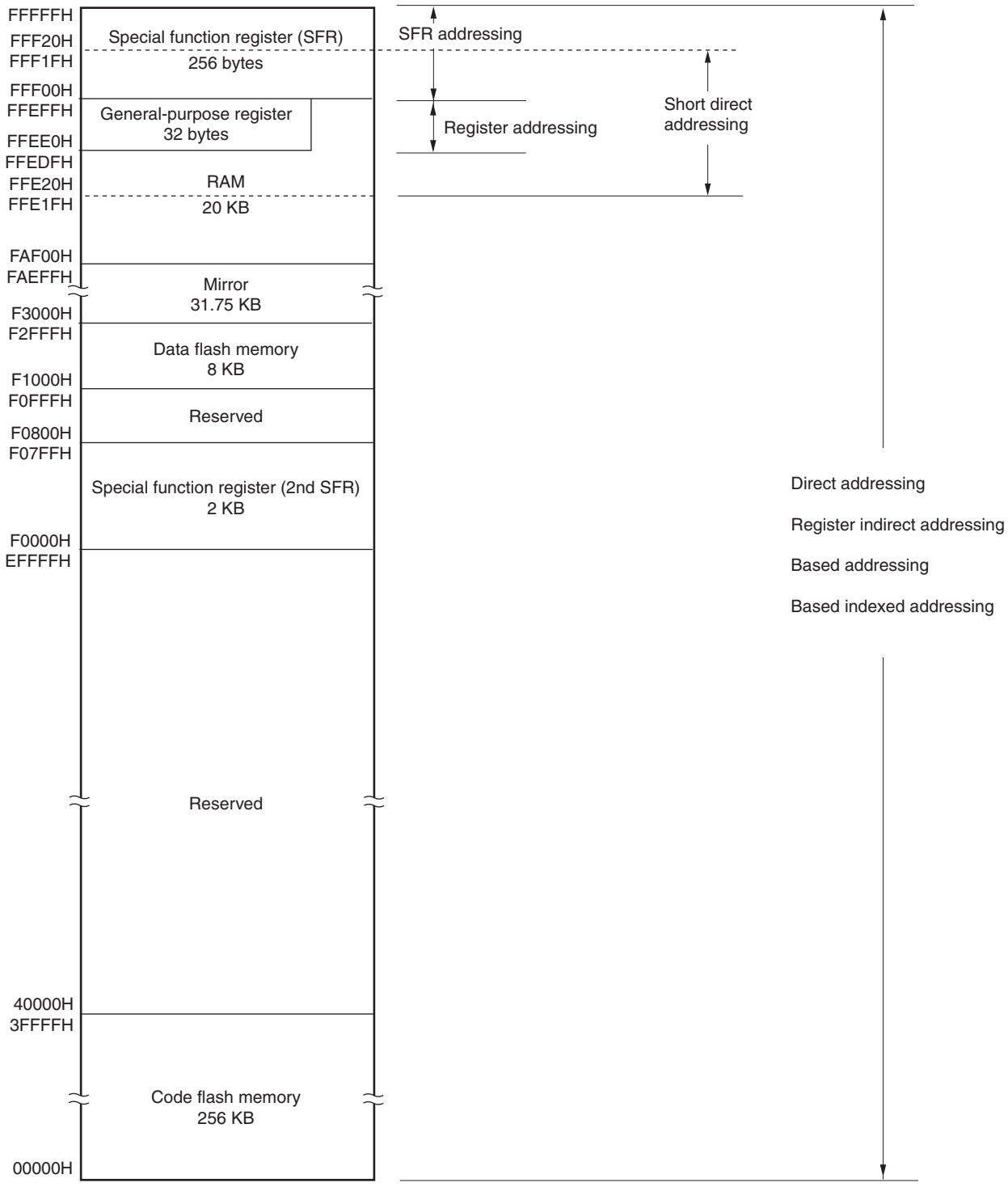

**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-34. Correspondence between Data Memory and Addressing (R5F10PmH (m = G, L, M, P))**




**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

**Figure 3-35. Correspondence between Data Memory and Addressing (R5F10PmJ (m = G, L, M, P))**




**Caution When executing instructions from the RAM area, be sure to initialize the used RAM area + 10 bytes with any desired value.**

## 3.2 Processor Registers

The RL78/F13 and RL78/F14 products incorporate the following processor registers.

### 3.2.1 Control registers

The control registers control the program sequence, statuses and stack memory. The control registers consist of a program counter (PC), a program status word (PSW) and a stack pointer (SP).

**(1) Program counter (PC)**

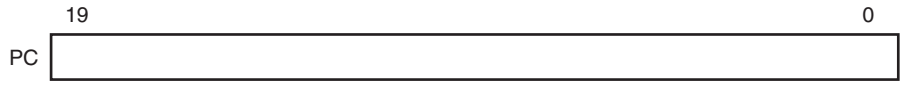
The program counter is a 20-bit register that holds the address information of the next program to be executed.

In normal operation, PC is automatically incremented according to the number of bytes of the instruction to be fetched.

When a branch instruction is executed, immediate data and register contents are set.

Reset signal generation sets the reset vector table values at addresses 0000H and 0001H to the program counter.

**Figure 3-36. Format of Program Counter**




**(2) Program status word (PSW)**

The program status word is an 8-bit register consisting of various flags set/reset by instruction execution.

Program status word contents are stored in the stack area upon vectored interrupt request is acknowledged or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions.

Reset signal generation sets the PSW register to 06H.

**Figure 3-37. Format of Program Status Word**

| | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|
| PSW | IE | Z | RBS1 | AC | RBS0 | ISP1 | ISP0 | CY |

**(a) Interrupt enable flag (IE)**

This flag controls the interrupt request acknowledge operations of the CPU.

When 0, the IE flag is set to the interrupt disabled (DI) state, and all maskable interrupt requests are disabled.

When 1, the IE flag is set to the interrupt enabled (EI) state and interrupt request acknowledgment is controlled with an in-service priority flag (ISP1, ISP0), an interrupt mask flag for various interrupt sources, and a priority specification flag.

The IE flag is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (1) upon EI instruction execution.

**(b) Zero flag (Z)**

When the operation result is zero, this flag is set (1). It is reset (0) in all other cases.

**(c) Register bank select flags (RBS0, RBS1)**

These are 2-bit flags to select one of the four register banks.

In these flags, the 2-bit information that indicates the register bank selected by SEL RBn instruction execution is stored.

**(d) Auxiliary carry flag (AC)**

If the operation result has a carry from bit 3 or a borrow at bit 3, this flag is set (1). It is reset (0) in all other cases.

**(e) In-service priority flags (ISP1, ISP0)**

This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H, PRn3L) (see **21.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L)**) can not be acknowledged.

Actual request acknowledgment is controlled by the interrupt enable flag (IE).

**Remark** n = 0, 1

**(f) Carry flag (CY)**

This flag stores overflow and underflow upon add/subtract instruction execution. It stores the shift-out value upon rotate instruction execution and functions as a bit accumulator during bit operation instruction execution.

**(3) Stack pointer (SP)**

This is a 16-bit register to hold the start address of the memory stack area. Only the internal RAM area can be set as the stack area.

**Figure 3-38. Format of Stack Pointer**

| | 15 | | | | | | | | | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SP | SP15 | SP14 | SP13 | SP12 | SP11 | SP10 | SP9 | SP8 | SP7 | SP6 | SP5 | SP4 | SP3 | SP2 | SP1 | SP0 |

The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

Each stack operation saves data as shown in Figure 3-39.

**Cautions 1. Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.**

**2. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as a stack area.**

**3. The internal RAM in the following products cannot be used as stack memory when using the self-programming and data flash function. However, the area to which this prohibition applies may vary with the version of the library. For details, refer to the manual for the individual library.**

| | |
|---|---|
| **R5F10AmE (m = 6, A, B, G, L):** | **FEF00H to FF2FFH** |
| **R5F10PmF (m = G, L, M):** | **FDF00H to FE2FFH** |
| **R5F10AmG (m = G, L, M), R5F10BnG (n = A, B, G, L, M):** | **FDF00H to FE2FFH** |
| **R5F10PmJ (m = G, L, M, P):** | **FAF00H to FB2FFH** |

**4. The internal RAM in the following products cannot be used as stack memory when using the tracing function of on-chip debugging.**

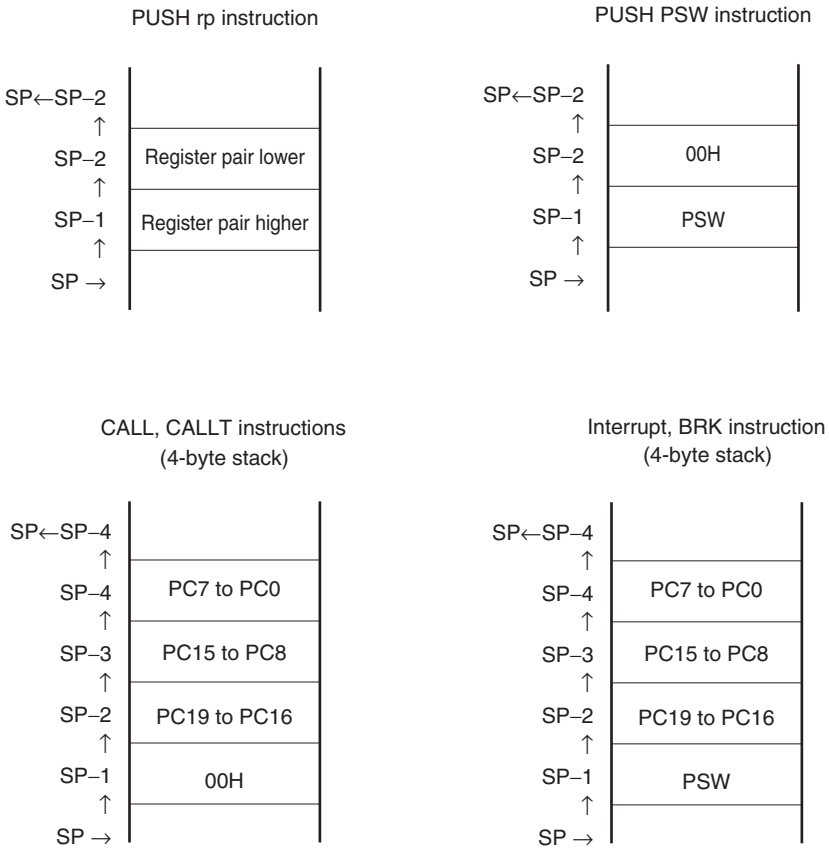
| | |
|---|---|
| **R5F10AmE (m = 6, A, B, G, L), R5F10AnD (n = 6, A, B, G, L):** | **FF300H to FF37FH** |
| **R5F10AmG (m = G, L, M), R5F10BnG (n = A, B, G, L, M):** | **FE300H to FE4FFH** |
| **R5F10PmF (m = G, L, M):** | **FE300H to FE4FFH** |
| **R5F10PmJ (m = G, L, M, P):** | **FB300H to FB4FFH** |

**5. The internal RAM in the following products cannot be used as stack memory when the hot plug-in function is used or when the DTC is in use for the RRM or DMM function.**

| | |
|---|---|
| **R5F10AmD, R5F10AmE (m = 6, A, B, G, L):** | **FF400H to FF42FH** |
| **R5F10AmG (m = G, L, M), R5F10BnG (n = A, B, G, L, M):** | **FE500H to FE52FH** |

**R5F10PmF (m = G, L, M):** **FE500H to FE52FH**
**R5F10PmJ (m = G, L, M, P):** **FB500H to FB52FH**

**Figure 3-39. Data to Be Saved to Stack Memory**



### 3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FFEE0H to FFEFFH) of the data memory. The general-purpose registers consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

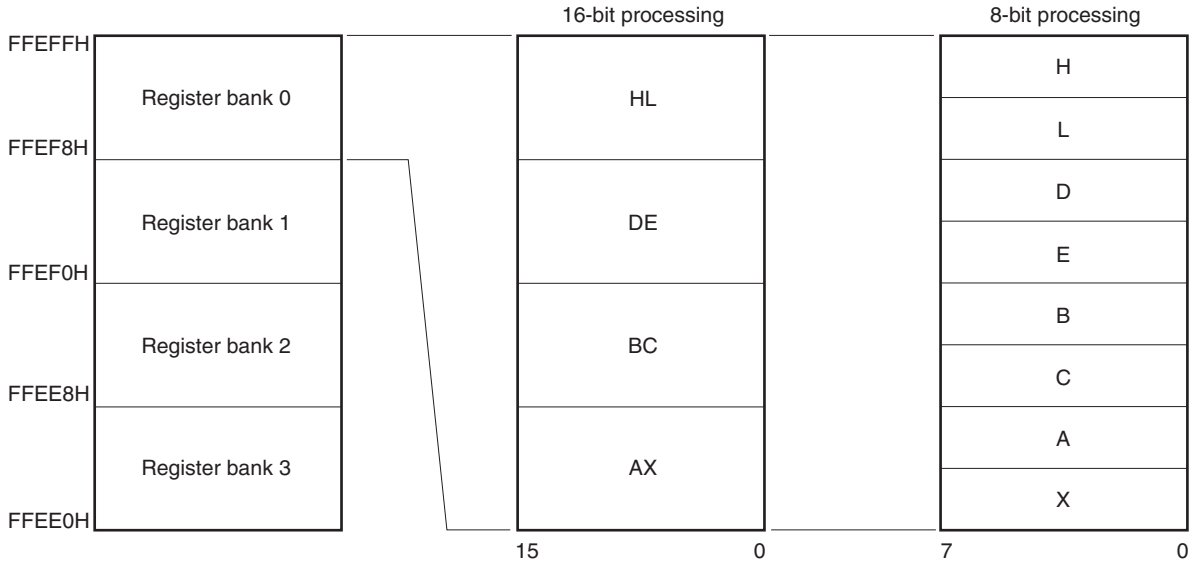
These registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

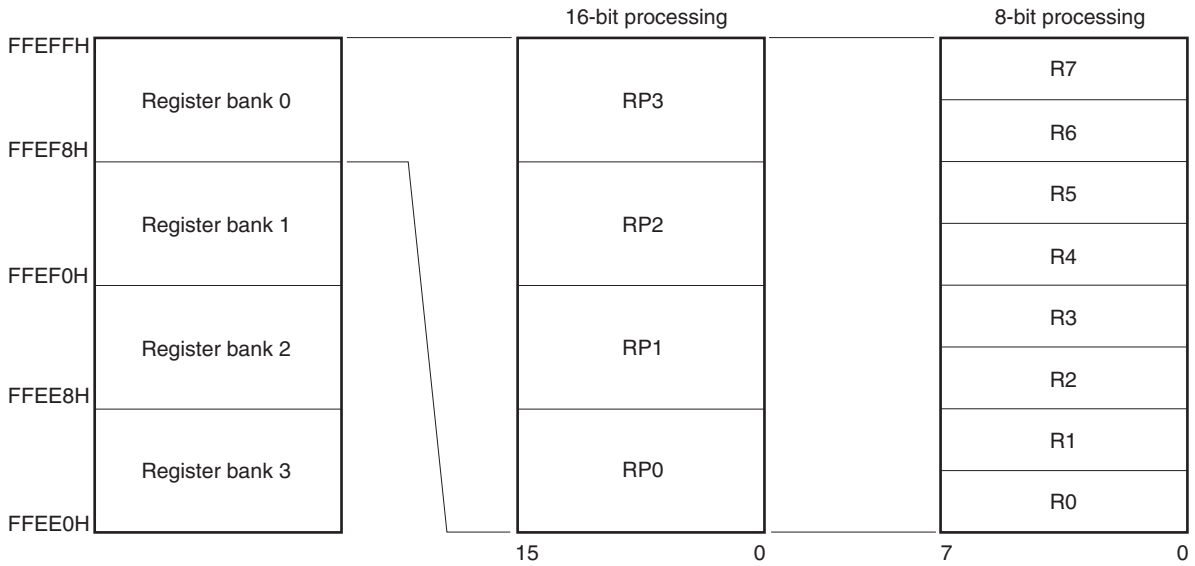
**Caution It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.**

**Figure 3-40. Configuration of General-Purpose Registers**

**(a) Function name**




**(b) Absolute name**



### 3.2.3 ES and CS registers

The ES register is used for data access and the CS register is used to specify the higher address when a branch instruction is executed.

The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

**Figure 3-41. Configuration of ES and CS Registers**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ES | 0 | 0 | 0 | 0 | ES3 | ES2 | ES1 | ES0 |

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CS | 0 | 0 | 0 | 0 | CS3 | CP2 | CP1 | CP0 |

### 3.2.4 Special function registers (SFRs)

Unlike a general-purpose register, each SFR has a special function.

SFRs are allocated to the FFF00H to FFFFFH area.

SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit).
  This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (sfr).
  This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol reserved by the assembler for the 16-bit manipulation instruction operand (sfrp).
  When specifying an address, describe an even address.

Table 3-5 gives a list of the SFRs. The meanings of items in the table are as follows.

- Symbol
  This item indicates the address of a special function register. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  This item indicates whether the corresponding SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  This item indicates each register status upon reset signal generation.

**Caution Do not access addresses to which SFRs are not assigned.**

**Remark** For extended SFRs (2nd SFRs), see **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

**Table 3-5. SFR List (1/4)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| FFF00H | Port register 0 | P0 | | R/W | √ | √ | – | 00H |
| FFF01H | Port register 1 | P1 | | R/W | √ | √ | – | 00H |
| FFF03H | Port register 3 | P3 | | R/W | √ | √ | – | 00H |
| FFF04H | Port register 4 | P4 | | R/W | √ | √ | – | 00H |
| FFF05H | Port register 5 | P5 | | R/W | √ | √ | – | 00H |
| FFF06H | Port register 6 | P6 | | R/W | √ | √ | – | 00H |
| FFF07H | Port register 7 | P7 | | R/W | √ | √ | – | 00H |
| FFF08H | Port register 8 | P8 | | R/W | √ | √ | – | 00H |
| FFF09H | Port register 9 | P9 | | R/W | √ | √ | – | 00H |
| FFF0AH | Port register 10 | P10 | | R/W | √ | √ | – | 00H |
| FFF0CH | Port register 12 | P12 | | R/W | √ | √ | – | Undefined |
| FFF0DH | Port register 13 | P13 | | R/W | √ | √ | – | Undefined Note |
| FFF0EH | Port register 14 | P14 | | R/W | √ | √ | – | 00H |
| FFF0FH | Port register 15 | P15 | | R/W | √ | √ | – | 00H |
| FFF10H | Serial data register 00 | SDR00L | SDR00 | R/W | – | √ | √ | 0000H |
| FFF11H | | – | | | – | – | | |
| FFF12H | Serial data register 01 | SDR01L | SDR01 | R/W | – | √ | √ | 0000H |
| FFF13H | | – | | | – | – | | |
| FFF18H | Timer data register 00 | TDR00 | | R/W | – | – | √ | 0000H |
| FFF19H | | | | | | | | |
| FFF1AH | Timer data register 01 | TDR01L | TDR01 | R/W | – | √ | √ | 00H |
| FFF1BH | | TDR01H | | | – | √ | | 00H |
| FFF1EH | 10-bit A/D conversion result register | ADCR | | R | – | – | √ | 0000H |
| FFF1FH | 8-bit A/D conversion result register | ADCRH | | R | – | √ | – | 00H |
| FFF20H | Port mode register 0 | PM0 | | R/W | √ | √ | – | FFH |
| FFF21H | Port mode register 1 | PM1 | | R/W | √ | √ | – | FFH |
| FFF23H | Port mode register 3 | PM3 | | R/W | √ | √ | – | FFH |
| FFF24H | Port mode register 4 | PM4 | | R/W | √ | √ | – | FFH |
| FFF25H | Port mode register 5 | PM5 | | R/W | √ | √ | – | FFH |
| FFF26H | Port mode register 6 | PM6 | | R/W | √ | √ | – | FFH |
| FFF27H | Port mode register 7 | PM7 | | R/W | √ | √ | – | FFH |
| FFF28H | Port mode register 8 | PM8 | | R/W | √ | √ | – | FFH |
| FFF29H | Port mode register 9 | PM9 | | R/W | √ | √ | – | FFH |
| FFF2AH | Port mode register 10 | PM10 | | R/W | √ | √ | – | FFH |
| FFF2CH | Port mode register 12 | PM12 | | R/W | √ | √ | – | FFH |
| FFF2EH | Port mode register 14 | PM14 | | R/W | √ | √ | – | FFH |
| FFF2FH | Port mode register 15 | PM15 | | R/W | √ | √ | – | FFH |
| FFF30H | A/D converter mode register 0 | ADM0 | | R/W | √ | √ | – | 00H |
| FFF31H | Analog input channel specification register | ADS | | R/W | √ | √ | – | 00H |
| FFF32H | A/D converter mode register 1 | ADM1 | | R/W | √ | √ | – | 00H |
| FFF34H | D/A conversion value setting register 0 | DACS0 | | R/W | – | √ | – | 00H |

<R>

**Note** P130 bit depends on the setting of User Option Byte (000C2H/020C2H).

**Table 3-5. SFR List (2/4)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| FFF36H | D/A converter mode register | DAM | | R/W | √ | √ | – | 00H |
| FFF37H | Key return mode register | KRM | | R/W | √ | √ | – | 00H |
| FFF38H | External interrupt rising edge enable register 0 | EGP0 | | R/W | √ | √ | – | 00H |
| FFF39H | External interrupt falling edge enable register 0 | EGN0 | | R/W | √ | √ | – | 00H |
| FFF3AH | External interrupt rising edge enable register 1 | EGP1 | | R/W | √ | √ | – | 00H |
| FFF3BH | External interrupt falling edge enable register 1 | EGN1 | | R/W | √ | √ | – | 00H |
| FFF48H | Serial data register 10 | SDR10L | SDR10 | R/W | – | √ | √ | 0000H |
| FFF49H | | – | | | – | – | | |
| FFF4AH | Serial data register 11 | SDR11L | SDR11 | R/W | – | √ | √ | 0000H |
| FFF4BH | | – | | | – | – | | |
| FFF50H | IICA shift register 0 | IICA0 | | R/W | – | √ | – | 00H |
| FFF51H | IICA status register 0 | IICS0 | | R | √ | √ | – | 00H |
| FFF52H | IICA flag register 0 | IICF0 | | R/W | √ | √ | – | 00H |
| FFF54H | 16-bit watch error correction register | SUBCUDW | | R/W | – | – | √ | 0000H |
| FFF55H | | | | | | | | |
| FFF58H | Timer RD general register C0 | TRDGRC0 | | R/W | – | – | √ | FFFFH Note |
| FFF59H | | | | | | | | |
| FFF5AH | Timer RD general register D0 | TRDGRD0 | | R/W | – | – | √ | FFFFH Note |
| FFF5BH | | | | | | | | |
| FFF5CH | Timer RD general register C1 | TRDGRC1 | | R/W | – | – | √ | FFFFH Note |
| FFF5DH | | | | | | | | |
| FFF5EH | Timer RD general register D1 | TRDGRD1 | | R/W | – | – | √ | FFFFH Note |
| FFF5FH | | | | | | | | |
| FFF64H | Timer data register 02 | TDR02 | | R/W | – | – | √ | 0000H |
| FFF65H | | | | | | | | |
| FFF66H | Timer data register 03 | TDR03L | TDR03 | R/W | – | √ | √ | 00H |
| FFF67H | | TDR03H | | | – | √ | | 00H |
| FFF68H | Timer data register 04 | TDR04 | | R/W | – | – | √ | 0000H |
| FFF69H | | | | | | | | |
| FFF6AH | Timer data register 05 | TDR05 | | R/W | – | – | √ | 0000H |
| FFF6BH | | | | | | | | |
| FFF6CH | Timer data register 06 | TDR06 | | R/W | – | – | √ | 0000H |
| FFF6DH | | | | | | | | |
| FFF6EH | Timer data register 07 | TDR07 | | R/W | – | – | √ | 0000H |
| FFF6FH | | | | | | | | |

**Note** The timer RD SFRs are undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**Table 3-5. SFR List (3/4)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| FFF70H | Timer data register 10 | TDR10 | | R/W | – | – | √ | 0000H |
| FFF71H | | | | | | | | |
| FFF72H | Timer data register 11 | TDR11L | TDR11 | R/W | – | √ | √ | 00H |
| FFF73H | | TDR11H | | | – | √ | | 00H |
| FFF74H | Timer data register 12 | TDR12 | | R/W | – | – | √ | 0000H |
| FFF75H | | | | | | | | |
| FFF76H | Timer data register 13 | TDR13L | TDR13 | R/W | – | √ | √ | 00H |
| FFF77H | | TDR13H | | | – | √ | | 00H |
| FFF78H | Timer data register 14 | TDR14 | | R/W | – | – | √ | 0000H |
| FFF79H | | | | | | | | |
| FFF7AH | Timer data register 15 | TDR15 | | R/W | – | – | √ | 0000H |
| FFF7BH | | | | | | | | |
| FFF7CH | Timer data register 16 | TDR16 | | R/W | – | – | √ | 0000H |
| FFF7DH | | | | | | | | |
| FFF7EH | Timer data register 17 | TDR17 | | R/W | – | – | √ | 0000H |
| FFF7FH | | | | | | | | |
| FFF92H | Second count register | SEC | | R/W | – | √ | – | 00H |
| FFF93H | Minute count register | MIN | | R/W | – | √ | – | 00H |
| FFF94H | Hour count register | HOUR | | R/W | – | √ | – | 12H Note 1 |
| FFF95H | Week count register | WEEK | | R/W | – | √ | – | 00H |
| FFF96H | Day count register | DAY | | R/W | – | √ | – | 01H |
| FFF97H | Month count register | MONTH | | R/W | – | √ | – | 01H |
| FFF98H | Year count register | YEAR | | R/W | – | √ | – | 00H |
| FFF99H | Watch error correction register | SUBCUD | | R/W | – | √ | – | 00H |
| FFF9AH | Alarm minute register | ALARMWM | | R/W | – | √ | – | 00H |
| FFF9BH | Alarm hour register | ALARMWH | | R/W | – | √ | – | 12H |
| FFF9CH | Alarm week register | ALARMWW | | R/W | – | √ | – | 00H |
| FFF9DH | Real-time clock control register 0 | RTCC0 | | R/W | √ | √ | – | 00H |
| FFF9EH | Real-time clock control register 1 | RTCC1 | | R/W | √ | √ | – | 00H |
| FFFA0H | Clock operation mode control register | CMC | | R/W | – | √ | – | 00H |
| FFFA1H | Clock operation status control register | CSC | | R/W | √ | √ | – | C0H |
| FFFA2H | Oscillation stabilization time counter status register | OSTC | | R | √ | √ | – | 00H |
| FFFA3H | Oscillation stabilization time select register | OSTS | | R/W | – | √ | – | 07H |
| FFFA4H | System clock control register | CKC | | R/W | √ | √ | – | 00H |
| FFFA5H | Clock output select register 0 | CKS0 | | R/W | √ | √ | – | 00H |
| FFFA8H | Reset control flag register | RESF | | R | – | √ | – | Undefined Note 2 |
| FFFA9H | Voltage detection register | LVIM | | R/W | √ | √ | – | 00H Note 3 |

**Notes 1.** The value of this register is 00H if the AMPM bit (bit 3 of real-time clock control register 0 (RTCC0)) is set to 1 after reset.

**2.** The reset value of the RESF register varies depending on the reset source.

**3.** The reset value of the LVIM register varies depending on the reset source.

**Table 3-5. SFR List (4/4)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | Undefined |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| FFFAAH | Voltage detection level register | LVIS | | R/W | √ | √ | – | 00H/01H/81H Note 1 |
| FFFABH | Watchdog timer enable register | WDTE | | R/W | – | √ | – | 1AH/9AH Note 2 |
| FFFACH | CRC input register | CRCIN | | R/W | – | √ | – | 00H |
| FFFD0H | Interrupt request flag register 2L | IF2L | IF2 | R/W | √ | √ | √ | 00H |
| FFFD1H | Interrupt request flag register 2H | IF2H | | R/W | √ | √ | | 00H |
| FFFD2H | Interrupt request flag register 3L | IF3L | | R/W | √ | √ | – | 00H |
| FFFD4H | Interrupt mask flag register 2L | MK2L | MK2 | R/W | √ | √ | √ | FFH |
| FFFD5H | Interrupt mask flag register 2H | MK2H | | R/W | √ | √ | | FFH |
| FFFD6H | Interrupt mask flag register 3L | MK3L | | R/W | √ | √ | – | FFH |
| FFFD8H | Priority specification flag register 02L | PR02L | PR02 | R/W | √ | √ | √ | FFH |
| FFFD9H | Priority specification flag register 02H | PR02H | | R/W | √ | √ | | FFH |
| FFFDAH | Priority specification flag register 03L | PR03L | | R/W | √ | √ | – | FFH |
| FFFDCH | Priority specification flag register 12L | PR12L | PR12 | R/W | √ | √ | √ | FFH |
| FFFDDH | Priority specification flag register 12H | PR12H | | R/W | √ | √ | | FFH |
| FFFDEH | Priority specification flag register 13L | PR13L | | R/W | √ | √ | – | FFH |
| FFFE0H | Interrupt request flag register 0L | IF0L | IF0 | R/W | √ | √ | √ | 00H |
| FFFE1H | Interrupt request flag register 0H | IF0H | | R/W | √ | √ | | 00H |
| FFFE2H | Interrupt request flag register 1L | IF1L | IF1 | R/W | √ | √ | √ | 00H |
| FFFE3H | Interrupt request flag register 1H | IF1H | | R/W | √ | √ | | 00H |
| FFFE4H | Interrupt mask flag register 0L | MK0L | MK0 | R/W | √ | √ | √ | FFH |
| FFFE5H | Interrupt mask flag register 0H | MK0H | | R/W | √ | √ | | FFH |
| FFFE6H | Interrupt mask flag register 1L | MK1L | MK1 | R/W | √ | √ | √ | FFH |
| FFFE7H | Interrupt mask flag register 1H | MK1H | | R/W | √ | √ | | FFH |
| FFFE8H | Priority specification flag register 00L | PR00L | PR00 | R/W | √ | √ | √ | FFH |
| FFFE9H | Priority specification flag register 00H | PR00H | | R/W | √ | √ | | FFH |
| FFFEAH | Priority specification flag register 01L | PR01L | PR01 | R/W | √ | √ | √ | FFH |
| FFFEBH | Priority specification flag register 01H | PR01H | | R/W | √ | √ | | FFH |
| FFFECH | Priority specification flag register 10L | PR10L | PR10 | R/W | √ | √ | √ | FFH |
| FFFEDH | Priority specification flag register 10H | PR10H | | R/W | √ | √ | | FFH |
| FFFEEH | Priority specification flag register 11L | PR11L | PR11 | R/W | √ | √ | √ | FFH |
| FFFEFH | Priority specification flag register 11H | PR11H | | R/W | √ | √ | | FFH |
| FFFF0H | Multiply and accumulation register (L) | MACRL | | R/W | – | – | √ | 0000H |
| FFFF1H | | | | | | | | |
| FFFF2H | Multiply and accumulation register (H) | MACRH | | R/W | – | – | √ | 0000H |
| FFFF3H | | | | | | | | |
| FFFFEH | Processor mode control register | PMC | | R/W | √ | √ | – | 00H |

**Notes 1.** The reset value of the LVIS register varies depending on the reset source and the setting of the option byte.

**2.** The reset value of the WDTE register is determined by the setting of the option byte.

**Remark** For extended SFRs (2nd SFRs), see **Table 3-6 Extended SFR (2nd SFR) List**.

### 3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)

Unlike a general-purpose register, each extended SFR (2nd SFR) has a special function.

Extended SFRs are allocated to the F0000H to F07FFH area. SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

Extended SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (laddr16.bit). This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (laddr16). This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol reserved by the assembler for the 16-bit manipulation instruction operand (laddr16s). When specifying an address, describe an even address.

Table 3-6 gives a list of the extended SFRs. The meanings of items in the table are as follows.

- Symbol
  This item indicates the address of an extended SFR. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  This item indicates whether the corresponding extended SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  This item indicates each register status upon reset signal generation.

**Caution Do not access addresses to which extended SFRs are not assigned.**

**Remark** For SFRs in the SFR area, see **3.2.4 Special function registers (SFRs)**.

**Table 3-6. Extended SFR (2nd SFR) List (1/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|
| | | | | 1-bit | 8-bit | 16-bit | |
| F0010H | A/D converter mode register 2 | ADM2 | R/W | √ | √ | – | 00H |
| F0011H | Conversion result comparison upper limit setting register | ADUL | R/W | – | √ | – | FFH |
| F0012H | Conversion result comparison lower limit setting register | ADLL | R/W | – | √ | – | 00H |
| F0013H | A/D test register | ADTES | R/W | – | √ | – | 00H |
| F0016H | Peripheral I/O redirection register 0 | PIOR0 | R/W | – | √ | – | 00H |
| F0017H | Peripheral I/O redirection register 1 | PIOR1 | R/W | – | √ | – | 00H |
| F0018H | Peripheral I/O redirection register 2 | PIOR2 | R/W | – | √ | – | 00H |
| F0019H | Peripheral I/O redirection register 3 | PIOR3 | R/W | – | √ | – | 00H |
| F001AH | Peripheral I/O redirection register 4 | PIOR4 | R/W | – | √ | – | 00H |
| F001BH | Peripheral I/O redirection register 5 | PIOR5 | R/W | – | √ | – | 00H |
| F001CH | Peripheral I/O redirection register 6 | PIOR6 | R/W | – | √ | – | 00H |
| F001DH | Peripheral I/O redirection register 7 | PIOR7 | R/W | – | √ | – | 00H |
| F001EH | Peripheral I/O redirection register 8 | PIOR8 | R/W | – | √ | – | 00H |
| F0021H | Port input threshold control register 1 | PITHL1 | R/W | √ | √ | – | 00H |
| F0023H | Port input threshold control register 3 | PITHL3 | R/W | √ | √ | – | 00H |
| F0024H | Port input threshold control register 4 | PITHL4 | R/W | √ | √ | – | 00H |
| F0025H | Port input threshold control register 5 | PITHL5 | R/W | √ | √ | – | 00H |
| F0026H | Port input threshold control register 6 | PITHL6 | R/W | √ | √ | – | 00H |
| F0027H | Port input threshold control register 7 | PITHL7 | R/W | √ | √ | – | 00H |
| F002AH | Port input threshold control register 10 | PITHL10 | R/W | √ | √ | – | 00H |
| F002CH | Port input threshold control register 12 | PITHL12 | R/W | √ | √ | – | 00H |
| F002FH | Port input threshold control register 15 | PITHL15 | R/W | √ | √ | – | 00H |
| F0030H | Pull-up resistor option register 0 | PU0 | R/W | √ | √ | – | 00H |
| F0031H | Pull-up resistor option register 1 | PU1 | R/W | √ | √ | – | 00H |
| F0033H | Pull-up resistor option register 3 | PU3 | R/W | √ | √ | – | 00H |
| F0034H | Pull-up resistor option register 4 | PU4 | R/W | √ | √ | – | 01H |
| F0035H | Pull-up resistor option register 5 | PU5 | R/W | √ | √ | – | 00H |
| F0036H | Pull-up resistor option register 6 | PU6 | R/W | √ | √ | – | 00H |
| F0037H | Pull-up resistor option register 7 | PU7 | R/W | √ | √ | – | 00H |
| F0039H | Pull-up resistor option register 9 | PU9 | R/W | √ | √ | – | 00H |
| F003AH | Pull-up resistor option register 10 | PU10 | R/W | √ | √ | – | 00H |
| F003CH | Pull-up resistor option register 12 | PU12 | R/W | √ | √ | – | 00H |
| F003EH | Pull-up resistor option register 14 | PU14 | R/W | √ | √ | – | 00H |
| F003FH | Pull-up resistor option register 15 | PU15 | R/W | √ | √ | – | 00H |
| F0041H | Port input mode register 1 | PIM1 | R/W | √ | √ | – | 00H |
| F0043H | Port input mode register 3 | PIM3 | R/W | √ | √ | – | 00H |
| F0045H | Port input mode register 5 | PIM5 | R/W | √ | √ | – | 00H |
| F0046H | Port input mode register 6 | PIM6 | R/W | √ | √ | – | 00H |
| F0047H | Port input mode register 7 | PIM7 | R/W | √ | √ | – | 00H |
| F004CH | Port input mode register 12 | PIM12 | R/W | √ | √ | – | 00H |
| F0051H | Port output mode register 1 | POM1 | R/W | √ | √ | – | 00H |
| F0056H | Port output mode register 6 | POM6 | R/W | √ | √ | – | 00H |
| F0057H | Port output mode register 7 | POM7 | R/W | √ | √ | – | 00H |

**Table 3-6. Extended SFR (2nd SFR) List (2/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F005CH | Port output mode register 12 | POM12 | | R/W | √ | √ | – | 00H |
| F0067H | Port mode control register 7 | PMC7 | | R/W | √ | √ | – | FFH |
| F0069H | Port mode control register 9 | PMC9 | | R/W | √ | √ | – | FFH |
| F006CH | Port mode control register 12 | PMC12 | | R/W | √ | √ | – | FFH |
| F0070H | Noise filter enable register 0 | NFEN0 | | R/W | √ | √ | – | 00H |
| F0071H | Noise filter enable register 1 | NFEN1 | | R/W | √ | √ | – | 00H |
| F0072H | Noise filter enable register 2 | NFEN2 | | R/W | √ | √ | – | 00H |
| F0073H | Input switch control register | ISC | | R/W | √ | √ | – | 00H |
| F0074H | Timer input select register 0 | TIS0 | | R/W | – | √ | – | 00H |
| F0075H | Timer input select register 1 | TIS1 | | R/W | – | √ | – | 00H |
| F0076H | A/D port configuration register | ADPC | | R/W | – | √ | – | 00H |
| F0077H | Port mode select register | PMS | | R/W | √ | √ | – | 00H |
| F0078H | Invalid memory access detection control register | IAWCTL | | R/W | – | √ | – | 00H |
| F0079H | Interrupt source determination flag register 0 | INTFLG0 | | R/W | – | √ | – | 00H |
| F007AH | Timer input select register 2 | TIS2 | | R/W | – | √ | – | 00H |
| F007BH | LIN channel select register | LCHSEL | | R/W | – | √ | – | 00H |
| F007CH | Interrupt mask register | INTMSK | | R/W | – | √ | – | FFH |
| F0090H | Data flash control register | DFLCTL | | R/W | √ | √ | – | 00H |
| F00A0H | High-speed on-chip oscillator trimming register | HIOTRM | | R/W | – | √ | – | Note |
| F00A8H | High-speed on-chip oscillator frequency select register | HOCODIV | | R/W | – | √ | – | Undefined |
| F00D8H | SPM control register | SPMCTRL | | R/W | – | √ | – | 00H |
| F00DAH | SP overflow address setting register | SPOFR | | R/W | – | – | √ | FFFEH |
| F00DBH | | | | | | | | |
| F00DCH | SP underflow address setting register | SPUFR | | R/W | – | – | √ | 0000H |
| F00DDH | | | | | | | | |
| F00F0H | Peripheral enable register 0 | PER0 | | R/W | √ | √ | – | 00H |
| F00F3H | Operation speed mode control register | OSMC | | R/W | – | √ | – | 00H |
| F00FEH | BCD correction result register | BCDADJ | | R | – | √ | – | Undefined |
| F0100H | Serial status register 00 | SSR00L | SSR00 | R | – | √ | √ | 0000H |
| F0101H | | – | | | – | – | | |
| F0102H | Serial status register 01 | SSR01L | SSR01 | R | – | √ | √ | 0000H |
| F0103H | | – | | | – | – | | |
| F0104H | Serial flag clear trigger register 00 | SIR00L | SIR00 | R/W | – | √ | √ | 0000H |
| F0105H | | – | | | – | – | | |
| F0106H | Serial flag clear trigger register 01 | SIR01L | SIR01 | R/W | – | √ | √ | 0000H |
| F0107H | | – | | | – | – | | |

**Note** The reset value differs for each chip.

**Table 3-6. Extended SFR (2nd SFR) List (3/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F0108H | Serial mode register 00 | SMR00 | | R/W | – | – | √ | 0020H |
| F0109H | | | | | | | | |
| F010AH | Serial mode register 01 | SMR01 | | R/W | – | – | √ | 0020H |
| F010BH | | | | | | | | |
| F010CH | Serial communication operation setting register 00 | SCR00 | | R/W | – | – | √ | 0087H |
| F010DH | | | | | | | | |
| F010EH | Serial communication operation setting register 01 | SCR01 | | R/W | – | – | √ | 0087H |
| F010FH | | | | | | | | |
| F0110H | Serial channel enable status register | SE0L | SE0 | R | √ | √ | √ | 0000H |
| F0111H | | – | | | – | – | | |
| F0112H | Serial channel start register 0 | SS0L | SS0 | R/W | √ | √ | √ | 0000H |
| F0113H | | – | | | – | – | | |
| F0114H | Serial channel stop register 0 | ST0L | ST0 | R/W | √ | √ | √ | 0000H |
| F0115H | | – | | | – | – | | |
| F0116H | Serial clock select register 0 | SPS0L | SPS0 | R/W | – | √ | √ | 0000H |
| F0117H | | – | | | – | – | | |
| F0118H | Serial output register 0 | SO0 | | R/W | – | – | √ | 0303H |
| F0119H | | | | | | | | |
| F011AH | Serial output enable register 0 | SOE0L | SOE0 | R/W | √ | √ | √ | 0000H |
| F011BH | | – | | | – | – | | |
| F0120H | Serial output level register 0 | SOL0L | SOL0 | R/W | – | √ | √ | 0000H |
| F0121H | | – | | | – | – | | |
| F0122H | Serial slave select enable register 0 | SSE0L | SSE0 | R/W | – | √ | √ | 0000H |
| F0123H | | – | | | – | – | | |
| F0140H | Serial status register 10 | SSR10L | SSR10 | R | – | √ | √ | 0000H |
| F0141H | | – | | | – | – | | |
| F0142H | Serial status register 11 | SSR11L | SSR11 | R | – | √ | √ | 0000H |
| F0143H | | – | | | – | – | | |
| F0144H | Serial flag clear trigger register 10 | SIR10L | SIR10 | R/W | – | √ | √ | 0000H |
| F0145H | | – | | | – | – | | |
| F0146H | Serial flag clear trigger register 11 | SIR11L | SIR11 | R/W | – | √ | √ | 0000H |
| F0147H | | – | | | – | – | | |
| F0148H | Serial mode register 10 | SMR10 | | R/W | – | – | √ | 0020H |
| F0149H | | | | | | | | |
| F014AH | Serial mode register 11 | SMR11 | | R/W | – | – | √ | 0020H |
| F014BH | | | | | | | | |
| F014CH | Serial communication operation setting register 10 | SCR10 | | R/W | – | – | √ | 0087H |
| F014DH | | | | | | | | |
| F014EH | Serial communication operation setting register 11 | SCR11 | | R/W | – | – | √ | 0087H |
| F014FH | | | | | | | | |
| F0150H | Serial channel enable status register 1 | SE1L | SE1 | R | √ | √ | √ | 0000H |
| F0151H | | – | | | – | – | | |

**Table 3-6. Extended SFR (2nd SFR) List (4/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F0152H | Serial channel start register 1 | SS1L | SS1 | R/W | √ | √ | √ | 0000H |
| F0153H | | – | | | – | – | | |
| F0154H | Serial channel stop register 1 | ST1L | ST1 | R/W | √ | √ | √ | 0000H |
| F0155H | | – | | | – | – | | |
| F0156H | Serial clock select register 1 | SPS1L | SPS1 | R/W | – | √ | √ | 0000H |
| F0157H | | – | | | – | – | | |
| F0158H | Serial output register 1 | SO1 | | R/W | – | – | √ | 0303H |
| F0159H | | | | | | | | |
| F015AH | Serial output enable register 1 | SOE1L | SOE1 | R/W | √ | √ | √ | 0000H |
| F015BH | | – | | | – | – | | |
| F0160H | Serial output level register 1 | SOL1L | SOL1 | R/W | – | √ | √ | 0000H |
| F0161H | | – | | | – | – | | |
| F0162H | Serial slave select enable register 1 | SSE1L | SSE1 | R/W | – | √ | √ | 0000H |
| F0163H | | – | | | – | – | | |
| F0180H | Timer counter register 00 | TCR00 | | R | – | – | √ | FFFFH |
| F0181H | | | | | | | | |
| F0182H | Timer counter register 01 | TCR01 | | R | – | – | √ | FFFFH |
| F0183H | | | | | | | | |
| F0184H | Timer counter register 02 | TCR02 | | R | – | – | √ | FFFFH |
| F0185H | | | | | | | | |
| F0186H | Timer counter register 03 | TCR03 | | R | – | – | √ | FFFFH |
| F0187H | | | | | | | | |
| F0188H | Timer counter register 04 | TCR04 | | R | – | – | √ | FFFFH |
| F0189H | | | | | | | | |
| F018AH | Timer counter register 05 | TCR05 | | R | – | – | √ | FFFFH |
| F018BH | | | | | | | | |
| F018CH | Timer counter register 06 | TCR06 | | R | – | – | √ | FFFFH |
| F018DH | | | | | | | | |
| F018EH | Timer counter register 07 | TCR07 | | R | – | – | √ | FFFFH |
| F018FH | | | | | | | | |
| F0190H | Timer mode register 00 | TMR00 | | R/W | – | – | √ | 0000H |
| F0191H | | | | | | | | |
| F0192H | Timer mode register 01 | TMR01 | | R/W | – | – | √ | 0000H |
| F0193H | | | | | | | | |
| F0194H | Timer mode register 02 | TMR02 | | R/W | – | – | √ | 0000H |
| F0195H | | | | | | | | |
| F0196H | Timer mode register 03 | TMR03 | | R/W | – | – | √ | 0000H |
| F0197H | | | | | | | | |
| F0198H | Timer mode register 04 | TMR04 | | R/W | – | – | √ | 0000H |
| F0199H | | | | | | | | |
| F019AH | Timer mode register 05 | TMR05 | | R/W | – | – | √ | 0000H |
| F019BH | | | | | | | | |

**Table 3-6. Extended SFR (2nd SFR) List (5/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F019CH | Timer mode register 06 | TMR06 | | R/W | – | – | √ | 0000H |
| F019DH | | | | | | | | |
| F019EH | Timer mode register 07 | TMR07 | | R/W | – | – | √ | 0000H |
| F019FH | | | | | | | | |
| F01A0H | Timer status register 00 | TSR00L | TSR00 | R | – | √ | √ | 0000H |
| F01A1H | | – | | | – | – | | |
| F01A2H | Timer status register 01 | TSR01L | TSR01 | R | – | √ | √ | 0000H |
| F01A3H | | – | | | – | – | | |
| F01A4H | Timer status register 02 | TSR02L | TSR02 | R | – | √ | √ | 0000H |
| F01A5H | | – | | | – | – | | |
| F01A6H | Timer status register 03 | TSR03L | TSR03 | R | – | √ | √ | 0000H |
| F01A7H | | – | | | – | – | | |
| F01A8H | Timer status register 04 | TSR04L | TSR04 | R | – | √ | √ | 0000H |
| F01A9H | | – | | | – | – | | |
| F01AAH | Timer status register 05 | TSR05L | TSR05 | R | – | √ | √ | 0000H |
| F01ABH | | – | | | – | – | | |
| F01ACH | Timer status register 06 | TSR06L | TSR06 | R | – | √ | √ | 0000H |
| F01ADH | | – | | | – | – | | |
| F01AEH | Timer status register 07 | TSR07L | TSR07 | R | – | √ | √ | 0000H |
| F01AFH | | – | | | – | – | | |
| F01B0H | Timer channel enable status register 0 | TE0L | TE0 | R | √ | √ | √ | 0000H |
| F01B1H | | – | | | – | – | | |
| F01B2H | Timer channel start register 0 | TS0L | TS0 | R/W | √ | √ | √ | 0000H |
| F01B3H | | – | | | – | – | | |
| F01B4H | Timer channel stop register 0 | TT0L | TT0 | R/W | √ | √ | √ | 0000H |
| F01B5H | | – | | | – | – | | |
| F01B6H | Timer clock select register 0 | TPS0 | | R/W | – | – | √ | 0000H |
| F01B7H | | | | | | | | |
| F01B8H | Timer output register 0 | TO0L | TO0 | R/W | – | √ | √ | 0000H |
| F01B9H | | – | | | – | – | | |
| F01BAH | Timer output enable register 0 | TOE0L | TOE0 | R/W | √ | √ | √ | 0000H |
| F01BBH | | – | | | – | – | | |
| F01BCH | Timer output level register 0 | TOL0L | TOL0 | R/W | – | √ | √ | 0000H |
| F01BDH | | – | | | – | – | | |
| F01BEH | Timer output mode register 0 | TOM0L | TOM0 | R/W | – | √ | √ | 0000H |
| F01BFH | | – | | | – | – | | |
| F01C0H | Timer counter register 10 | TCR10 | | R | – | – | √ | FFFFH |
| F01C1H | | | | | | | | |
| F01C2H | Timer counter register 11 | TCR11 | | R | – | – | √ | FFFFH |
| F01C3H | | | | | | | | |
| F01C4H | Timer counter register 12 | TCR12 | | R | – | – | √ | FFFFH |
| F01C5H | | | | | | | | |

**Table 3-6. Extended SFR (2nd SFR) List (6/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F01C6H | Timer counter register 13 | TCR13 | | R | – | – | √ | FFFFH |
| F01C7H | | | | | | | | |
| F01C8H | Timer counter register 14 | TCR14 | | R | – | – | √ | FFFFH |
| F01C9H | | | | | | | | |
| F01CAH | Timer counter register 15 | TCR15 | | R | – | – | √ | FFFFH |
| F01CBH | | | | | | | | |
| F01CCH | Timer counter register 16 | TCR16 | | R | – | – | √ | FFFFH |
| F01CDH | | | | | | | | |
| F01CEH | Timer counter register 17 | TCR17 | | R | – | – | √ | FFFFH |
| F01CFH | | | | | | | | |
| F01D0H | Timer mode register 10 | TMR10 | | R/W | – | – | √ | 0000H |
| F01D1H | | | | | | | | |
| F01D2H | Timer mode register 11 | TMR11 | | R/W | – | – | √ | 0000H |
| F01D3H | | | | | | | | |
| F01D4H | Timer mode register 12 | TMR12 | | R/W | – | – | √ | 0000H |
| F01D5H | | | | | | | | |
| F01D6H | Timer mode register 13 | TMR13 | | R/W | – | – | √ | 0000H |
| F01D7H | | | | | | | | |
| F01D8H | Timer mode register 14 | TMR14 | | R/W | – | – | √ | 0000H |
| F01D9H | | | | | | | | |
| F01DAH | Timer mode register 15 | TMR15 | | R/W | – | – | √ | 0000H |
| F01DBH | | | | | | | | |
| F01DCH | Timer mode register 16 | TMR16 | | R/W | – | – | √ | 0000H |
| F01DDH | | | | | | | | |
| F01DEH | Timer mode register 17 | TMR17 | | R/W | – | – | √ | 0000H |
| F01DFH | | | | | | | | |
| F01E0H | Timer status register 10 | TSR10L | TSR10 | R | – | √ | √ | 0000H |
| F01E1H | | – | | | – | – | | |
| F01E2H | Timer status register 11 | TSR11L | TSR11 | R | – | √ | √ | 0000H |
| F01E3H | | – | | | – | – | | |
| F01E4H | Timer status register 12 | TSR12L | TSR12 | R | – | √ | √ | 0000H |
| F01E5H | | – | | | – | – | | |
| F01E6H | Timer status register 13 | TSR13L | TSR13 | R | – | √ | √ | 0000H |
| F01E7H | | – | | | – | – | | |
| F01E8H | Timer status register 14 | TSR14L | TSR14 | R | – | √ | √ | 0000H |
| F01E9H | | – | | | – | – | | |
| F01EAH | Timer status register 15 | TSR15L | TSR15 | R | – | √ | √ | 0000H |
| F01EBH | | – | | | – | – | | |
| F01ECH | Timer status register 16 | TSR16L | TSR16 | R | – | √ | √ | 0000H |
| F01EDH | | – | | | – | – | | |
| F01EEH | Timer status register 17 | TSR17L | TSR17 | R | – | √ | √ | 0000H |
| F01EFH | | – | | | – | – | | |

**Table 3-6. Extended SFR (2nd SFR) List (7/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F01F0H | Timer channel enable status register 1 | TE1L | TE1 | R | √ | √ | √ | 0000H |
| F01F1H | | – | | | – | – | | |
| F01F2H | Timer channel start register 1 | TS1L | TS1 | R/W | √ | √ | √ | 0000H |
| F01F3H | | – | | | – | – | | |
| F01F4H | Timer channel stop register 1 | TT1L | TT1 | R/W | √ | √ | √ | 0000H |
| F01F5H | | – | | | – | – | | |
| F01F6H | Timer clock select register 1 | TPS1 | | R/W | – | – | √ | 0000H |
| F01F7H | | | | | | | | |
| F01F8H | Timer output register 1 | TO1L | TO1 | R/W | – | √ | √ | 0000H |
| F01F9H | | – | | | – | – | | |
| F01FAH | Timer output enable register 1 | TOE1L | TOE1 | R/W | √ | √ | √ | 0000H |
| F01FBH | | – | | | – | – | | |
| F01FCH | Timer output level register 1 | TOL1L | TOL1 | R/W | – | √ | √ | 0000H |
| F01FDH | | – | | | – | – | | |
| F01FEH | Timer output mode register 1 | TOM1L | TOM1 | R/W | – | √ | √ | 0000H |
| F01FFH | | – | | | – | – | | |
| F0200H | Error address storage register | ERADR | | R | – | – | √ | 0000H |
| F0201H | | | | | | | | |
| F0202H | 1-bit error detection interrupt enable register | ECCIER | | R/W | – | √ | – | 00H |
| F0203H | Bit error detection register | ECCER | | R/W | – | √ | – | 00H |
| F0204H | ECC test protect register | ECCTPR | | R/W | – | √ | – | 00H |
| F0205H | ECC test mode register | ECCTMDR | | R/W | – | √ | – | 00H |
| F0206H | Write data inversion register | ECCDWRVR | | R/W | – | – | √ | 0000H |
| F0207H | | | | | | | | |
| F0220H | Port output slew rate select register | PSRSEL | | R/W | √ | √ | – | 00H |
| F0222H | SNOOZE status output control register 0 | PSNZCNT0 | | R/W | √ | √ | – | 00H |
| F0223H | SNOOZE status output control register 1 | PSNZCNT1 | | R/W | √ | √ | – | 00H |
| F0224H | SNOOZE status output control register 2 | PSNZCNT2 | | R/W | √ | √ | – | 00H |
| F0225H | SNOOZE status output control register 3 | PSNZCNT3 | | R/W | √ | √ | – | 00H |
| F0227H | D/A converter mode register 2 | DAM2 | | R/W | √ | √ | – | 00H |
| F0228H | PWM output delay control register 0 | PWMDLY0 | | R/W | – | – | √ | 0000H |
| F0229H | | | | | | | | |
| F022AH | PWM output delay control register 1 | PWMDLY1 | | R/W | – | – | √ | 0000H |
| F022BH | | | | | | | | |
| F022CH | PWM output delay control register 2 | PWMDLY2 | | R/W | – | – | √ | 0000H |
| F022DH | | | | | | | | |
| F0230H | IICA control register 00 | IICCTL00 | | R/W | √ | √ | – | 00H |
| F0231H | IICA control register 01 | IICCTL01 | | R/W | √ | √ | – | 00H |
| F0232H | IICA low-level width setting register 0 | IICWL0 | | R/W | – | √ | – | FFH |
| F0233H | IICA high-level width setting register 1 | IICWH0 | | R/W | – | √ | – | FFH |
| F0234H | Slave address register 0 | SVA0 | | R/W | – | √ | – | 00H |

**Table 3-6. Extended SFR (2nd SFR) List (8/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|
| | | | | 1-bit | 8-bit | 16-bit | |
| F0240H | Timer RJ control register 0 | TRJCR0 | R/W | – | √ | – | 00H |
| F0241H | Timer RJ I/O control register 0 | TRJIOC0 | R/W | √ | √ | – | 00H |
| F0242H | Timer RJ mode register 0 | TRJMR0 | R/W | √ | √ | – | 00H |
| F0243H | Timer RJ event pin select register 0 | TRJISR0 | R/W | √ | √ | – | 00H |
| F0260H | Timer RD ELC register | TRDELC | R/W | √ | √ | – | 00H Note |
| F0263H | Timer RD start register | TRDSTR | R/W | – | √ | – | 0CH Note |
| F0264H | Timer RD mode register | TRDMR | R/W | √ | √ | – | 00H Note |
| F0265H | Timer RD PWM function select register | TRDPMR | R/W | √ | √ | – | 00H Note |
| F0266H | Timer RD function control register | TRDFCR | R/W | √ | √ | – | 80H Note |
| F0267H | Timer RD output master enable register 1 | TRDOER1 | R/W | √ | √ | – | FFH Note |
| F0268H | Timer RD output master enable register 2 | TRDOER2 | R/W | √ | √ | – | 00H Note |
| F0269H | Timer RD output control register | TRDOCR | R/W | √ | √ | – | 00H Note |
| F026AH | Timer RD digital filter function select register 0 | TRDDF0 | R/W | √ | √ | – | 00H Note |
| F026BH | Timer RD digital filter function select register 1 | TRDDF1 | R/W | √ | √ | – | 00H Note |
| F0270H | Timer RD control register 0 | TRDCR0 | R/W | √ | √ | – | 00H Note |
| F0271H | Timer RD I/O control register A0 | TRDIORA0 | R/W | √ | √ | – | 00H Note |
| F0272H | Timer RD I/O control register C0 | TRDIORC0 | R/W | √ | √ | – | 88H Note |
| F0273H | Timer RD status register 0 | TRDSR0 | R/W | √ | √ | – | 00H Note |
| F0274H | Timer RD interrupt enable register 0 | TRDIER0 | R/W | √ | √ | – | 00H Note |
| F0275H | Timer RD PWM function output level control register 0 | TRDPOCR0 | R/W | √ | √ | – | 00H Note |
| F0276H | Timer RD counter 0 | TRD0 | R/W | – | – | √ | 0000H Note |
| F0277H | | | | | | | |
| F0278H | Timer RD general register A0 | TRDGRA0 | R/W | – | – | √ | FFFFH Note |
| F0279H | | | | | | | |
| F027AH | Timer RD general register B0 | TRDGRB0 | R/W | – | – | √ | FFFFH Note |
| F027BH | | | | | | | |
| F0280H | Timer RD control register 1 | TRDCR1 | R/W | √ | √ | – | 00H Note |
| F0281H | Timer RD I/O control register A1 | TRDIORA1 | R/W | √ | √ | – | 00H Note |
| F0282H | Timer RD I/O control register C1 | TRDIORC1 | R/W | √ | √ | – | 88H Note |
| F0283H | Timer RD status register 1 | TRDSR1 | R/W | √ | √ | – | 00H Note |
| F0284H | Timer RD interrupt enable register 1 | TRDIER1 | R/W | √ | √ | – | 00H Note |
| F0285H | Timer RD PWM function output level control register 1 | TRDPOCR1 | R/W | √ | √ | – | 00H Note |
| F0286H | Timer RD counter 1 | TRD1 | R/W | – | – | √ | 0000H Note |
| F0287H | | | | | | | |
| F0288H | Timer RD general register A1 | TRDGRA1 | R/W | – | – | √ | FFFFH Note |
| F0289H | | | | | | | |
| F028AH | Timer RD general register B1 | TRDGRB1 | R/W | – | – | √ | FFFFH Note |
| F028BH | | | | | | | |

**Note** The timer RD SFRs are undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**Table 3-6. Extended SFR (2nd SFR) List (9/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|
| | | | | 1-bit | 8-bit | 16-bit | |
| F02A0H | Comparator control register | CMPCTL | R/W | √ | √ | – | 00H |
| F02A1H | Comparator I/O switch register | CMPSEL | R/W | √ | √ | – | 00H |
| F02A2H | Comparator output monitor register | CMPMON | R | √ | √ | – | 00H |
| F02C0H | Peripheral enable register 1 | PER1 | R/W | √ | √ | – | 00H |
| F02C1H | Peripheral enable register 2 | PER2 | R/W | √ | √ | – | 00H |
| F02C2H | CAN clock select register | CANCKSEL | R/W | √ | √ | – | 00H |
| F02C3H | LIN clock select register | LINCKSEL | R/W | √ | √ | – | 00H |
| F02C4H | Clock select register | CKSEL | R/W | √ | √ | – | 00H |
| F02C5H | PLL control register | PLLCTL | R/W | √ | √ | – | 00H |
| F02C6H | PLL status register | PLLSTS | R | – | √ | – | 00H |
| F02C7H | $f_{MP}$ clock division register | MDIV | R/W | – | √ | – | 00H |
| F02C8H | RTC clock select register | RTCCL | R/W | √ | √ | – | 00H |
| F02C9H | POR/CLM reset confirmation register | POCRES | R/W | √ | √ | – | Note 1 |
| F02CAH | STOP status output control register | STPSTC | R/W | √ | √ | – | 00H |
| F02D0H | High-speed DTC control register 0 | HDTCCR0 | R/W | √ | √ | – | 00H |
| F02D2H | High-speed DTC transfer number register 0 | HDTCCT0 | R/W | √ | √ | – | 00H |
| F02D3H | High-speed DTC transfer number reload register 0 | HDTRLD0 | R/W | √ | √ | – | 00H |
| F02D4H | High-speed DTC source address register 0 | HDTSAR0 | R/W | – | – | √ | 0000H |
| F02D5H | | | | | | | |
| F02D6H | High-speed DTC destination address register 0 | HDTDAR0 | R/W | – | – | √ | 0000H |
| F02D7H | | | | | | | |
| F02D8H | High-speed DTC control register 1 | HDTCCR1 | R/W | √ | √ | – | 00H |
| F02DAH | High-speed DTC transfer number register 1 | HDTCCT1 | R/W | √ | √ | – | 00H |
| F02DBH | High-speed DTC transfer number reload register 1 | HDTRLD1 | R/W | √ | √ | – | 00H |
| F02DCH | High-speed DTC source address register 1 | HDTSAR1 | R/W | – | – | √ | 0000H |
| F02DDH | | | | | | | |
| F02DEH | High-speed DTC destination address register 1 | HDTDAR1 | R/W | – | – | √ | 0000H |
| F02DFH | | | | | | | |
| F02E0H | DTC base address register | DTCBAR | R/W | – | √ | – | FDH |
| F02E1H | High-speed DTC channel select register 0 | SELHS0 | R/W | √ | √ | – | 3FH |
| F02E2H | High-speed DTC channel select register 1 | SELHS1 | R/W | √ | √ | – | 3FH |
| F02E8H | DTC activation enable register 0 | DTCEN0 | R/W | √ | √ | – | 00H |
| F02E9H | DTC activation enable register 1 | DTCEN1 | R/W | √ | √ | – | 00H |
| F02EAH | DTC activation enable register 2 | DTCEN2 | R/W | √ | √ | – | 00H |
| F02EBH | DTC activation enable register 3 | DTCEN3 | R/W | √ | √ | – | 00H |
| F02ECH | DTC activation enable register 4 | DTCEN4 | R/W | √ | √ | – | 00H |
| F02EDH | DTC activation enable register 5 Note 2 | DTCEN5 | R/W | √ | √ | – | 00H |
| F02F0H | Flash memory CRC control register | CRC0CTL | R/W | √ | √ | – | 00H |
| F02F2H | Flash memory CRC operation result register | PGCRCL | R/W | – | – | √ | 0000H |
| F02F3H | | | | | | | |
| F02F9H | CRC operation mode control register | CRCMD | R/W | – | √ | – | 00H |
| F02FAH | CRC data register | CRCD | R/W | – | – | √ | 0000H |
| F02FBH | | | | | | | |

**Notes 1.** When a reset source other than POR occurs, bit 0 (the POCRES0 bit) retains a value right before reset.

**2.** Only in the RL78/F14.

**Table 3-6. Extended SFR (2nd SFR) List (10/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F0300H | CAN0 bit configuration register L | C0CFGLL | C0CFGL | R/W | – | √ | √ | 0000H |
| F0301H | | C0CFGLH | | | – | √ | | |
| F0302H | CAN0 bit configuration register H | C0CFGHL | C0CFGH | R/W | – | √ | √ | 0000H |
| F0303H | | C0CFGHH | | | – | √ | | |
| F0304H | CAN0 control register L | C0CTRLL | C0CTRL | R/W | – | √ | √ | 0005H Note |
| F0305H | | C0CTRLH | | | – | √ | | |
| F0306H | CAN0 control register H | C0CTRHL | C0CTRH | R/W | – | √ | √ | 0000H |
| F0307H | | C0CTRHH | | | – | √ | | |
| F0308H | CAN0 status register L | C0STSLL | C0STSL | R | – | √ | √ | 0005H Note |
| F0309H | | C0STSLH | | | – | √ | | |
| F030AH | CAN0 status register H | C0STSHL | C0STSH | R | – | √ | √ | 0000H |
| F030BH | | C0STSHH | | | – | √ | | |
| F030CH | CAN0 error flag register L | C0ERFLLL | C0ERFLL | R/W | – | √ | √ | 0000H |
| F030DH | | C0ERFLLH | | | – | √ | | |
| F030EH | CAN0 error flag register H | C0ERFLHL | C0ERFLH | R | – | √ | √ | 0000H |
| F030FH | | C0ERFLHH | | | – | √ | | |
| F0322H | CAN global configuration register L | GCFGLL | GCFGL | R/W | – | √ | √ | 0000H |
| F0323H | | GCFGLH | | | – | √ | | |
| F0324H | CAN global configuration register H | GCFGHL | GCFGH | R/W | – | √ | √ | 0000H |
| F0325H | | GCFGHH | | | – | √ | | |
| F0326H | CAN global control register L | GCTRLL | GCTRL | R/W | – | √ | √ | 0005H Note |
| F0327H | | GCTRLH | | | – | √ | | |
| F0328H | CAN global control register H | GCTRHL | GCTRH | R/W | – | √ | √ | 0000H |
| F0329H | | GCTRHH | | | – | √ | | |
| F032AH | CAN global status register | GSTSL | GSTS | R | – | √ | √ | 000DH Note |
| F032BH | | GSTSH | | | – | √ | | |
| F032CH | CAN global error flag register | GERFLL | – | R/W | – | √ | – | 00H |
| F032EH | CAN timestamp register | GTSC | | R | – | – | √ | 0000H |
| F032FH | | | | | – | – | | |
| F0330H | CAN receive rule number configuration register | GAFLCFGL | GAFLCFG | R/W | – | √ | √ | 0000H |
| F0331H | | GAFLCFGH | | | – | √ | | |
| F0332H | CAN receive buffer number configuration register | RMNBL | RMNB | R/W | – | √ | √ | 0000H |
| F0333H | | – | | | – | – | | |
| F0334H | CAN receive buffer receive complete flag register | RMND0L | RMND0 | R/W | – | √ | √ | 0000H |
| F0335H | | RMND0H | | | – | √ | | |
| F0338H | CAN receive FIFO control register 0 | RFCC0L | RFCC0 | R/W | – | √ | √ | 0000H |
| F0339H | | RFCC0H | | | – | √ | | |
| F033AH | CAN receive FIFO control register 1 | RFCC1L | RFCC1 | R/W | – | √ | √ | 0000H |
| F033BH | | RFCC1H | | | – | √ | | |
| F0340H | CAN receive FIFO status register 0 | RFSTS0L | RFSTS0 | R/W | – | √ | √ | 0001H Note |
| F0341H | | RFSTS0H | | R | – | √ | | |
| F0342H | CAN receive FIFO status register 1 | RFSTS1L | RFSTS1 | R/W | – | √ | √ | 0001H Note |
| F0343H | | RFSTS1H | | R | – | √ | | |
| F0348H | CAN receive FIFO pointer control register 0 | RFPCTR0L | RFPCTR0 | W | – | √ | √ | 0000H |
| F0349H | | RFPCTR0H | | | – | √ | | |

**Note** When the CAN0EN bit in the PER2 register is 0, the read value is undefined.
When the CAN0EN bit in the PER2 register is 1, the read value is the initial value listed above.

**Table 3-6. Extended SFR (2nd SFR) List (11/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F034AH | CAN receive FIFO pointer control register 1 | RFPCTR1L | RFPCTR1 | W | – | √ | √ | 0000H |
| F034BH | | RFPCTR1H | | | – | √ | | |
| F0350H | CAN0 transmit/receive FIFO control register 0L | CFCCL0L | CFCCL0 | R/W | – | √ | √ | 0000H |
| F0351H | | CFCCL0H | | | – | √ | | |
| F0352H | CAN0 transmit/receive FIFO control register 0H | CFCCH0L | CFCCH0 | R/W | – | √ | √ | 0000H |
| F0353H | | CFCCH0H | | | – | √ | | |
| F0358H | CAN0 transmit/receive FIFO status register 0 | CFSTS0L | CFSTS0 | R/W | – | √ | √ | 0001H Note |
| F0359H | | CFSTS0H | | R | – | √ | | |
| F035CH | CAN0 transmit/receive FIFO pointer control register 0 | CFPCTR0L | CFPCTR0 | W | – | √ | √ | 0000H |
| F035DH | | – | | | – | – | | |
| F0360H | Receive FIFO message lost status register | RFMSTS | | R | – | √ | – | 00H |
| F0361H | CAN0 transmit//receive FIFO message lost status register | CFMSTS | | R | – | √ | – | 00H |
| F0362H | CAN receive FIFO interrupt status register | RFISTS | | R | – | √ | – | 00H |
| F0363H | CAN transmit/receive FIFO receive interrupt status register | CFISTS | | R | – | √ | – | 00H |
| F0364H | CAN0 transmit buffer control register 0 | TMC0 | | R/W | – | √ | – | 00H |
| F0365H | CAN0 transmit buffer control register 1 | TMC1 | | R/W | – | √ | – | 00H |
| F0366H | CAN0 transmit buffer control register 2 | TMC2 | | R/W | – | √ | – | 00H |
| F0367H | CAN0 transmit buffer control register 3 | TMC3 | | R/W | – | √ | – | 00H |
| F036CH | CAN0 transmit buffer status register 0 | TMSTS0 | | R/W | – | √ | – | 00H |
| F036DH | CAN0 transmit buffer status register 1 | TMSTS1 | | R/W | – | √ | – | 00H |
| F036EH | CAN0 transmit buffer status register 2 | TMSTS2 | | R/W | – | √ | – | 00H |
| F036FH | CAN0 transmit buffer status register 3 | TMSTS3 | | R/W | – | √ | – | 00H |
| F0374H | CAN0 transmit buffer transmit request status register | TMTRSTSL | TMTRSTS | R | – | √ | √ | 0000H |
| F0375H | | TMTRSTSH | | | – | √ | | |
| F0376H | CAN0 transmit buffer transmit complete status register | TMTCSTSL | TMTCSTS | R | – | √ | √ | 0000H |
| F0377H | | TMTCSTSH | | | – | √ | | |
| F0378H | CAN0 transmit buffer transmit abort status register | TMTASTSL | TMTASTS | R | – | √ | √ | 0000H |
| F0379H | | TMTASTSH | | | – | √ | | |
| F037AH | CAN0 transmit buffer interrupt enable register | TMIECL | TMIEC | R/W | – | √ | √ | 0000H |
| F037BH | | TMIECH | | | – | √ | | |
| F037CH | CAN0 transmit history buffer control register | THLCC0L | THLCC0 | R/W | – | √ | √ | 0000H |
| F037DH | | THLCC0H | | | – | √ | | |
| F0380H | CAN0 transmit history buffer status register | THLSTS0L | THLSTS0 | R/W | – | √ | √ | 0001H Note |
| F0381H | | THLSTS0H | | | – | √ | | |
| F0384H | CAN0 transmit history buffer pointer control register | THLPCTR0L | THLPCTR0 | W | – | √ | √ | 0000H |
| F0385H | | THLPCTR0H | | | – | √ | | |
| F0388H | CAN global transmit interrupt status register | GTINTSTSL | GTINTSTS | R | – | √ | √ | 0000H |
| F0389H | | GTINTSTSH | | | – | √ | | |
| F038AH | CAN global RAM window control register | GRWCRL | GRWCR | R/W | – | √ | √ | 0000H |
| F038BH | | GRWCRH | | | – | √ | | |
| F038CH | CAN global test configuration register | GTSTCFGL | GTSTCFG | R/W | – | √ | √ | 0000H |
| F038DH | | GTSTCFGH | | | – | √ | | |

**Note** When the CAN0EN bit in the PER2 register is 0, the read value is undefined.
When the CAN0EN bit in the PER2 register is 1, the read value is the initial value listed above.

**Table 3-6. Extended SFR (2nd SFR) List (12/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F038EH | CAN global test control register | GTSTCTRL | – | R/W | – | √ | – | 00H |
| F038FH | | – | | | – | – | | |
| F0394H | CAN global test protection unlock register | GLOCKK | | W | – | – | √ | 0000H |
| F0395H | | | | | – | – | | |
| F03A0H | CAN receive rule entry register 0AL Note 1 | GAFLIDL0L | GAFLIDL0 | R/W | – | √ | √ | 0000H |
| F03A1H | | GAFLIDL0H | | | – | √ | | |
| F03A0H | CAN receive buffer register 0AL Note 2 | RMIDL0L | RMIDL0 | R | – | √ | √ | 0000H |
| F03A1H | | RMIDL0H | | | – | √ | | |
| F03A2H | CAN receive rule entry register 0AH Note 1 | GAFLIDH0L | GAFLIDH0 | R/W | – | √ | √ | 0000H |
| F03A3H | | GAFLIDH0H | | | – | √ | | |
| F03A2H | CAN receive buffer register 0AH Note 2 | RMIDH0L | RMIDH0 | R | – | √ | √ | 0000H |
| F03A3H | | RMIDH0H | | | – | √ | | |
| F03A4H | CAN receive rule entry register 0BL Note 1 | GAFLML0L | GAFLML0 | R/W | – | √ | √ | 0000H |
| F03A5H | | GAFLML0H | | | – | √ | | |
| F03A4H | CAN receive buffer register 0BL Note 2 | RMTS0L | RMTS0 | R | – | √ | √ | 0000H |
| F03A5H | | RMTS0H | | | – | √ | | |
| F03A6H | CAN receive rule entry register 0BH Note 1 | GAFLMH0L | GAFLMH0 | R/W | – | √ | √ | 0000H |
| F03A7H | | GAFLMH0H | | | – | √ | | |
| F03A6H | CAN receive buffer register 0BH Note 2 | RMPTR0L | RMPTR0 | R | – | √ | √ | 0000H |
| F03A7H | | RMPTR0H | | | – | √ | | |
| F03A8H | CAN receive rule entry register 0CL Note 1 | GAFLPL0L | GAFLPL0 | R/W | – | √ | √ | 0000H |
| F03A9H | | GAFLPL0H | | | – | √ | | |
| F03A8H | CAN receive buffer register 0CL Note 2 | RMDF00L | RMDF00 | R | – | √ | √ | 0000H |
| F03A9H | | RMDF00H | | | – | √ | | |
| F03AAH | CAN receive rule entry register 0CH Note 1 | GAFLPH0L | GAFLPH0 | R/W | – | √ | √ | 0000H |
| F03ABH | | GAFLPH0H | | | – | √ | | |
| F03AAH | CAN receive buffer register 0CH Note 2 | RMDF10L | RMDF10 | R | – | √ | √ | 0000H |
| F03ABH | | RMDF10H | | | – | √ | | |
| F03ACH | CAN receive rule entry register 1AL Note 1 | GAFLIDL1L | GAFLIDL1 | R/W | – | √ | √ | 0000H |
| F03ADH | | GAFLIDL1H | | | – | √ | | |
| F03ACH | CAN receive buffer register 0DL Note 2 | RMDF20L | RMDF20 | R | – | √ | √ | 0000H |
| F03ADH | | RMDF20H | | | – | √ | | |
| F03AEH | CAN receive rule entry register 1AH Note 1 | GAFLIDH1L | GAFLIDH1 | R/W | – | √ | √ | 0000H |
| F03AFH | | GAFLIDH1H | | | – | √ | | |
| F03AEH | CAN receive buffer register 0DH Note 2 | RMDF30L | RMDF30 | R | – | √ | √ | 0000H |
| F03AFH | | RMDF30H | | | – | √ | | |
| F03B0H | CAN receive rule entry register 1BL Note 1 | GAFLML1L | GAFLML1 | R/W | – | √ | √ | 0000H |
| F03B1H | | GAFLML1H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (13/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F03B0H | CAN receive buffer register 1AL Note 2 | RMIDL1L | RMIDL1 | R | – | √ | √ | 0000H |
| F03B1H | | RMIDL1H | | | – | √ | | |
| F03B2H | CAN receive rule entry register 1BH Note 1 | GAFLMH1L | GAFLMH1 | R/W | – | √ | √ | 0000H |
| F03B3H | | GAFLMH1H | | | – | √ | | |
| F03B2H | CAN receive buffer register 1AH Note 2 | RMIDH1L | RMIDH1 | R | – | √ | √ | 0000H |
| F03B3H | | RMIDH1H | | | – | √ | | |
| F03B4H | CAN receive rule entry register 1CL Note 1 | GAFLPL1L | GAFLPL1 | R/W | – | √ | √ | 0000H |
| F03B5H | | GAFLPL1H | | | – | √ | | |
| F03B4H | CAN receive buffer register 1BL Note 2 | RMTS1L | RMTS1 | R | – | √ | √ | 0000H |
| F03B5H | | RMTS1H | | | – | √ | | |
| F03B6H | CAN receive rule entry register 1CH Note 1 | GAFLPH1L | GAFLPH1 | R/W | – | √ | √ | 0000H |
| F03B7H | | GAFLPH1H | | | – | √ | | |
| F03B6H | CAN receive buffer register 1BH Note 2 | RMPTR1L | RMPTR1 | R | – | √ | √ | 0000H |
| F03B7H | | RMPTR1H | | | – | √ | | |
| F03B8H | CAN receive rule entry register 2AL Note 1 | GAFLIDL2L | GAFLIDL2 | R/W | – | √ | √ | 0000H |
| F03B9H | | GAFLIDL2H | | | – | √ | | |
| F03B8H | CAN receive buffer register 1CL Note 2 | RMDF01L | RMDF01 | R | – | √ | √ | 0000H |
| F03B9H | | RMDF01H | | | – | √ | | |
| F03BAH | CAN receive rule entry register 2AH Note 1 | GAFLIDH2L | GAFLIDH2 | R/W | – | √ | √ | 0000H |
| F03BBH | | GAFLIDH2H | | | – | √ | | |
| F03BAH | CAN receive buffer register 1CH Note 2 | RMDF11L | RMDF11 | R | – | √ | √ | 0000H |
| F03BBH | | RMDF11H | | | – | √ | | |
| F03BCH | CAN receive rule entry register 2BL Note 1 | GAFLML2L | GAFLML2 | R/W | – | √ | √ | 0000H |
| F03BDH | | GAFLML2H | | | – | √ | | |
| F03BCH | CAN receive buffer register 1DL Note 2 | RMDF21L | RMDF21 | R | – | √ | √ | 0000H |
| F03BDH | | RMDF21H | | | – | √ | | |
| F03BEH | CAN receive rule entry register 2BH Note 1 | GAFLMH2L | GAFLMH2 | R/W | – | √ | √ | 0000H |
| F03BFH | | GAFLMH2H | | | – | √ | | |
| F03BEH | CAN receive buffer register 1DH Note 2 | RMDF31L | RMDF31 | R | – | √ | √ | 0000H |
| F03BFH | | RMDF31H | | | – | √ | | |
| F03C0H | CAN receive rule entry register 2CL Note 1 | GAFLPL2L | GAFLPL2 | R/W | – | √ | √ | 0000H |
| F03C1H | | GAFLPL2H | | | – | √ | | |
| F03C0H | CAN receive buffer register 2AL Note 2 | RMIDL2L | RMIDL2 | R | – | √ | √ | 0000H |
| F03C1H | | RMIDL2H | | | – | √ | | |
| F03C2H | CAN receive rule entry register 2CH Note 1 | GAFLPH2L | GAFLPH2 | R/W | – | √ | √ | 0000H |
| F03C3H | | GAFLPH2H | | | – | √ | | |
| F03C2H | CAN receive buffer register 2AH Note 2 | RMIDH2L | RMIDH2 | R | – | √ | √ | 0000H |
| F03C3H | | RMIDH2H | | | – | √ | | |
| F03C4H | CAN receive rule entry register 3AL Note 1 | GAFLIDL3L | GAFL1DL3 | R/W | – | √ | √ | 0000H |
| F03C5H | | GAFLIDL3H | | | – | √ | | |
| F03C4H | CAN receive buffer register 2BL Note 2 | RMTS2L | RMTS2 | R | – | √ | √ | 0000H |
| F03C5H | | RMTS2H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (14/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F03C6H | CAN receive rule entry register 3AH Note 1 | GAFLIDH3L | GAFLIDH3 | R/W | – | √ | √ | 0000H |
| F03C7H | | GAFLIDH3H | | | – | √ | | |
| F03C6H | CAN receive buffer register 2BH Note 2 | RMPTR2L | RMPTR2 | R | – | √ | √ | 0000H |
| F03C7H | | RMPTR2H | | | – | √ | | |
| F03C8H | CAN receive rule entry register 3BL Note 1 | GAFLML3L | GAFLML3 | R/W | – | √ | √ | 0000H |
| F03C9H | | GAFLML3H | | | – | √ | | |
| F03C8H | CAN receive buffer register 2CL Note 2 | RMDF02L | RMDF02 | R | – | √ | √ | 0000H |
| F03C9H | | RMDF02H | | | – | √ | | |
| F03CAH | CAN receive rule entry register 3BH Note 1 | GAFLMH3L | GAFLMH3 | R/W | – | √ | √ | 0000H |
| F03CBH | | GAFLMH3H | | | – | √ | | |
| F03CAH | CAN receive buffer register 2CH Note 2 | RMDF12L | RMDF12 | R | – | √ | √ | 0000H |
| F03CBH | | RMDF12H | | | – | √ | | |
| F03CCH | CAN receive rule entry register 3CL Note 1 | GAFLPL3L | GAFLPL3 | R/W | – | √ | √ | 0000H |
| F03CDH | | GAFLPL3H | | | – | √ | | |
| F03CCH | CAN receive buffer register 2DL Note 2 | RMDF22L | RMDF22 | R | – | √ | √ | 0000H |
| F03CDH | | RMDF22H | | | – | √ | | |
| F03CEH | CAN receive rule entry register 3CH Note 1 | GAFLPH3L | GAFLPH3 | R/W | – | √ | √ | 0000H |
| F03CFH | | GAFLPH3H | | | – | √ | | |
| F03CEH | CAN receive buffer register 2DH Note 2 | RMDF32L | RMDF32 | R | – | √ | √ | 0000H |
| F03CFH | | RMDF32H | | | – | √ | | |
| F03D0H | CAN receive rule entry register 4AL Note 1 | GAFLIDL4L | GAFLIDL4 | R/W | – | √ | √ | 0000H |
| F03D1H | | GAFLIDL4H | | | – | √ | | |
| F03D0H | CAN receive buffer register 3AL Note 2 | RMIDL3L | RMIDL3 | R | – | √ | √ | 0000H |
| F03D1H | | RMIDL3H | | | – | √ | | |
| F03D2H | CAN receive rule entry register 4AH Note 1 | GAFLIDH4L | GAFLIDH4 | R/W | – | √ | √ | 0000H |
| F03D3H | | GAFLIDH4H | | | – | √ | | |
| F03D2H | CAN receive buffer register 3AH Note 2 | RMIDH3L | RMIDH3 | R | – | √ | √ | 0000H |
| F03D3H | | RMIDH3H | | | – | √ | | |
| F03D4H | CAN receive rule entry register 4BL Note 1 | GAFLML4L | GAFLML4 | R/W | – | √ | √ | 0000H |
| F03D5H | | GAFLML4H | | | – | √ | | |
| F03D4H | CAN receive buffer register 3BL Note 2 | RMTS3L | RMTS3 | R | – | √ | √ | 0000H |
| F03D5H | | RMTS3H | | | – | √ | | |
| F03D6H | CAN receive rule entry register 4BH Note 1 | GAFLMH4L | GAFLMH4 | R/W | – | √ | √ | 0000H |
| F03D7H | | GAFLMH4H | | | – | √ | | |
| F03D6H | CAN receive buffer register 3BH Note 2 | RMPTR3L | RMPTR3 | R | – | √ | √ | 0000H |
| F03D7H | | RMPTR3H | | | – | √ | | |
| F03D8H | CAN receive rule entry register 4CL Note 1 | GAFLPL4L | GAFLPL4 | R/W | – | √ | √ | 0000H |
| F03D9H | | GAFLPL4H | | | – | √ | | |
| F03D8H | CAN receive buffer register 3CL Note 2 | RMDF03L | RMDF03 | R | – | √ | √ | 0000H |
| F03D9H | | RMDF03H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (15/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F03DAH | CAN receive rule entry register 4CH Note 1 | GAFLPH4L | GAFLPH4 | R/W | – | √ | √ | 0000H |
| F03DBH | | GAFLPH4H | | | – | √ | | |
| F03DAH | CAN receive buffer register 3CH Note 2 | RMDF13L | RMDF13 | R | – | √ | √ | 0000H |
| F03DBH | | RMDF13H | | | – | √ | | |
| F03DCH | CAN receive rule entry register 5AL Note 1 | GAFLIDL5L | GAFLIDL5 | R/W | – | √ | √ | 0000H |
| F03DDH | | GAFLIDL5H | | | – | √ | | |
| F03DCH | CAN receive buffer register 3DL Note 2 | RMDF23L | RMDF23 | R | – | √ | √ | 0000H |
| F03DDH | | RMDF23H | | | – | √ | | |
| F03DEH | CAN receive rule entry register 5AH Note 1 | GAFLIDH5L | GAFLIDH5 | R/W | – | √ | √ | 0000H |
| F03DFH | | GAFLIDH5H | | | – | √ | | |
| F03DEH | CAN receive buffer register 3DH Note 2 | RMDF33L | RMDF33 | R | – | √ | √ | 0000H |
| F03DFH | | RMDF33H | | | – | √ | | |
| F03E0H | CAN receive rule entry register 5BL Note 1 | GAFLML5L | GAFLML5 | R/W | – | √ | √ | 0000H |
| F03E1H | | GAFLML5H | | | – | √ | | |
| F03E0H | CAN receive buffer register 4AL Note 2 | RMIDL4L | RMIDL4 | R | – | √ | √ | 0000H |
| F03E1H | | RMIDL4H | | | – | √ | | |
| F03E2H | CAN receive rule entry register 5BH Note 1 | GAFLMH5L | GAFLMH5 | R/W | – | √ | √ | 0000H |
| F03E3H | | GAFLMH5H | | | – | √ | | |
| F03E2H | CAN receive buffer register 4AH Note 2 | RMIDH4L | RMIDH4 | R | – | √ | √ | 0000H |
| F03E3H | | RMIDH4H | | | – | √ | | |
| F03E4H | CAN receive rule entry register 5CL Note 1 | GAFLPL5L | GAFLPL5 | R/W | – | √ | √ | 0000H |
| F03E5H | | GAFLPL5H | | | – | √ | | |
| F03E4H | CAN receive buffer register 4BL Note 2 | RMTS4L | RMTS4 | R | – | √ | √ | 0000H |
| F03E5H | | RMTS4H | | | – | √ | | |
| F03E6H | CAN receive rule entry register 5CH Note 1 | GAFLPH5L | GAFLPH5 | R/W | – | √ | √ | 0000H |
| F03E7H | | GAFLPH5H | | | – | √ | | |
| F03E6H | CAN receive buffer register 4BH Note 2 | RMPTR4L | RMPTR4 | R | – | √ | √ | 0000H |
| F03E7H | | RMPTR4H | | | – | √ | | |
| F03E8H | CAN receive rule entry register 6AL Note 1 | GAFLIDL6L | GAFLIDL6 | R/W | – | √ | √ | 0000H |
| F03E9H | | GAFLIDL6H | | | – | √ | | |
| F03E8H | CAN receive buffer register 4CL Note 2 | RMDF04L | RMDF04 | R | – | √ | √ | 0000H |
| F03E9H | | RMDF04H | | | – | √ | | |
| F03EAH | CAN receive rule entry register 6AH Note 1 | GAFLIDH6L | GAFLIDH6 | R/W | – | √ | √ | 0000H |
| F03EBH | | GAFLIDH6H | | | – | √ | | |
| F03EAH | CAN receive buffer register 4CH Note 2 | RMDF14L | RMDF14 | R | – | √ | √ | 0000H |
| F03EBH | | RMDF14H | | | – | √ | | |
| F03ECH | CAN receive rule entry register 6BL Note 1 | GAFLML6L | GAFLML6 | R/W | – | √ | √ | 0000H |
| F03EDH | | GAFLML6H | | | – | √ | | |
| F03ECH | CAN receive buffer register 4DL Note 2 | RMDF24L | RMDF24 | R | – | √ | √ | 0000H |
| F03EDH | | RMDF24H | | | – | √ | | |
| F03EEH | CAN receive rule entry register 6BH Note 1 | GAFLMH6L | GAFLMH6 | R/W | – | √ | √ | 0000H |
| F03EFH | | GAFLMH6H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (16/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F03EEH | CAN receive buffer register 4DH Note 2 | RMDF34L | RMDF34 | R | – | √ | √ | 0000H |
| F03EFH | | RMDF34H | | | – | √ | | |
| F03F0H | CAN receive rule entry register 6CL Note 1 | GAFLPL6L | GAFLPL6 | R/W | – | √ | √ | 0000H |
| F03F1H | | GAFLPL6H | | | – | √ | | |
| F03F0H | CAN receive buffer register 5AL Note 2 | RMIDL5L | RMIDL5 | R | – | √ | √ | 0000H |
| F03F1H | | RMIDL5H | | | – | √ | | |
| F03F2H | CAN receive rule entry register 6CH Note 1 | GAFLPH6L | GAFLPH6 | R/W | – | √ | √ | 0000H |
| F03F3H | | GAFLPH6H | | | – | √ | | |
| F03F2H | CAN receive buffer register 5AH Note 2 | RMIDH5L | RMIDH5 | R | – | √ | √ | 0000H |
| F03F3H | | RMIDH5H | | | – | √ | | |
| F03F4H | CAN receive rule entry register 7AL Note 1 | GAFLIDL7L | GAFLIDL7 | R/W | – | √ | √ | 0000H |
| F03F5H | | GAFLIDL7H | | | – | √ | | |
| F03F4H | CAN receive buffer register 5BL Note 2 | RMTS5L | RMTS5 | R | – | √ | √ | 0000H |
| F03F5H | | RMTS5H | | | – | √ | | |
| F03F6H | CAN receive rule entry register 7AH Note 1 | GAFLIDH7L | GAFLIDH7 | R/W | – | √ | √ | 0000H |
| F03F7H | | GAFLIDH7H | | | – | √ | | |
| F03F6H | CAN receive buffer register 5BH Note 2 | RMPTR5L | RMPTR5 | R | – | √ | √ | 0000H |
| F03F7H | | RMPTR5H | | | – | √ | | |
| F03F8H | CAN receive rule entry register 7BL Note 1 | GAFLML7L | GAFLML7 | R/W | – | √ | √ | 0000H |
| F03F9H | | GAFLML7H | | | – | √ | | |
| F03F8H | CAN receive buffer register 5CL Note 2 | RMDF05L | RMDF05 | R | – | √ | √ | 0000H |
| F03F9H | | RMDF05H | | | – | √ | | |
| F03FAH | CAN receive rule entry register 7BH Note 1 | GAFLMH7L | GAFLMH7 | R/W | – | √ | √ | 0000H |
| F03FBH | | GAFLMH7H | | | – | √ | | |
| F03FAH | CAN receive buffer register 5CH Note 2 | RMDF15L | RMDF15 | R | – | √ | √ | 0000H |
| F03FBH | | RMDF15H | | | – | √ | | |
| F03FCH | CAN receive rule entry register 7CL Note 1 | GAFLPL7L | GAFLPL7 | R/W | – | √ | √ | 0000H |
| F03FDH | | GAFLPL7H | | | – | √ | | |
| F03FCH | CAN receive buffer register 5DL Note 2 | RMDF25L | RMDF25 | R | – | √ | √ | 0000H |
| F03FDH | | RMDF25H | | | – | √ | | |
| F03FEH | CAN receive rule entry register 7CH Note 1 | GAFLPH7L | GAFLPH7 | R/W | – | √ | √ | 0000H |
| F03FFH | | GAFLPH7H | | | – | √ | | |
| F03FEH | CAN receive buffer register 5DH Note 2 | RMDF35L | RMDF35 | R | – | √ | √ | 0000H |
| F03FFH | | RMDF35H | | | – | √ | | |
| F0400H | CAN receive rule entry register 8AL Note 1 | GAFLIDL8L | GAFLIDL8 | R/W | – | √ | √ | 0000H |
| F0401H | | GAFLIDL8H | | | – | √ | | |
| F0400H | CAN receive buffer register 6AL Note 2 | RMIDL6L | RMIDL6 | R | – | √ | √ | 0000H |
| F0401H | | RMIDL6H | | | – | √ | | |
| F0402H | CAN receive rule entry register 8AH Note 1 | GAFLIDH8L | GAFLIDH8 | R/W | – | √ | √ | 0000H |
| F0403H | | GAFLIDH8H | | | – | √ | | |
| F0402H | CAN receive buffer register 6AH Note 2 | RMIDH6L | RMIDH6 | R | – | √ | √ | 0000H |
| F0403H | | RMIDH6H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (17/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F0404H | CAN receive rule entry register 8BL Note 1 | GAFLML8L | GAFLML8 | R/W | – | √ | √ | 0000H |
| F0405H | | GAFLML8H | | | – | √ | | |
| F0404H | CAN receive buffer register 6BL Note 2 | RMTS6L | RMTS6 | R | – | √ | √ | 0000H |
| F0405H | | RMTS6H | | | – | √ | | |
| F0406H | CAN receive rule entry register 8BH Note 1 | GAFLMH8L | GAFLMH8 | R/W | – | √ | √ | 0000H |
| F0407H | | GAFLMH8H | | | – | √ | | |
| F0406H | CAN receive buffer register 6BH Note 2 | RMPTR6L | RMPTR6 | R | – | √ | √ | 0000H |
| F0407H | | RMPTR6H | | | – | √ | | |
| F0408H | CAN receive rule entry register 8CL Note 1 | GAFLPL8L | GAFLPL8 | R/W | – | √ | √ | 0000H |
| F0409H | | GAFLPL8H | | | – | √ | | |
| F0408H | CAN receive buffer register 6CL Note 2 | RMDF06L | RMDF06 | R | – | √ | √ | 0000H |
| F0409H | | RMDF06H | | | – | √ | | |
| F040AH | CAN receive rule entry register 8CH Note 1 | GAFLPH8L | GAFLPH8 | R/W | – | √ | √ | 0000H |
| F040BH | | GAFLPH8H | | | – | √ | | |
| F040AH | CAN receive buffer register 6CH Note 2 | RMDF16L | RMDF16 | R | – | √ | √ | 0000H |
| F040BH | | RMDF16H | | | – | √ | | |
| F040CH | CAN receive rule entry register 9AL Note 1 | GAFLIDL9L | GAFLIDL9 | R/W | – | √ | √ | 0000H |
| F040DH | | GAFLIDL9H | | | – | √ | | |
| F040CH | CAN receive buffer register 6DL Note 2 | RMDF26L | RMDF26 | R | – | √ | √ | 0000H |
| F040DH | | RMDF26H | | | – | √ | | |
| F040EH | CAN receive rule entry register 9AH Note 1 | GAFLIDH9L | GAFLIDH9 | R/W | – | √ | √ | 0000H |
| F040FH | | GAFLIDH9H | | | – | √ | | |
| F040EH | CAN receive buffer register 6DH Note 2 | RMDF36L | RMDF36 | R | – | √ | √ | 0000H |
| F040FH | | RMDF36H | | | – | √ | | |
| F0410H | CAN receive rule entry register 9BL Note 1 | GAFLML9L | GAFLML9 | R/W | – | √ | √ | 0000H |
| F0411H | | GAFLML9H | | | – | √ | | |
| F0410H | CAN receive buffer register 7AL Note 2 | RMIDL7L | RMIDL7 | R | – | √ | √ | 0000H |
| F0411H | | RMIDL7H | | | – | √ | | |
| F0412H | CAN receive rule entry register 9BH Note 1 | GAFLMH9L | GAFLMH9 | R/W | – | √ | √ | 0000H |
| F0413H | | GAFLMH9H | | | – | √ | | |
| F0412H | CAN receive buffer register 7AH Note 2 | RMIDH7L | RMIDH7 | R | – | √ | √ | 0000H |
| F0413H | | RMIDH7H | | | – | √ | | |
| F0414H | CAN receive rule entry register 9CL Note 1 | GAFLPL9L | GAFLPL9 | R/W | – | √ | √ | 0000H |
| F0415H | | GAFLPL9H | | | – | √ | | |
| F0414H | CAN receive buffer register 7BL Note 2 | RMTS7L | RMTS7 | R | – | √ | √ | 0000H |
| F0415H | | RMTS7H | | | – | √ | | |
| F0416H | CAN receive rule entry register 9CH Note 1 | GAFLPH9L | GAFLPH9 | R/W | – | √ | √ | 0000H |
| F0417H | | GAFLPH9H | | | – | √ | | |
| F0416H | CAN receive buffer register 7BH Note 2 | RMPTR7L | RMPTR7 | R | – | √ | √ | 0000H |
| F0417H | | RMPTR7H | | | – | √ | | |
| F0418H | CAN receive rule entry register 10AL Note 1 | GAFLIDL10L | GAFLIDL10 | R/W | – | √ | √ | 0000H |
| F0419H | | GAFLIDL10H | | | – | √ | | |
| F0418H | CAN receive buffer register 7CL Note 2 | RMDF07L | RMDF07 | R | – | √ | √ | 0000H |
| F0419H | | RMDF07H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (18/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F041AH | CAN receive rule entry register 10AH Note 1 | GAFLIDH10L | GAFLIDH10 | R/W | – | √ | √ | 0000H |
| F041BH | | GAFLIDH10H | | | – | √ | | |
| F041AH | CAN receive buffer register 7CH Note 2 | RMDF17L | RMDF17 | R | – | √ | √ | 0000H |
| F041BH | | RMDF17H | | | – | √ | | |
| F041CH | CAN receive rule entry register 10BL Note 1 | GAFLML10L | GAFLML10 | R/W | – | √ | √ | 0000H |
| F041DH | | GAFLML10H | | | – | √ | | |
| F041CH | CAN receive buffer register 7DL Note 2 | RMDF27L | RMDF27 | R | – | √ | √ | 0000H |
| F041DH | | RMDF27H | | | – | √ | | |
| F041EH | CAN receive rule entry register 10BH Note 1 | GAFLMH10L | GAFLMH10 | R/W | – | √ | √ | 0000H |
| F041FH | | GAFLMH10H | | | – | √ | | |
| F041EH | CAN receive buffer register 7DH Note 2 | RMDF37L | RMDF37 | R | – | √ | √ | 0000H |
| F041FH | | RMDF37H | | | – | √ | | |
| F0420H | CAN receive rule entry register 10CL Note 1 | GAFLPL10L | GAFLPL10 | R/W | – | √ | √ | 0000H |
| F0421H | | GAFLPL10H | | | – | √ | | |
| F0420H | CAN receive buffer register 8AL Note 2 | RMIDL8L | RMIDL8 | R | – | √ | √ | 0000H |
| F0421H | | RMIDL8H | | | – | √ | | |
| F0422H | CAN receive rule entry register 10CH Note 1 | GAFLPH10L | GAFLPH10 | R/W | – | √ | √ | 0000H |
| F0423H | | GAFLPH10H | | | – | √ | | |
| F0422H | CAN receive buffer register 8AH Note 2 | RMIDH8L | RMIDH8 | R | – | √ | √ | 0000H |
| F0423H | | RMIDH8H | | | – | √ | | |
| F0424H | CAN receive rule entry register 11AL Note 1 | GAFLIDL11L | GAFLIDL11 | R/W | – | √ | √ | 0000H |
| F0425H | | GAFLIDL11H | | | – | √ | | |
| F0424H | CAN receive buffer register 8BL Note 2 | RMTS8L | RMTS8 | R | – | √ | √ | 0000H |
| F0425H | | RMTS8H | | | – | √ | | |
| F0426H | CAN receive rule entry register 11AH Note 1 | GAFLIDH11L | GAFLIDH11 | R/W | – | √ | √ | 0000H |
| F0427H | | GAFLIDH11H | | | – | √ | | |
| F0426H | CAN receive buffer register 8BH Note 2 | RMPTR8L | RMPTR8 | R | – | √ | √ | 0000H |
| F0427H | | RMPTR8H | | | – | √ | | |
| F0428H | CAN receive rule entry register 11BL Note 1 | GAFLML11L | GAFLML11 | R/W | – | √ | √ | 0000H |
| F0429H | | GAFLML11H | | | – | √ | | |
| F0428H | CAN receive buffer register 8CL Note 2 | RMDF08L | RMDF08 | R | – | √ | √ | 0000H |
| F0429H | | RMDF08H | | | – | √ | | |
| F042AH | CAN receive rule entry register 11BH Note 1 | GAFLMH11L | GAFLMH11 | R/W | – | √ | √ | 0000H |
| F042BH | | GAFLMH11H | | | – | √ | | |
| F042AH | CAN receive buffer register 8CH Note 2 | RMDF18L | RMDF18 | R | – | √ | √ | 0000H |
| F042BH | | RMDF18H | | | – | √ | | |
| F042CH | CAN receive rule entry register 11CL Note 1 | GAFLPL11L | GAFLPL11 | R/W | – | √ | √ | 0000H |
| F042DH | | GAFLPL11H | | | – | √ | | |
| F042CH | CAN receive buffer register 8DL Note 2 | RMDF28L | RMDF28 | R | – | √ | √ | 0000H |
| F042DH | | RMDF28H | | | – | √ | | |
| F042EH | CAN receive rule entry register 11CH Note 1 | GAFLPH11L | GAFLPH11 | R/W | – | √ | √ | 0000H |
| F042FH | | GAFLPH11H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (19/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F042EH | CAN receive buffer register 8DH Note 2 | RMDF38L | RMDF38 | R | – | √ | √ | 0000H |
| F042FH | | RMDF38H | | | – | √ | | |
| F0430H | CAN receive rule entry register 12AL Note 1 | GAFLIDL12L | GAFLIDL12 | R/W | – | √ | √ | 0000H |
| F0431H | | GAFLIDL12H | | | – | √ | | |
| F0430H | CAN receive buffer register 9AL Note 2 | RMIDL9L | RMIDL9 | R | – | √ | √ | 0000H |
| F0431H | | RMIDL9H | | | – | √ | | |
| F0432H | CAN receive rule entry register 12AH Note 1 | GAFLIDH12L | GAFLIDH12 | R/W | – | √ | √ | 0000H |
| F0433H | | GAFLIDH12H | | | – | √ | | |
| F0432H | CAN receive buffer register 9AH Note 2 | RMIDH9L | RMIDH9 | R | – | √ | √ | 0000H |
| F0433H | | RMIDH9H | | | – | √ | | |
| F0434H | CAN receive rule entry register 12BL Note 1 | GAFLML12L | GAFLML12 | R/W | – | √ | √ | 0000H |
| F0435H | | GAFLML12H | | | – | √ | | |
| F0434H | CAN receive buffer register 9BL Note 2 | RMTS9L | RMTS9 | R | – | √ | √ | 0000H |
| F0435H | | RMTS9H | | | – | √ | | |
| F0436H | CAN receive rule entry register 12BH Note 1 | GAFLMH12L | GAFLMH12 | R/W | – | √ | √ | 0000H |
| F0437H | | GAFLMH12H | | | – | √ | | |
| F0436H | CAN receive buffer register 9BH Note 2 | RMPTR9L | RMPTR9 | R | – | √ | √ | 0000H |
| F0437H | | RMPTR9H | | | – | √ | | |
| F0438H | CAN receive rule entry register 12CL Note 1 | GAFLPL12L | GAFLPL12 | R/W | – | √ | √ | 0000H |
| F0439H | | GAFLPL12H | | | – | √ | | |
| F0438H | CAN receive buffer register 9CL Note 2 | RMDF09L | RMDF09 | R | – | √ | √ | 0000H |
| F0439H | | RMDF09H | | | – | √ | | |
| F043AH | CAN receive rule entry register 12CH Note 1 | GAFLPH12L | GAFLPH12 | R/W | – | √ | √ | 0000H |
| F043BH | | GAFLPH12H | | | – | √ | | |
| F043AH | CAN receive buffer register 9CH Note 2 | RMDF19L | RMDF19 | R | – | √ | √ | 0000H |
| F043BH | | RMDF19H | | | – | √ | | |
| F043CH | CAN receive rule entry register 13AL Note 1 | GAFLIDL13L | GAFLIDL13 | R/W | – | √ | √ | 0000H |
| F043DH | | GAFLIDL13H | | | – | √ | | |
| F043CH | CAN receive buffer register 9DL Note 2 | RMDF29L | RMDF29 | R | – | √ | √ | 0000H |
| F043DH | | RMDF29H | | | – | √ | | |
| F043EH | CAN receive rule entry register 13AH Note 1 | GAFLIDH13L | GAFLIDH13 | R/W | – | √ | √ | 0000H |
| F043FH | | GAFLIDH13H | | | – | √ | | |
| F043EH | CAN receive buffer register 9DH Note 2 | RMDF39L | RMDF39 | R | – | √ | √ | 0000H |
| F043FH | | RMDF39H | | | – | √ | | |
| F0440H | CAN receive rule entry register 13BL Note 1 | GAFLML13L | GAFLML13 | R/W | – | √ | √ | 0000H |
| F0441H | | GAFLML13H | | | – | √ | | |
| F0440H | CAN receive buffer register 10AL Note 2 | RMIDL10L | RMIDL10 | R | – | √ | √ | 0000H |
| F0441H | | RMIDL10H | | | – | √ | | |
| F0442H | CAN receive rule entry register 13BH Note 1 | GAFLMH13L | GAFLMH13 | R/W | – | √ | √ | 0000H |
| F0443H | | GAFLMH13H | | | – | √ | | |
| F0442H | CAN receive buffer register 10AH Note 2 | RMIDH10L | RMIDH10 | R | – | √ | √ | 0000H |
| F0443H | | RMIDH10H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (20/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F0444H | CAN receive rule entry register 13CL Note 1 | GAFLPL13L | GAFLPL13 | R/W | – | √ | √ | 0000H |
| F0445H | | GAFLPL13H | | | – | √ | | |
| F0444H | CAN receive buffer register 10BL Note 2 | RMTS10L | RMTS10 | R | – | √ | √ | 0000H |
| F0445H | | RMTS10H | | | – | √ | | |
| F0446H | CAN receive rule entry register 13CH Note 1 | GAFLPH13L | GAFLPH13 | R/W | – | √ | √ | 0000H |
| F0447H | | GAFLPH13H | | | – | √ | | |
| F0446H | CAN receive buffer register 10BH Note 2 | RMPTR10L | RMPTR10 | R | – | √ | √ | 0000H |
| F0447H | | RMPTR10H | | | – | √ | | |
| F0448H | CAN receive rule entry register 14AL Note 1 | GAFLIDL14L | GAFLIDL14 | R/W | – | √ | √ | 0000H |
| F0449H | | GAFLIDL14H | | | – | √ | | |
| F0448H | CAN receive buffer register 10CL Note 2 | RMDF010L | RMDF010 | R | – | √ | √ | 0000H |
| F0449H | | RMDF010H | | | – | √ | | |
| F044AH | CAN receive rule entry register 14AH Note 1 | GAFLIDH14L | GAFLIDH14 | R/W | – | √ | √ | 0000H |
| F044BH | | GAFLIDH14H | | | – | √ | | |
| F044AH | CAN receive buffer register 10CH Note 2 | RMDF110L | RMDF110 | R | – | √ | √ | 0000H |
| F044BH | | RMDF110H | | | – | √ | | |
| F044CH | CAN receive rule entry register 14BL Note 1 | GAFLML14L | GAFLML14 | R/W | – | √ | √ | 0000H |
| F044DH | | GAFLML14H | | | – | √ | | |
| F044CH | CAN receive buffer register 10DL Note 2 | RMDF210L | RMDF210 | R | – | √ | √ | 0000H |
| F044DH | | RMDF210H | | | – | √ | | |
| F044EH | CAN receive rule entry register 14BH Note 1 | GAFLMH14L | GAFLMH14 | R/W | – | √ | √ | 0000H |
| F044FH | | GAFLMH14H | | | – | √ | | |
| F044EH | CAN receive buffer register 10DH Note 2 | RMDF310L | RMDF310 | R | – | √ | √ | 0000H |
| F044FH | | RMDF310H | | | – | √ | | |
| F0450H | CAN receive rule entry register 14CL Note 1 | GAFLPL14L | GAFLPL14 | R/W | – | √ | √ | 0000H |
| F0451H | | GAFLPL14H | | | – | √ | | |
| F0450H | CAN receive buffer register 11AL Note 2 | RMIDL11L | RMIDL11 | R | – | √ | √ | 0000H |
| F0451H | | RMIDL11H | | | – | √ | | |
| F0452H | CAN receive rule entry register 14CH Note 1 | GAFLPH14L | GAFLPH14 | R/W | – | √ | √ | 0000H |
| F0453H | | GAFLPH14H | | | – | √ | | |
| F0452H | CAN receive buffer register 11AH Note 2 | RMIDH11L | RMIDH11 | R | – | √ | √ | 0000H |
| F0453H | | RMIDH11H | | | – | √ | | |
| F0454H | CAN receive rule entry register 15AL Note 1 | GAFLIDL15L | GAFLIDL15 | R/W | – | √ | √ | 0000H |
| F0455H | | GAFLIDL15H | | | – | √ | | |
| F0454H | CAN receive buffer register 11BL Note 2 | RMTS11L | RMTS11 | R | – | √ | √ | 0000H |
| F0455H | | RMTS11H | | | – | √ | | |
| F0456H | CAN receive rule entry register 15AH Note 1 | GAFLIDH15L | GAFLIDH15 | R/W | – | √ | √ | 0000H |
| F0457H | | GAFLIDH15H | | | – | √ | | |
| F0456H | CAN receive buffer register 11BH Note 2 | RMPTR11L | RMPTR11 | R | – | √ | √ | 0000H |
| F0457H | | RMPTR11H | | | – | √ | | |
| F0458H | CAN receive rule entry register 15BL Note 1 | GAFLML15L | GAFLML15 | R/W | – | √ | √ | 0000H |
| F0459H | | GAFLML15H | | | – | √ | | |
| F0458H | CAN receive buffer register 11CL Note 2 | RMDF011L | RMDF011 | R | – | √ | √ | 0000H |
| F0459H | | RMDF011H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (21/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F045AH | CAN receive rule entry register 15BH Note 1 | GAFLMH15L | GAFLMH15 | R/W | – | √ | √ | 0000H |
| F045BH | | GAFLMH15H | | | – | √ | | |
| F045AH | CAN receive buffer register 11CH Note 2 | RMDF111L | RMDF111 | R | – | √ | √ | 0000H |
| F045BH | | RMDF111H | | | – | √ | | |
| F045CH | CAN receive rule entry register 15CL Note 1 | GAFLPL15L | GAFLPL15 | R/W | – | √ | √ | 0000H |
| F045DH | | GAFLPL15H | | | – | √ | | |
| F045CH | CAN receive buffer register 11DL Note 2 | RMDF211L | RMDF211 | R | – | √ | √ | 0000H |
| F045DH | | RMDF211H | | | – | √ | | |
| F045EH | CAN receive rule entry register 15CH Note 1 | GAFLPH15L | GAFLPH15 | R/W | – | √ | √ | 0000H |
| F045FH | | GAFLPH15H | | | – | √ | | |
| F045EH | CAN receive buffer register 11DH Note 2 | RMDF311L | RMDF311 | R | – | √ | √ | 0000H |
| F045FH | | RMDF311H | | | – | √ | | |
| F0460H | CAN receive buffer register 12AL Note 2 | RMIDL12L | RMIDL12 | R | – | √ | √ | 0000H |
| F0461H | | RMIDL12H | | | – | √ | | |
| F0462H | CAN receive buffer register 12AH Note 2 | RMIDH12L | RMIDH12 | R | – | √ | √ | 0000H |
| F0463H | | RMIDH12H | | | – | √ | | |
| F0464H | CAN receive buffer register 12BL Note 2 | RMTS12L | RMTS12 | R | – | √ | √ | 0000H |
| F0465H | | RMTS12H | | | – | √ | | |
| F0466H | CAN receive buffer register 12BH Note 2 | RMPTR12L | RMPTR12 | R | – | √ | √ | 0000H |
| F0467H | | RMPTR12H | | | – | √ | | |
| F0468H | CAN receive buffer register 12CL Note 2 | RMDF012L | RMDF012 | R | – | √ | √ | 0000H |
| F0469H | | RMDF012H | | | – | √ | | |
| F046AH | CAN receive buffer register 12CH Note 2 | RMDF112L | RMDF112 | R | – | √ | √ | 0000H |
| F046BH | | RMDF112H | | | – | √ | | |
| F046CH | CAN receive buffer register 12DL Note 2 | RMDF212L | RMDF212 | R | – | √ | √ | 0000H |
| F046DH | | RMDF212H | | | – | √ | | |
| F046EH | CAN receive buffer register 12DH Note 2 | RMDF312L | RMDF312 | R | – | √ | √ | 0000H |
| F046FH | | RMDF312H | | | – | √ | | |
| F0470H | CAN receive buffer register 13AL Note 2 | RMIDL13L | RMIDL13 | R | – | √ | √ | 0000H |
| F0471H | | RMIDL13H | | | – | √ | | |
| F0472H | CAN receive buffer register 13AH Note 2 | RMIDH13L | RMIDH13 | R | – | √ | √ | 0000H |
| F0473H | | RMIDH13H | | | – | √ | | |
| F0474H | CAN receive buffer register 13BL Note 2 | RMTS13L | RMTS13 | R | – | √ | √ | 0000H |
| F0475H | | RMTS13H | | | – | √ | | |
| F0476H | CAN receive buffer register 13BH Note 2 | RMPTR13L | RMPTR13 | R | – | √ | √ | 0000H |
| F0477H | | RMPTR13H | | | – | √ | | |
| F0478H | CAN receive buffer register 13CL Note 2 | RMDF013L | RMDF013 | R | – | √ | √ | 0000H |
| F0479H | | RMDF013H | | | – | √ | | |
| F047AH | CAN receive buffer register 13CH Note 2 | RMDF113L | RMDF113 | R | – | √ | √ | 0000H |
| F047BH | | RMDF113H | | | – | √ | | |
| F047CH | CAN receive buffer register 13DL Note 2 | RMDF213L | RMDF213 | R | – | √ | √ | 0000H |
| F047DH | | RMDF213H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (22/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F047EH | CAN receive buffer register 13DH Note 2 | RMDF313L | RMDF313 | R | – | √ | √ | 0000H |
| F047FH | | RMDF313H | | | – | √ | | |
| F0480H | CAN receive buffer register 14AL Note 2 | RMIDL14L | RMIDL14 | R | – | √ | √ | 0000H |
| F0481H | | RMIDL14H | | | – | √ | | |
| F0482H | CAN receive buffer register 14AH Note 2 | RMIDH14L | RMIDH14 | R | – | √ | √ | 0000H |
| F0483H | | RMIDH14H | | | – | √ | | |
| F0484H | CAN receive buffer register 14BL Note 2 | RMTS14L | RMTS14 | R | – | √ | √ | 0000H |
| F0485H | | RMTS14H | | | – | √ | | |
| F0486H | CAN receive buffer register 14BH Note 2 | RMPTR14L | RMPTR14 | R | – | √ | √ | 0000H |
| F0487H | | RMPTR14H | | | – | √ | | |
| F0488H | CAN receive buffer register 14CL Note 2 | RMDF014L | RMDF014 | R | – | √ | √ | 0000H |
| F0489H | | RMDF014H | | | – | √ | | |
| F048AH | CAN receive buffer register 14CH Note 2 | RMDF114L | RMDF114 | R | – | √ | √ | 0000H |
| F048BH | | RMDF114H | | | – | √ | | |
| F048CH | CAN receive buffer register 14DL Note 2 | RMDF214L | RMDF214 | R | – | √ | √ | 0000H |
| F048DH | | RMDF214H | | | – | √ | | |
| F048EH | CAN receive buffer register 14DH Note 2 | RMDF314L | RMDF314 | R | – | √ | √ | 0000H |
| F048FH | | RMDF314H | | | – | √ | | |
| F0490H | CAN receive buffer register 15AL Note 2 | RMIDL15L | RMIDL15 | R | – | √ | √ | 0000H |
| F0491H | | RMIDL15H | | | – | √ | | |
| F0492H | CAN receive buffer register 15AH Note 2 | RMIDH15L | RMIDH15 | R | – | √ | √ | 0000H |
| F0493H | | RMIDH15H | | | – | √ | | |
| F0494H | CAN receive buffer register 15BL Note 2 | RMTS15L | RMTS15 | R | – | √ | √ | 0000H |
| F0495H | | RMTS15H | | | – | √ | | |
| F0496H | CAN receive buffer register 15BH Note 2 | RMPTR15L | RMPTR15 | R | – | √ | √ | 0000H |
| F0497H | | RMPTR15H | | | – | √ | | |
| F0498H | CAN receive buffer register 15CL Note 2 | RMDF015L | RMDF015 | R | – | √ | √ | 0000H |
| F0499H | | RMDF015H | | | – | √ | | |
| F049AH | CAN receive buffer register 15CH Note 2 | RMDF115L | RMDF115 | R | – | √ | √ | 0000H |
| F049BH | | RMDF115H | | | – | √ | | |
| F049CH | CAN receive buffer register 15DL Note 2 | RMDF215L | RMDF215 | R | – | √ | √ | 0000H |
| F049DH | | RMDF215H | | | – | √ | | |
| F049EH | CAN receive buffer register 15DH Note 2 | RMDF315L | RMDF315 | R | – | √ | √ | 0000H |
| F049FH | | RMDF315H | | | – | √ | | |
| F0580H | CAN RAM test register 0 Note 1 | RPGACC0L | RPGACC0 | R/W | – | √ | √ | 0000H |
| F0581H | | RPGACC0H | | | – | √ | | |
| F0582H | CAN RAM test register 1 Note 1 | RPGACC1L | RPGACC1 | R/W | – | √ | √ | 0000H |
| F0583H | | RPGACC1H | | | – | √ | | |
| F0584H | CAN RAM test register 2 Note 1 | RPGACC2L | RPGACC2 | R/W | – | √ | √ | 0000H |
| F0585H | | RPGACC2H | | | – | √ | | |
| F0586H | CAN RAM test register 3 Note 1 | RPGACC3L | RPGACC3 | R/W | – | √ | √ | 0000H |
| F0587H | | RPGACC3H | | | – | √ | | |
| F0588H | CAN RAM test register 4 Note 1 | RPGACC4L | RPGACC4 | R/W | – | √ | √ | 0000H |
| F0589H | | RPGACC4H | | | – | √ | | |
| F058AH | CAN RAM test register 5 Note 1 | RPGACC5L | RPGACC5 | R/W | – | √ | √ | 0000H |
| F058BH | | RPGACC5H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (23/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F058CH | CAN RAM test register 6 Note 1 | RPGACC6L | RPGACC6 | R/W | – | √ | √ | 0000H |
| F058DH | | RPGACC6H | | | – | √ | | |
| F058EH | CAN RAM test register 7 Note 1 | RPGACC7L | RPGACC7 | R/W | – | √ | √ | 0000H |
| F058FH | | RPGACC7H | | | – | √ | | |
| F0590H | CAN RAM test register 8 Note 1 | RPGACC8L | RPGACC8 | R/W | – | √ | √ | 0000H |
| F0591H | | RPGACC8H | | | – | √ | | |
| F0592H | CAN RAM test register 9 Note 1 | RPGACC9L | RPGACC9 | R/W | – | √ | √ | 0000H |
| F0593H | | RPGACC9H | | | – | √ | | |
| F0594H | CAN RAM test register 10 Note 1 | RPGACC10L | RPGACC10 | R/W | – | √ | √ | 0000H |
| F0595H | | RPGACC10H | | | – | √ | | |
| F0596H | CAN RAM test register 11 Note 1 | RPGACC11L | RPGACC11 | R/W | – | √ | √ | 0000H |
| F0597H | | RPGACC11H | | | – | √ | | |
| F0598H | CAN RAM test register 12 Note 1 | RPGACC12L | RPGACC12 | R/W | – | √ | √ | 0000H |
| F0599H | | RPGACC12H | | | – | √ | | |
| F059AH | CAN RAM test register 13 Note 1 | RPGACC13L | RPGACC13 | R/W | – | √ | √ | 0000H |
| F059BH | | RPGACC13H | | | – | √ | | |
| F059CH | CAN RAM test register 14 Note 1 | RPGACC14L | RPGACC14 | R/W | – | √ | √ | 0000H |
| F059DH | | RPGACC14H | | | – | √ | | |
| F059EH | CAN RAM test register 15 Note 1 | RPGACC15L | RPGACC15 | R/W | – | √ | √ | 0000H |
| F059FH | | RPGACC15H | | | – | √ | | |
| F05A0H | CAN RAM test register 16 Note 1 | RPGACC16L | RPGACC16 | R/W | – | √ | √ | 0000H |
| F05A1H | | RPGACC16H | | | – | √ | | |
| F05A0H | CAN receive FIFO access register 0AL Note 2 | RFIDL0L | RFIDL0 | R | – | √ | √ | 0000H |
| F05A1H | | RFIDL0H | | | – | √ | | |
| F05A2H | CAN RAM test register 17 Note 1 | RPGACC17L | RPGACC17 | R/W | – | √ | √ | 0000H |
| F05A3H | | RPGACC17H | | | – | √ | | |
| F05A2H | CAN receive FIFO access register 0AH Note 2 | RFIDH0L | RFIDH0 | R | – | √ | √ | 0000H |
| F05A3H | | RFIDH0H | | | – | √ | | |
| F05A4H | CAN RAM test register 18 Note 1 | RPGACC18L | RPGACC18 | R/W | – | √ | √ | 0000H |
| F05A5H | | RPGACC18H | | | – | √ | | |
| F05A4H | CAN receive FIFO access register 0BL Note 2 | RFTS0L | RFTS0 | R | – | √ | √ | 0000H |
| F05A5H | | RFTS0H | | | – | √ | | |
| F05A6H | CAN RAM test register 19 Note 1 | RPGACC19L | RPGACC19 | R/W | – | √ | √ | 0000H |
| F05A7H | | RPGACC19H | | | – | √ | | |
| F05A6H | CAN receive FIFO access register 0BH Note 2 | RFPTR0L | RFPTR0 | R | – | √ | √ | 0000H |
| F05A7H | | RFPTR0H | | | – | √ | | |
| F05A8H | CAN RAM test register 20 Note 1 | RPGACC20L | RPGACC20 | R/W | – | √ | √ | 0000H |
| F05A9H | | RPGACC20H | | | – | √ | | |
| F05A8H | CAN receive FIFO access register 0CL Note 2 | RFDF00L | RFDF00 | R | – | √ | √ | 0000H |
| F05A9H | | RFDF00H | | | – | √ | | |
| F05AAH | CAN RAM test register 21 Note 1 | RPGACC21L | RPGACC21 | R/W | – | √ | √ | 0000H |
| F05ABH | | RPGACC21H | | | – | √ | | |
| F05AAH | CAN receive FIFO access register 0CH Note 2 | RFDF10L | RFDF10 | R | – | √ | √ | 0000H |
| F05ABH | | RFDF10H | | | – | √ | | |
| F05ACH | CAN RAM test register 22 Note 1 | RPGACC22L | RPGACC22 | R/W | – | √ | √ | 0000H |
| F05ADH | | RPGACC22H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (24/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F05ACH | CAN receive FIFO access register 0DL Note 2 | RFDF20L | RFDF20 | R | – | √ | √ | 0000H |
| F05ADH | | RFDF20H | | | – | √ | | |
| F05AEH | CAN RAM test register 23 Note 1 | RPGACC23L | RPGACC23 | R/W | – | √ | √ | 0000H |
| F05AFH | | RPGACC23H | | | – | √ | | |
| F05AEH | CAN receive FIFO access register 0DH Note 2 | RFDF30L | RFDF30 | R | – | √ | √ | 0000H |
| F05AFH | | RFDF30H | | | – | √ | | |
| F05B0H | CAN RAM test register 24 Note 1 | RPGACC24L | RPGACC24 | R/W | – | √ | √ | 0000H |
| F05B1H | | RPGACC24H | | | – | √ | | |
| F05B0H | CAN receive FIFO access register 1AL Note 2 | RFIDL1L | RFIDL1 | R | – | √ | √ | 0000H |
| F05B1H | | RFIDL1H | | | – | √ | | |
| F05B2H | CAN RAM test register 25 Note 1 | RPGACC25L | RPGACC25 | R/W | – | √ | √ | 0000H |
| F05B3H | | RPGACC25H | | | – | √ | | |
| F05B2H | CAN receive FIFO access register 1AH Note 2 | RFIDH1L | RFIDH1 | R | – | √ | √ | 0000H |
| F05B3H | | RFIDH1H | | | – | √ | | |
| F05B4H | CAN RAM test register 26 Note 1 | RPGACC26L | RPGACC26 | R/W | – | √ | √ | 0000H |
| F05B5H | | RPGACC26H | | | – | √ | | |
| F05B4H | CAN receive FIFO access register 1BL Note 2 | RFTS1L | RFTS1 | R | – | √ | √ | 0000H |
| F05B5H | | RFTS1H | | | – | √ | | |
| F05B6H | CAN RAM test register 27 Note 1 | RPGACC27L | RPGACC27 | R/W | – | √ | √ | 0000H |
| F05B7H | | RPGACC27H | | | – | √ | | |
| F05B6H | CAN receive FIFO access register 1BH Note 2 | RFPTR1L | RFPTR1 | R | – | √ | √ | 0000H |
| F05B7H | | RFPTR1H | | | – | √ | | |
| F05B8H | CAN RAM test register 28 Note 1 | RPGACC28L | RPGACC28 | R/W | – | √ | √ | 0000H |
| F05B9H | | RPGACC28H | | | – | √ | | |
| F05B8H | CAN receive FIFO access register 1CL Note 2 | RFDF01L | RFDF01 | R | – | √ | √ | 0000H |
| F05B9H | | RFDF01H | | | – | √ | | |
| F05BAH | CAN RAM test register 29 Note 1 | RPGACC29L | RPGACC29 | R/W | – | √ | √ | 0000H |
| F05BBH | | RPGACC29H | | | – | √ | | |
| F05BAH | CAN receive FIFO access register 1CH Note 2 | RFDF11L | RFDF11 | R | – | √ | √ | 0000H |
| F05BBH | | RFDF11H | | | – | √ | | |
| F05BCH | CAN RAM test register 30 Note 1 | RPGACC30L | RPGACC30 | R/W | – | √ | √ | 0000H |
| F05BDH | | RPGACC30H | | | – | √ | | |
| F05BCH | CAN receive FIFO access register 1DL Note 2 | RFDF21L | RFDF21 | R | – | √ | √ | 0000H |
| F05BDH | | RFDF21H | | | – | √ | | |
| F05BEH | CAN RAM test register 31 Note 1 | RPGACC31L | RPGACC31 | R/W | – | √ | √ | 0000H |
| F05BFH | | RPGACC31H | | | – | √ | | |
| F05BEH | CAN receive FIFO access register 1DH Note 2 | RFDF31L | RFDF31 | R | – | √ | √ | 0000H |
| F05BFH | | RFDF31H | | | – | √ | | |
| F05C0H | CAN RAM test register 32 Note 1 | RPGACC32L | RPGACC32 | R/W | – | √ | √ | 0000H |
| F05C1H | | RPGACC32H | | | – | √ | | |
| F05C2H | CAN RAM test register 33 Note 1 | RPGACC33L | RPGACC33 | R/W | – | √ | √ | 0000H |
| F05C3H | | RPGACC33H | | | – | √ | | |
| F05C4H | CAN RAM test register 34 Note 1 | RPGACC34L | RPGACC34 | R/W | – | √ | √ | 0000H |
| F05C5H | | RPGACC34H | | | – | √ | | |
| F05C6H | CAN RAM test register 35 Note 1 | RPGACC35L | RPGACC35 | R/W | – | √ | √ | 0000H |
| F05C7H | | RPGACC35H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (25/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F05C8H | CAN RAM test register 36 Note 1 | RPGACC36L | RPGACC36 | R/W | – | √ | √ | 0000H |
| F05C9H | | RPGACC36H | | | – | √ | | |
| F05CAH | CAN RAM test register 37 Note 1 | RPGACC37L | RPGACC37 | R/W | – | √ | √ | 0000H |
| F05CBH | | RPGACC37H | | | – | √ | | |
| F05CCH | CAN RAM test register 38 Note 1 | RPGACC38L | RPGACC38 | R/W | – | √ | √ | 0000H |
| F05CDH | | RPGACC38H | | | – | √ | | |
| F05CEH | CAN RAM test register 39 Note 1 | RPGACC39L | RPGACC39 | R/W | – | √ | √ | 0000H |
| F05CFH | | RPGACC39H | | | – | √ | | |
| F05D0H | CAN RAM test register 40 Note 1 | RPGACC40L | RPGACC40 | R/W | – | √ | √ | 0000H |
| F05D1H | | RPGACC40H | | | – | √ | | |
| F05D2H | CAN RAM test register 41 Note 1 | RPGACC41L | RPGACC41 | R/W | – | √ | √ | 0000H |
| F05D3H | | RPGACC41H | | | – | √ | | |
| F05D4H | CAN RAM test register 42 Note 1 | RPGACC42L | RPGACC42 | R/W | – | √ | √ | 0000H |
| F05D5H | | RPGACC42H | | | – | √ | | |
| F05D6H | CAN RAM test register 43 Note 1 | RPGACC43L | RPGACC43 | R/W | – | √ | √ | 0000H |
| F05D7H | | RPGACC43H | | | – | √ | | |
| F05D8H | CAN RAM test register 44 Note 1 | RPGACC44L | RPGACC44 | R/W | – | √ | √ | 0000H |
| F05D9H | | RPGACC44H | | | – | √ | | |
| F05DAH | CAN RAM test register 45 Note 1 | RPGACC45L | RPGACC45 | R/W | – | √ | √ | 0000H |
| F05DBH | | RPGACC45H | | | – | √ | | |
| F05DCH | CAN RAM test register 46 Note 1 | RPGACC46L | RPGACC46 | R/W | – | √ | √ | 0000H |
| F05DDH | | RPGACC46H | | | – | √ | | |
| F05DEH | CAN RAM test register 47 Note 1 | RPGACC47L | RPGACC47 | R/W | – | √ | √ | 0000H |
| F05DFH | | RPGACC47H | | | – | √ | | |
| F05E0H | CAN RAM test register 48 Note 1 | RPGACC48L | RPGACC48 | R/W | – | √ | √ | 0000H |
| F05E1H | | RPGACC48H | | | – | √ | | |
| F05E0H | CAN0 transmit/receive FIFO access register 0AL Note 2 | CFIDL0L | CFIDL0 | R/W | – | √ | √ | 0000H |
| F05E1H | | CFIDL0H | | | – | √ | | |
| F05E2H | CAN RAM test register 49 Note 1 | RPGACC49L | RPGACC49 | R/W | – | √ | √ | 0000H |
| F05E3H | | RPGACC49H | | | – | √ | | |
| F05E2H | CAN0 transmit/receive FIFO access register 0AH Note 2 | CFIDH0L | CFIDH0 | R/W | – | √ | √ | 0000H |
| F05E3H | | CFIDH0H | | | – | √ | | |
| F05E4H | CAN RAM test register 50 Note 1 | RPGACC50L | RPGACC50 | R/W | – | √ | √ | 0000H |
| F05E5H | | RPGACC50H | | | – | √ | | |
| F05E4H | CAN0 transmit/receive FIFO access register 0BL Note 2 | CFTS0L | CFTS0 | R | – | √ | √ | 0000H |
| F05E5H | | CFTS0H | | | – | √ | | |
| F05E6H | CAN RAM test register 51 Note 1 | RPGACC51L | RPGACC51 | R/W | – | √ | √ | 0000H |
| F05E7H | | RPGACC51H | | | – | √ | | |
| F05E6H | CAN0 transmit/receive FIFO access register 0BH Note 2 | CFPTR0L | CFPTR0 | R/W | – | √ | √ | 0000H |
| F05E7H | | CFPTR0H | | | – | √ | | |
| F05E8H | CAN RAM test register 52 Note 1 | RPGACC52L | RPGACC52 | R/W | – | √ | √ | 0000H |
| F05E9H | | RPGACC52H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (26/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F05E8H | CAN0 transmit/receive FIFO access register 0CL Note 2 | CFDF00L | CFDF00 | R/W | – | √ | √ | 0000H |
| F05E9H | | CFDF00H | | | – | √ | | |
| F05EAH | CAN RAM test register 53 Note 1 | RPGACC53L | RPGACC53 | R/W | – | √ | √ | 0000H |
| F05EBH | | RPGACC53H | | | – | √ | | |
| F05EAH | CAN0 transmit/receive FIFO access register 0CH Note 2 | CFDF10L | CFDF10 | R/W | – | √ | √ | 0000H |
| F05EBH | | CFDF10H | | | – | √ | | |
| F05ECH | CAN RAM test register 54 Note 1 | RPGACC54L | RPGACC54 | R/W | – | √ | √ | 0000H |
| F05EDH | | RPGACC54H | | | – | √ | | |
| F05ECH | CAN0 transmit/receive FIFO access register 0DL Note 2 | CFDF20L | CFDF20 | R/W | – | √ | √ | 0000H |
| F05EDH | | CFDF20H | | | – | √ | | |
| F05EEH | CAN RAM test register 55 Note 1 | RPGACC55L | RPGACC55 | R/W | – | √ | √ | 0000H |
| F05EFH | | RPGACC55H | | | – | √ | | |
| F05EEH | CAN0 transmit/receive FIFO access register 0DH Note 2 | CFDF30L | CFDF30 | R/W | – | √ | √ | 0000H |
| F05EFH | | CFDF30H | | | – | √ | | |
| F05F0H | CAN RAM test register 56 Note 1 | RPGACC56L | RPGACC56 | R/W | – | √ | √ | 0000H |
| F05F1H | | RPGACC56H | | | – | √ | | |
| F05F2H | CAN RAM test register 57 Note 1 | RPGACC57L | RPGACC57 | R/W | – | √ | √ | 0000H |
| F05F3H | | RPGACC57H | | | – | √ | | |
| F05F4H | CAN RAM test register 58 Note 1 | RPGACC58L | RPGACC58 | R/W | – | √ | √ | 0000H |
| F05F5H | | RPGACC58H | | | – | √ | | |
| F05F6H | CAN RAM test register 59 Note 1 | RPGACC59L | RPGACC59 | R/W | – | √ | √ | 0000H |
| F05F7H | | RPGACC59H | | | – | √ | | |
| F05F8H | CAN RAM test register 60 Note 1 | RPGACC60L | RPGACC60 | R/W | – | √ | √ | 0000H |
| F05F9H | | RPGACC60H | | | – | √ | | |
| F05FAH | CAN RAM test register 61 Note 1 | RPGACC61L | RPGACC61 | R/W | – | √ | √ | 0000H |
| F05FBH | | RPGACC61H | | | – | √ | | |
| F05FCH | CAN RAM test register 62 Note 1 | RPGACC62L | RPGACC62 | R/W | – | √ | √ | 0000H |
| F05FDH | | RPGACC62H | | | – | √ | | |
| F05FEH | CAN RAM test register 63 Note 1 | RPGACC63L | RPGACC63 | R/W | – | √ | √ | 0000H |
| F05FFH | | RPGACC63H | | | – | √ | | |
| F0600H | CAN RAM test register 64 Note 1 | RPGACC64L | RPGACC64 | R/W | – | √ | √ | 0000H |
| F0601H | | RPGACC64H | | | – | √ | | |
| F0600H | CAN0 transmit buffer register 0AL Note 2 | TMIDL0L | TMIDL0 | R/W | – | √ | √ | 0000H |
| F0601H | | TMIDL0H | | | – | √ | | |
| F0602H | CAN RAM test register 65 Note 1 | RPGACC65L | RPGACC65 | R/W | – | √ | √ | 0000H |
| F0603H | | RPGACC65H | | | – | √ | | |
| F0602H | CAN0 transmit buffer register 0AH Note 2 | TMIDH0L | TMIDH0 | R/W | – | √ | √ | 0000H |
| F0603H | | TMIDH0H | | | – | √ | | |
| F0604H | CAN RAM test register 66 Note 1 | RPGACC66L | RPGACC66 | R/W | – | √ | √ | 0000H |
| F0605H | | RPGACC66H | | | – | √ | | |
| F0606H | CAN RAM test register 67 Note 1 | RPGACC67L | RPGACC67 | R/W | – | √ | √ | 0000H |
| F0607H | | RPGACC67H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (27/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F0606H | CAN0 transmit buffer register 0BH Note 2 | TMPTR0L | TMPTR0 | R/W | – | √ | √ | 0000H |
| F0607H | | TMPTR0H | | | – | √ | | |
| F0608H | CAN RAM test register 68 Note 1 | RPGACC68L | RPGACC68 | R/W | – | √ | √ | 0000H |
| F0609H | | RPGACC68H | | | – | √ | | |
| F0608H | CAN0 transmit buffer register 0CL Note 2 | TMDF00L | TMDF00 | R/W | – | √ | √ | 0000H |
| F0609H | | TMDF00H | | | – | √ | | |
| F060AH | CAN RAM test register 69 Note 1 | RPGACC69L | RPGACC69 | R/W | – | √ | √ | 0000H |
| F060BH | | RPGACC69H | | | – | √ | | |
| F060AH | CAN0 transmit buffer register 0CH Note 2 | TMDF10L | TMDF10 | R/W | – | √ | √ | 0000H |
| F060BH | | TMDF10H | | | – | √ | | |
| F060CH | CAN RAM test register 70 Note 1 | RPGACC70L | RPGACC70 | R/W | – | √ | √ | 0000H |
| F060DH | | RPGACC70H | | | – | √ | | |
| F060CH | CAN0 transmit buffer register 0DL Note 2 | TMDF20L | TMDF20 | R/W | – | √ | √ | 0000H |
| F060DH | | TMDF20H | | | – | √ | | |
| F060EH | CAN RAM test register 71 Note 1 | RPGACC71L | RPGACC71 | R/W | – | √ | √ | 0000H |
| F060FH | | RPGACC71H | | | – | √ | | |
| F060EH | CAN0 transmit buffer register 0DH Note 2 | TMDF30L | TMDF30 | R/W | – | √ | √ | 0000H |
| F060FH | | TMDF30H | | | – | √ | | |
| F0610H | CAN RAM test register 72 Note 1 | RPGACC72L | RPGACC72 | R/W | – | √ | √ | 0000H |
| F0611H | | RPGACC72H | | | – | √ | | |
| F0610H | CAN0 transmit buffer register 1AL Note 2 | TMIDL1L | TMIDL1 | R/W | – | √ | √ | 0000H |
| F0611H | | TMIDL1H | | | – | √ | | |
| F0612H | CAN RAM test register 73 Note 1 | RPGACC73L | RPGACC73 | R/W | – | √ | √ | 0000H |
| F0613H | | RPGACC73H | | | – | √ | | |
| F0612H | CAN0 transmit buffer register 1AH Note 2 | TMIDH1L | TMIDH1 | R/W | – | √ | √ | 0000H |
| F0613H | | TMIDH1H | | | – | √ | | |
| F0614H | CAN RAM test register 74 Note 1 | RPGACC74L | RPGACC74 | R/W | – | √ | √ | 0000H |
| F0615H | | RPGACC74H | | | – | √ | | |
| F0616H | CAN RAM test register 75 Note 1 | RPGACC75L | RPGACC75 | R/W | – | √ | √ | 0000H |
| F0617H | | RPGACC75H | | | – | √ | | |
| F0616H | CAN0 transmit buffer register 1BH Note 2 | TMPTR1L | TMPTR1 | R/W | – | √ | √ | 0000H |
| F0617H | | TMPTR1H | | | – | √ | | |
| F0618H | CAN RAM test register 76 Note 1 | RPGACC76L | RPGACC76 | R/W | – | √ | √ | 0000H |
| F0619H | | RPGACC76H | | | – | √ | | |
| F0618H | CAN0 transmit buffer register 1CL Note 2 | TMDF01L | TMDF01 | R/W | – | √ | √ | 0000H |
| F0619H | | TMDF01H | | | – | √ | | |
| F061AH | CAN RAM test register 77 Note 1 | RPGACC77L | RPGACC77 | R/W | – | √ | √ | 0000H |
| F061BH | | RPGACC77H | | | – | √ | | |
| F061AH | CAN0 transmit buffer register 1CH Note 2 | TMDF11L | TMDF11 | R/W | – | √ | √ | 0000H |
| F061BH | | TMDF11H | | | – | √ | | |
| F061CH | CAN RAM test register 78 Note 1 | RPGACC78L | RPGACC78 | R/W | – | √ | √ | 0000H |
| F061DH | | RPGACC78H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (28/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F061CH | CAN0 transmit buffer register 1DL Note 2 | TMDF21L | TMDF21 | R/W | – | √ | √ | 0000H |
| F061DH | | TMDF21H | | | – | √ | | |
| F061EH | CAN RAM test register 79 Note 1 | RPGACC79L | RPGACC79 | R/W | – | √ | √ | 0000H |
| F061FH | | RPGACC79H | | | – | √ | | |
| F061EH | CAN0 transmit buffer register 1DH Note 2 | TMDF31L | TMDF31 | R/W | – | √ | √ | 0000H |
| F061FH | | TMDF31H | | | – | √ | | |
| F0620H | CAN RAM test register 80 Note 1 | RPGACC80L | RPGACC80 | R/W | – | √ | √ | 0000H |
| F0621H | | RPGACC80H | | | – | √ | | |
| F0620H | CAN0 transmit buffer register 2AL Note 2 | TMIDL2L | TMIDL2 | R/W | – | √ | √ | 0000H |
| F0621H | | TMIDL2H | | | – | √ | | |
| F0622H | CAN RAM test register 81 Note 1 | RPGACC81L | RPGACC81 | R/W | – | √ | √ | 0000H |
| F0623H | | RPGACC81H | | | – | √ | | |
| F0622H | CAN0 transmit buffer register 2AH Note 2 | TMIDH2L | TMIDH2 | R/W | – | √ | √ | 0000H |
| F0623H | | TMIDH2H | | | – | √ | | |
| F0624H | CAN RAM test register 82 Note 1 | RPGACC82L | RPGACC82 | R/W | – | √ | √ | 0000H |
| F0625H | | RPGACC82H | | | – | √ | | |
| F0626H | CAN RAM test register 83 Note 1 | RPGACC83L | RPGACC83 | R/W | – | √ | √ | 0000H |
| F0627H | | RPGACC83H | | | – | √ | | |
| F0626H | CAN0 transmit buffer register 2BH Note 2 | TMPTR2L | TMPTR2 | R/W | – | √ | √ | 0000H |
| F0627H | | TMPTR2H | | | – | √ | | |
| F0628H | CAN RAM test register 84 Note 1 | RPGACC84L | RPGACC84 | R/W | – | √ | √ | 0000H |
| F0629H | | RPGACC84H | | | – | √ | | |
| F0628H | CAN0 transmit buffer register 2CL Note 2 | TMDF02L | TMDF02 | R/W | – | √ | √ | 0000H |
| F0629H | | TMDF02H | | | – | √ | | |
| F062AH | CAN RAM test register 85 Note 1 | RPGACC85L | RPGACC85 | R/W | – | √ | √ | 0000H |
| F062BH | | RPGACC85H | | | – | √ | | |
| F062AH | CAN0 transmit buffer register 2CH Note 2 | TMDF12L | TMDF12 | R/W | – | √ | √ | 0000H |
| F062BH | | TMDF12H | | | – | √ | | |
| F062CH | CAN RAM test register 86 Note 1 | RPGACC86L | RPGACC86 | R/W | – | √ | √ | 0000H |
| F062DH | | RPGACC86H | | | – | √ | | |
| F062CH | CAN0 transmit buffer register 2DL Note 2 | TMDF22L | TMDF22 | R/W | – | √ | √ | 0000H |
| F062DH | | TMDF22H | | | – | √ | | |
| F062EH | CAN RAM test register 87 Note 1 | RPGACC87L | RPGACC87 | R/W | – | √ | √ | 0000H |
| F062FH | | RPGACC87H | | | – | √ | | |
| F062EH | CAN0 transmit buffer register 2DH Note 2 | TMDF32L | TMDF32 | R/W | – | √ | √ | 0000H |
| F062FH | | TMDF32H | | | – | √ | | |
| F0630H | CAN RAM test register 88 Note 1 | RPGACC88L | RPGACC88 | R/W | – | √ | √ | 0000H |
| F0631H | | RPGACC88H | | | – | √ | | |
| F0630H | CAN0 transmit buffer register 3AL Note 2 | TMIDL3L | TMIDL3 | R/W | – | √ | √ | 0000H |
| F0631H | | TMIDL3H | | | – | √ | | |
| F0632H | CAN RAM test register 89 Note 1 | RPGACC89L | RPGACC89 | R/W | – | √ | √ | 0000H |
| F0633H | | RPGACC89H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (29/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F0632H | CAN0 transmit buffer register 3AH Note 2 | TMIDH3L | TMIDH3 | R/W | – | √ | √ | 0000H |
| F0633H | | TMIDH3H | | | – | √ | | |
| F0634H | CAN RAM test register 90 Note 1 | RPGACC90L | RPGACC90 | R/W | – | √ | √ | 0000H |
| F0635H | | RPGACC90H | | | – | √ | | |
| F0636H | CAN RAM test register 91 Note 1 | RPGACC91L | RPGACC91 | R/W | – | √ | √ | 0000H |
| F0637H | | RPGACC91H | | | – | √ | | |
| F0636H | CAN0 transmit buffer register 3BH Note 2 | TMPTR3L | TMPTR3 | R/W | – | √ | √ | 0000H |
| F0637H | | TMPTR3H | | | – | √ | | |
| F0638H | CAN RAM test register 92 Note 1 | RPGACC92L | RPGACC92 | R/W | – | √ | √ | 0000H |
| F0639H | | RPGACC92H | | | – | √ | | |
| F0638H | CAN0 transmit buffer register 3CL Note 2 | TMDF03L | TMDF03 | R/W | – | √ | √ | 0000H |
| F0639H | | TMDF03H | | | – | √ | | |
| F063AH | CAN RAM test register 93 Note 1 | RPGACC93L | RPGACC93 | R/W | – | √ | √ | 0000H |
| F063BH | | RPGACC93H | | | – | √ | | |
| F063AH | CAN0 transmit buffer register 3CH Note 2 | TMDF13L | TMDF13 | R/W | – | √ | √ | 0000H |
| F063BH | | TMDF13H | | | – | √ | | |
| F063CH | CAN RAM test register 94 Note 1 | RPGACC94L | RPGACC94 | R/W | – | √ | √ | 0000H |
| F063DH | | RPGACC94H | | | – | √ | | |
| F063CH | CAN0 transmit buffer register 3DL Note 2 | TMDF23L | TMDF23 | R/W | – | √ | √ | 0000H |
| F063DH | | TMDF23H | | | – | √ | | |
| F063EH | CAN RAM test register 95 Note 1 | RPGACC95L | RPGACC95 | R/W | – | √ | √ | 0000H |
| F063FH | | RPGACC95H | | | – | √ | | |
| F063EH | CAN0 transmit buffer register 3DH Note 2 | TMDF33L | TMDF33 | R/W | – | √ | √ | 0000H |
| F063FH | | TMDF33H | | | – | √ | | |
| F0640H | CAN RAM test register 96 Note 1 | RPGACC96L | RPGACC96 | R/W | – | √ | √ | 0000H |
| F0641H | | RPGACC96H | | | – | √ | | |
| F0642H | CAN RAM test register 97 Note 1 | RPGACC97L | RPGACC97 | R/W | – | √ | √ | 0000H |
| F0643H | | RPGACC97H | | | – | √ | | |
| F0644H | CAN RAM test register 98 Note 1 | RPGACC98L | RPGACC98 | R/W | – | √ | √ | 0000H |
| F0645H | | RPGACC98H | | | – | √ | | |
| F0646H | CAN RAM test register 99 Note 1 | RPGACC99L | RPGACC99 | R/W | – | √ | √ | 0000H |
| F0647H | | RPGACC99H | | | – | √ | | |
| F0648H | CAN RAM test register 100 Note 1 | RPGACC100L | RPGACC100 | R/W | – | √ | √ | 0000H |
| F0649H | | RPGACC100H | | | – | √ | | |
| F064AH | CAN RAM test register 101 Note 1 | RPGACC101L | RPGACC101 | R/W | – | √ | √ | 0000H |
| F064BH | | RPGACC101H | | | – | √ | | |
| F064CH | CAN RAM test register 102 Note 1 | RPGACC102L | RPGACC102 | R/W | – | √ | √ | 0000H |
| F064DH | | RPGACC102H | | | – | √ | | |
| F064EH | CAN RAM test register 103 Note 1 | RPGACC103L | RPGACC103 | R/W | – | √ | √ | 0000H |
| F064FH | | RPGACC103H | | | – | √ | | |
| F0650H | CAN RAM test register 104 Note 1 | RPGACC104L | RPGACC104 | R/W | – | √ | √ | 0000H |
| F0651H | | RPGACC104H | | | – | √ | | |

**Notes 1.** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (30/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F0652H | CAN RAM test register 105 Note | RPGACC105L | RPGACC105 | R/W | – | √ | √ | 0000H |
| F0653H | | RPGACC105H | | | – | √ | | |
| F0654H | CAN RAM test register 106 Note | RPGACC106L | RPGACC106 | R/W | – | √ | √ | 0000H |
| F0655H | | RPGACC106H | | | – | √ | | |
| F0656H | CAN RAM test register 107 Note | RPGACC107L | RPGACC107 | R/W | – | √ | √ | 0000H |
| F0657H | | RPGACC107H | | | – | √ | | |
| F0658H | CAN RAM test register 108 Note | RPGACC108L | RPGACC108 | R/W | – | √ | √ | 0000H |
| F0659H | | RPGACC108H | | | – | √ | | |
| F065AH | CAN RAM test register 109 Note | RPGACC109L | RPGACC109 | R/W | – | √ | √ | 0000H |
| F065BH | | RPGACC109H | | | – | √ | | |
| F065CH | CAN RAM test register 110 Note | RPGACC110L | RPGACC110 | R/W | – | √ | √ | 0000H |
| F065DH | | RPGACC110H | | | – | √ | | |
| F065EH | CAN RAM test register 111 Note | RPGACC111L | RPGACC111 | R/W | – | √ | √ | 0000H |
| F065FH | | RPGACC111H | | | – | √ | | |
| F0660H | CAN RAM test register 112 Note | RPGACC112L | RPGACC112 | R/W | – | √ | √ | 0000H |
| F0661H | | RPGACC112H | | | – | √ | | |
| F0662H | CAN RAM test register 113 Note | RPGACC113L | RPGACC113 | R/W | – | √ | √ | 0000H |
| F0663H | | RPGACC113H | | | – | √ | | |
| F0664H | CAN RAM test register 114 Note | RPGACC114L | RPGACC114 | R/W | – | √ | √ | 0000H |
| F0665H | | RPGACC114H | | | – | √ | | |
| F0666H | CAN RAM test register 115 Note | RPGACC115L | RPGACC115 | R/W | – | √ | √ | 0000H |
| F0667H | | RPGACC115H | | | – | √ | | |
| F0668H | CAN RAM test register 116 Note | RPGACC116L | RPGACC116 | R/W | – | √ | √ | 0000H |
| F0669H | | RPGACC116H | | | – | √ | | |
| F066AH | CAN RAM test register 117 Note | RPGACC117L | RPGACC117 | R/W | – | √ | √ | 0000H |
| F066BH | | RPGACC117H | | | – | √ | | |
| F066CH | CAN RAM test register 118 Note | RPGACC118L | RPGACC118 | R/W | – | √ | √ | 0000H |
| F066DH | | RPGACC118H | | | – | √ | | |
| F066EH | CAN RAM test register 119 Note | RPGACC119L | RPGACC119 | R/W | – | √ | √ | 0000H |
| F066FH | | RPGACC119H | | | – | √ | | |
| F0670H | CAN RAM test register 120 Note | RPGACC120L | RPGACC120 | R/W | – | √ | √ | 0000H |
| F0671H | | RPGACC120H | | | – | √ | | |
| F0672H | CAN RAM test register 121 Note | RPGACC121L | RPGACC121 | R/W | – | √ | √ | 0000H |
| F0673H | | RPGACC121H | | | – | √ | | |
| F0674H | CAN RAM test register 122 Note | RPGACC122L | RPGACC122 | R/W | – | √ | √ | 0000H |
| F0675H | | RPGACC122H | | | – | √ | | |
| F0676H | CAN RAM test register 123 Note | RPGACC123L | RPGACC123 | R/W | – | √ | √ | 0000H |
| F0677H | | RPGACC123H | | | – | √ | | |
| F0678H | CAN RAM test register 124 Note | RPGACC124L | RPGACC124 | R/W | – | √ | √ | 0000H |
| F0679H | | RPGACC124H | | | – | √ | | |
| F067AH | CAN RAM test register 125 Note | RPGACC125L | RPGACC125 | R/W | – | √ | √ | 0000H |
| F067BH | | RPGACC125H | | | – | √ | | |
| F067CH | CAN RAM test register 126 Note | RPGACC126L | RPGACC126 | R/W | – | √ | √ | 0000H |
| F067DH | | RPGACC126H | | | – | √ | | |
| F067EH | CAN RAM test register 127 Note | RPGACC127L | RPGACC127 | R/W | – | √ | √ | 0000H |
| F067FH | | RPGACC127H | | | – | √ | | |

**Note** These registers are allocated to the RAM window 0 for the CAN module (receive rule and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**Table 3-6. Extended SFR (2nd SFR) List (31/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F0680H | CAN0 transmit history buffer access register Note | THLACC0L | THLACC0 | R | – | √ | √ | 0000H |
| F0681H | | THLACC0H | | | – | √ | | |
| F06C1H | LIN wakeup baud rate select register | LWBR0/LWBR1 | | R/W | – | √ | – | 00H |
| F06C2H | LIN/UART baud rate prescaler 0 register | LBRP00/LBRP10 | LBRP0/ LBRP1 | R/W | – | √ | √ | 00H |
| F06C3H | LIN/UART baud rate prescaler 1 register | LBRP01/LBRP11 | | R/W | – | √ | | 00H |
| F06C4H | LIN self–test control register | LSTC0/LSTC1 | | R/W | – | √ | – | 00H |
| F06C5H | UART standby control register | LUSC0/LUSC1 | | R/W | – | √ | – | 00H |
| F06C8H | LIN/UART mode register | LMD0/LMD1 | | R/W | – | √ | – | 00H |
| F06C9H | LIN/UART break field configuration register/ UART configuration register | LBFC0/LBFC1 | | R/W | – | √ | – | 00H |
| F06CAH | LIN/UART space configuration register | LSC0/LSC1 | | R/W | – | √ | – | 00H |
| F06CBH | LIN wakeup configuration register | LWUP0/LWUP1 | | R/W | – | √ | – | 00H |
| F06CCH | LIN interrupt enable register | LIE0/LIE1 | | R/W | – | √ | – | 00H |
| F06CDH | LIN/UART error detection enable register | LEDE0/LEDE1 | | R/W | – | √ | – | 00H |
| F06CEH | LIN/UART control register | LCUC0/LCUC1 | | R/W | – | √ | – | 00H |
| F06D0H | LIN/UART transmit control register | LTRC0/LTRC1 | | R/W | – | √ | – | 00H |
| F06D1H | LIN/UART mode status register | LMST0/LMST1 | | R | – | √ | – | 00H |
| F06D2H | LIN/UART status register | LST0/LST1 | | R/W | – | √ | – | 00H |
| F06D3H | LIN/UART error status register | LEST0/LEST1 | | R/W | – | √ | – | 00H |
| F06D4H | LIN/UART data field configuration register | LDFC0/LDFC1 | | R/W | – | √ | – | 00H |
| F06D5H | LIN/UART ID buffer register | LIDB0/LIDB1 | | R/W | – | √ | – | 00H |
| F06D6H | LIN checksum buffer register | LCBR0/LCBR1 | | R/W | – | √ | – | 00H |
| F06D7H | UART data buffer 0 register | LUDB00/LUDB10 | | R/W | – | √ | – | 00H |
| F06D8H | LIN/UART data buffer 1 register | LDB01/LDB11 | | R/W | – | √ | – | 00H |
| F06D9H | LIN/UART data buffer 2 register | LDB02/LDB12 | | R/W | – | √ | – | 00H |
| F06DAH | LIN/UART data buffer 3 register | LDB03/LDB13 | | R/W | – | √ | – | 00H |
| F06DBH | LIN/UART data buffer 4 register | LDB04/LDB14 | | R/W | – | √ | – | 00H |
| F06DCH | LIN/UART data buffer 5 register | LDB05/LDB15 | | R/W | – | √ | – | 00H |
| F06DDH | LIN/UART data buffer 6 register | LDB06/LDB16 | | R/W | – | √ | – | 00H |
| F06DEH | LIN/UART data buffer 7 register | LDB07/LDB17 | | R/W | – | √ | – | 00H |
| F06DFH | LIN/UART data buffer 8 register | LDB08/LDB18 | | R/W | – | √ | – | 00H |
| F06E0H | UART operation enable register | LUOER0/LUOER1 | | R/W | – | √ | – | 00H |
| F06E1H | UART option register 1 | LUOR01/LUOR11 | | R/W | – | √ | – | 00H |
| F06E4H | UART transmit data register | LUTDR0L/ LUTDR1L | LUTDR0/ LUTDR1 | R/W | – | √ | √ | 0000H |
| F06E5H | | LUTDR0H/ LUTDR1H | | | – | √ | | |
| F06E6H | UART receive data register | LURDR0L/ LURDR1L | LURDR0/ LURDR1 | R | – | √ | √ | 0000H |
| F06E7H | | LURDR0H/ LURDR1H | | | – | √ | | |

**Note** These registers are allocated to the RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 3-6. Extended SFR (2nd SFR) List (32/32)**

| Address | Special Function Register (2nd SFR) Name | Symbol | | R/W | Manipulable Bit Range | | | After reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1-bit | 8-bit | 16-bit | |
| F06E8H | UART wait transmit data register | LUWTDR0L/ LUWTDR1L | LUWTDR0/ LUWTDR1 | R/W | – | √ | √ | 0000H |
| F06E9H | | LUWTDR0H/ LUWTDR1H | | | – | √ | | |
| F06F0H | Timer RJ counter register 0 | TRJ0 | | R/W | – | – | √ | FFFFH |
| F06F1H | | | | | | | | |
| F0780H | Event output destination select register 00 | ELSELR00 | | R/W | √ | √ | – | 00H |
| F0781H | Event output destination select register 01 | ELSELR01 | | R/W | √ | √ | – | 00H |
| F0782H | Event output destination select register 02 | ELSELR02 | | R/W | √ | √ | – | 00H |
| F0783H | Event output destination select register 03 | ELSELR03 | | R/W | √ | √ | – | 00H |
| F0784H | Event output destination select register 04 | ELSELR04 | | R/W | √ | √ | – | 00H |
| F0785H | Event output destination select register 05 | ELSELR05 | | R/W | √ | √ | – | 00H |
| F0786H | Event output destination select register 06 | ELSELR06 | | R/W | √ | √ | – | 00H |
| F0787H | Event output destination select register 07 | ELSELR07 | | R/W | √ | √ | – | 00H |
| F0788H | Event output destination select register 08 | ELSELR08 | | R/W | √ | √ | – | 00H |
| F0789H | Event output destination select register 09/ A/D converter trigger select register 0 Note | ELSELR09/ADTRGS0 Note | | R/W | √ | √ | – | 00H |
| F078AH | Event output destination select register 10 | ELSELR10 | | R/W | √ | √ | – | 00H |
| F078BH | Event output destination select register 11 | ELSELR11 | | R/W | √ | √ | – | 00H |
| F078CH | Event output destination select register 12 | ELSELR12 | | R/W | √ | √ | – | 00H |
| F078DH | Event output destination select register 13/ A/D converter trigger select register 1 Note | ELSELR13/ADTRGS1 Note | | R/W | √ | √ | – | 00H |
| F078EH | Event output destination select register 14 | ELSELR14 | | R/W | √ | √ | – | 00H |
| F078FH | Event output destination select register 15 | ELSELR15 | | R/W | √ | √ | – | 00H |
| F0790H | Event output destination select register 16 | ELSELR16 | | R/W | √ | √ | – | 00H |
| F0791H | Event output destination select register 17 | ELSELR17 | | R/W | √ | √ | – | 00H |
| F0792H | Event output destination select register 18 | ELSELR18 | | R/W | √ | √ | – | 00H |
| F0793H | Event output destination select register 19 | ELSELR19 | | R/W | √ | √ | – | 00H |
| F0794H | Event output destination select register 20 | ELSELR20 | | R/W | √ | √ | – | 00H |
| F0795H | Event output destination select register 21 | ELSELR21 | | R/W | √ | √ | – | 00H |
| F0796H | Event output destination select register 22 | ELSELR22 | | R/W | √ | √ | – | 00H |
| F0797H | Event output destination select register 23 | ELSELR23 | | R/W | √ | √ | – | 00H |
| F0798H | Event output destination select register 24 | ELSELR24 | | R/W | √ | √ | – | 00H |
| F0799H | Event output destination select register 25 | ELSELR25 | | R/W | √ | √ | – | 00H |

**Note** RL78/F13 only.

**Remark** For SFRs in the SFR area, see **Table 3-5 SFR List**.

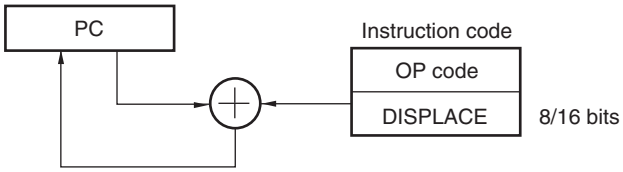
## 3.3 Instruction Address Addressing

### 3.3.1 Relative addressing

**[Function]**

Relative addressing stores in the program counter (PC) the result of adding a displacement value included in the instruction word (signed complement data: –128 to +127 or –32768 to +32767) to the program counter (PC)'s value (the start address of the next instruction), and specifies the program address to be used as the branch destination. Relative addressing is applied only to branch instructions.

**Figure 3-42. Outline of Relative Addressing**




### 3.3.2 Immediate addressing

**[Function]**

Immediate addressing stores immediate data of the instruction word in the program counter, and specifies the program address to be used as the branch destination.

For immediate addressing, CALL !!addr20 or BR !!addr20 is used to specify 20-bit addresses and CALL !addr16 or BR !addr16 is used to specify 16-bit addresses. 0000 is set to the higher 4 bits when specifying 16-bit addresses.

**Figure 3-43. Example of CALL !!addr20/BR !!addr20**

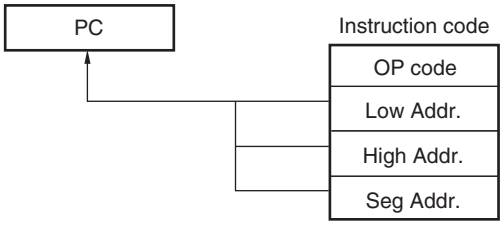
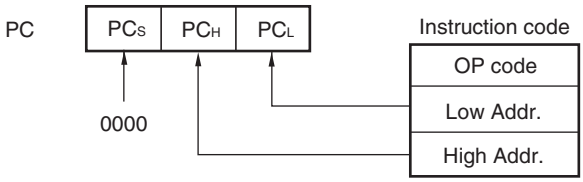



**Figure 3-44. Example of CALL !addr16/BR !addr16**



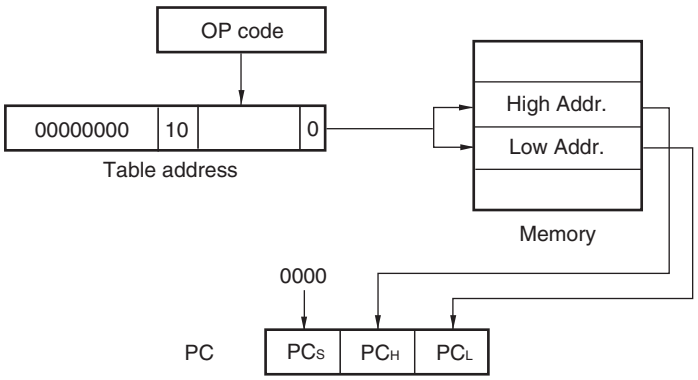
### 3.3.3 Table indirect addressing

**[Function]**

Table indirect addressing specifies a table address in the CALLT table area (0080H to 00BFH) with the 5-bit immediate data in the instruction word, stores the contents at that table address and the next address in the program counter (PC) as 16-bit data, and specifies the program address. Table indirect addressing is applied only for CALLT instructions.

In the RL78 microcontrollers, branching is enabled only to the 64 KB space from 00000H to 0FFFFH.

**Figure 3-45. Outline of Table Indirect Addressing**

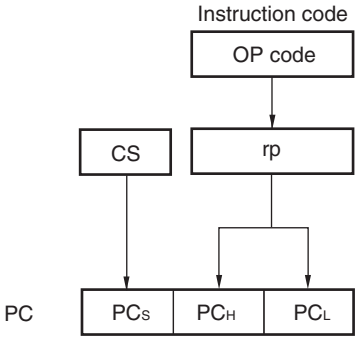


### 3.3.4 Register direct addressing

**[Function]**

Register direct addressing stores in the program counter (PC) the contents of a general-purpose register pair (AX/BC/DE/HL) and CS register of the current register bank specified with the instruction word as 20-bit data, and specifies the program address. Register direct addressing can be applied only to the CALL AX, BC, DE, HL, and BR AX instructions.

**Figure 3-46. Outline of Register Direct Addressing**



## 3.4 Addressing for Processing Data Addresses

### 3.4.1 Implied addressing

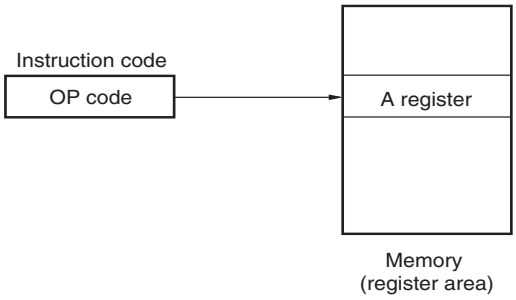
**[Function]**

Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

**[Operand format]**

Implied addressing can be applied only to MULU X.

**Figure 3-47. Outline of Implied Addressing**




### 3.4.2 Register addressing

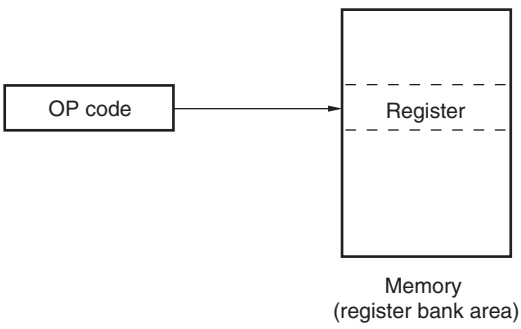
**[Function]**

Register addressing accesses a general-purpose register as an operand. The instruction word of 3-bit long is used to select an 8-bit register and the instruction word of 2-bit long is used to select a 16-bit register.

**[Operand format]**

| Identifier | Description |
|---|---|
| r | X, A, C, B, E, D, L, H |
| rp | AX, BC, DE, HL |

**Figure 3-48. Outline of Register Addressing**



### 3.4.3 Direct addressing

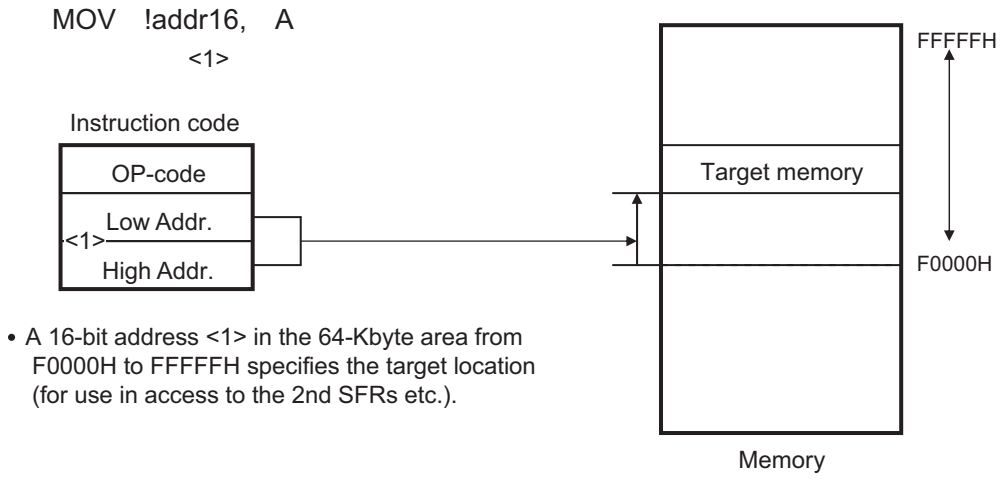
**[Function]**

Direct addressing uses immediate data in the instruction word as an operand address to directly specify the target address.

**[Operand format]**

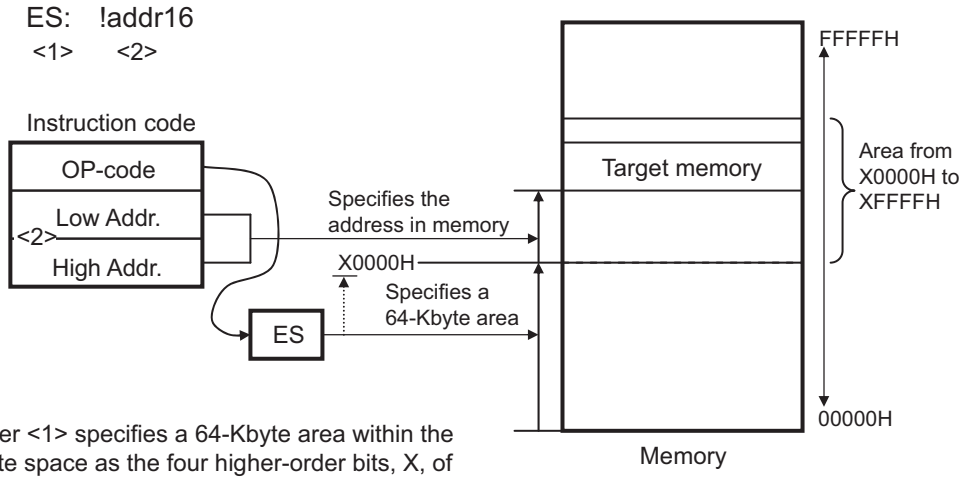
| Identifier | Description |
|---|---|
| !addr16 | Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable) |
| ES:!addr16 | Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register) |

**Figure 3-49. Example of !addr16**




- A 16-bit address <1> in the 64-Kbyte area from F0000H to FFFFFH specifies the target location (for use in access to the 2nd SFRs etc.).

**Figure 3-50. Example of ES:!addr16**




- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- A 16-bit address <2> in the area from X0000H to XFFFFH and the ES register <1> specify the target location; this is used for access to fixed data other than that in mirrored areas.

### 3.4.4 Short direct addressing

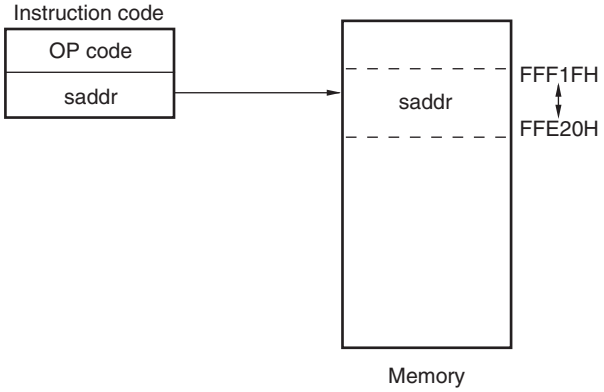
**[Function]**

Short direct addressing directly specifies the target addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFE20H to FFF1FH.

**[Operand format]**

| Identifier | Description |
|---|---|
| SADDR | Label, FFE20H to FFF1FH immediate data, or 0FE20H to 0FF1FH immediate data<br>(only the space from FFE20H to FFF1FH is specifiable) |
| SADDRP | Label, FFE20H to FFF1FH immediate data, or 0FE20H to 0FF1FH immediate data (even address only)<br>(only the space from FFE20H to FFF1FH is specifiable) |

**Figure 3-51. Outline of Short Direct Addressing**




**Remark** SADDR and SADDRP are used to describe the values of addresses FE20H to FF1FH with 16-bit immediate data (higher 4 bits of actual address are omitted), and the values of addresses FFE20H to FFF1FH with 20-bit immediate data.
Regardless of whether SADDR or SADDRP is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

### 3.4.5 SFR addressing

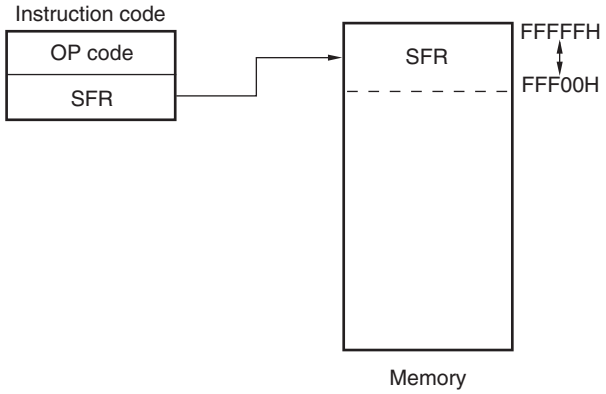
**[Function]**

SFR addressing directly specifies the target SFR addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFF00H to FFFFFH.

**[Operand format]**

| Identifier | Description |
|---|---|
| SFR | SFR name |
| SFRP | 16-bit-manipulatable SFR name (even address) |

**Figure 3-52. Outline of SFR Addressing**



### 3.4.6 Register indirect addressing

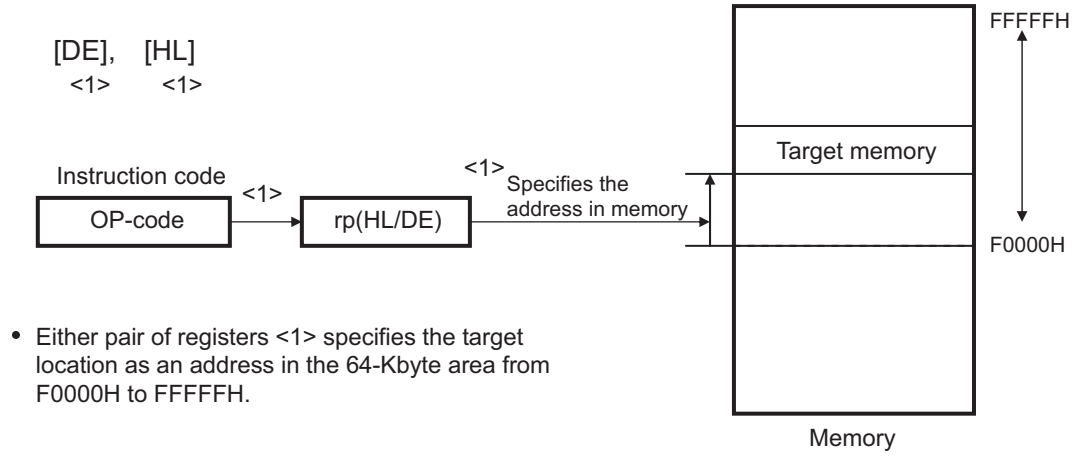
**[Function]**

Register indirect addressing directly specifies the target addresses using the contents of the register pair specified with the instruction word as an operand address.

**[Operand format]**

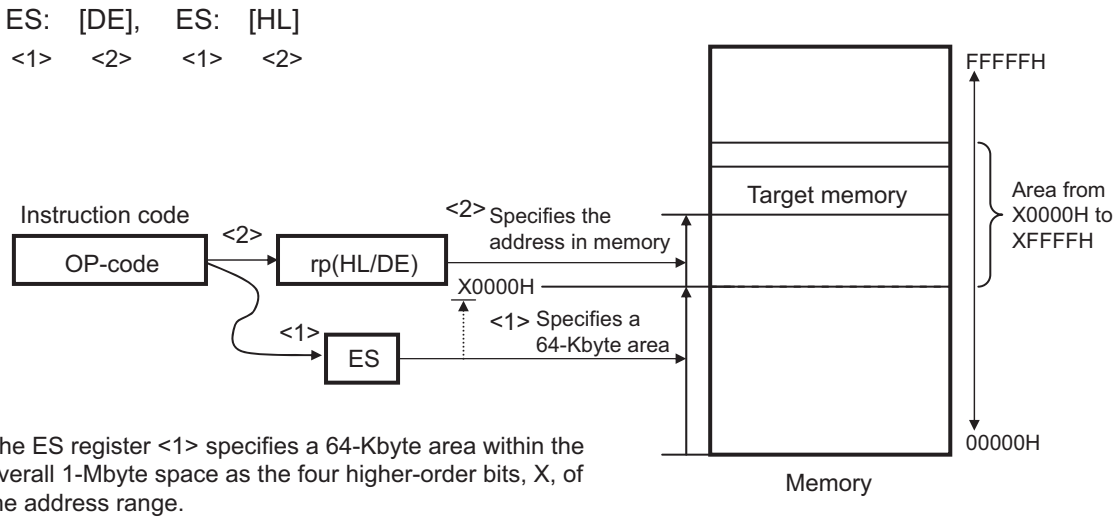
| Identifier | Description |
|---|---|
| – | [DE], [HL] (only the space from F0000H to FFFFFH is specifiable) |
| – | ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register) |

**Figure 3-53. Example of [DE], [HL]**




- Either pair of registers <1> specifies the target location as an address in the 64-Kbyte area from F0000H to FFFFFH.

**Figure 3-54. Example of ES:[DE], ES:[HL]**




- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> and the ES register <1> specify the target location in the area from X0000H to XFFFFH.

### 3.4.7 Based addressing

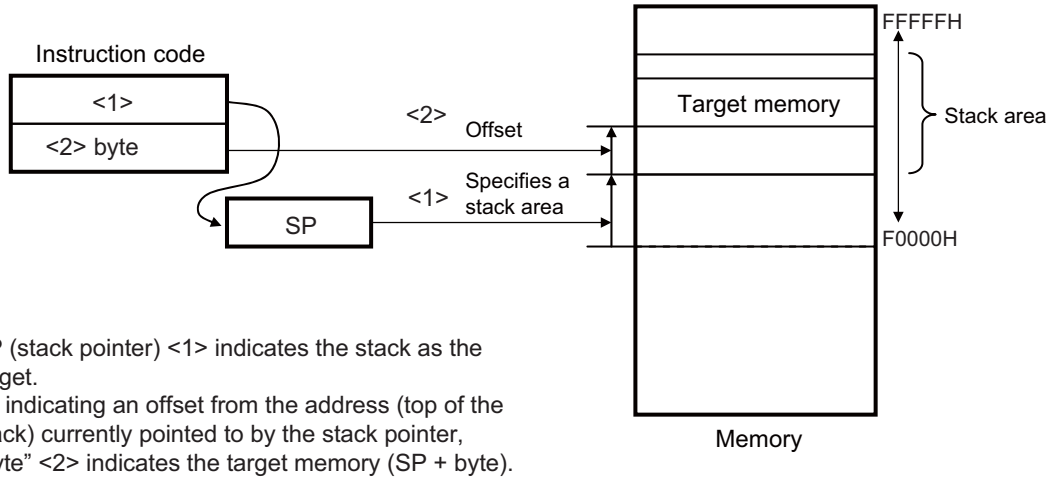
**[Function]**

Based addressing uses the contents of a register pair specified with the instruction word or 16-bit immediate data as a base address, and 8-bit immediate data or 16-bit immediate data as offset data. The sum of these values is used to specify the target address.

**[Operand format]**

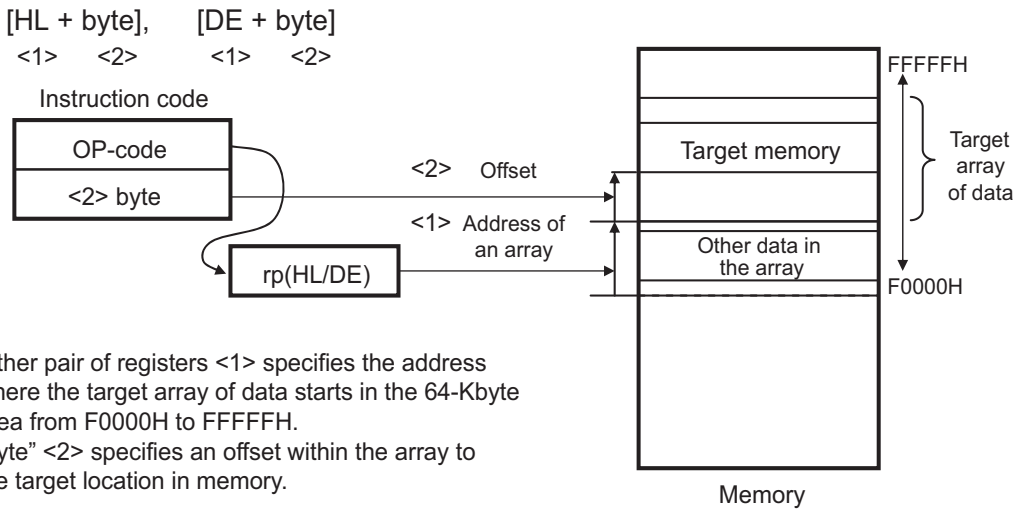
| Identifier | Description |
|---|---|
| – | [HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable) |
| – | word[B], word[C] (only the space from F0000H to FFFFFH is specifiable) |
| – | word[BC] (only the space from F0000H to FFFFFH is specifiable) |
| – | ES:[HL + byte], ES:[DE + byte] (higher 4-bit addresses are specified by the ES register) |
| – | ES:word[B], ES:word[C] (higher 4-bit addresses are specified by the ES register) |
| – | ES:word[BC] (higher 4-bit addresses are specified by the ES register) |

**Figure 3-55. Example of [SP+byte]**



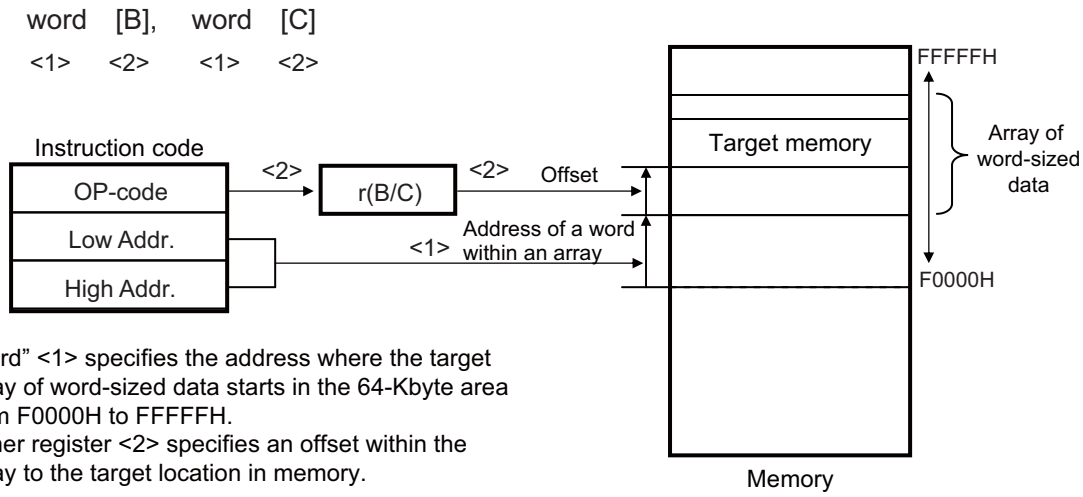

- SP (stack pointer) <1> indicates the stack as the target.
- By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, “byte” <2> indicates the target memory (SP + byte).

**Figure 3-56. Example of [HL + byte], [DE + byte]**



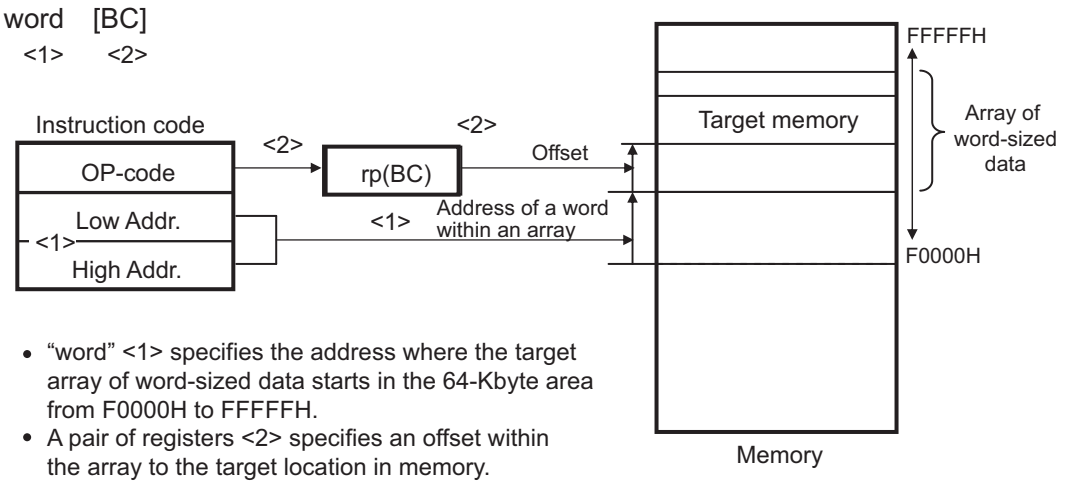

- Either pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- "byte" <2> specifies an offset within the array to the target location in memory.

**Figure 3-57. Example of word[B], word[C]**



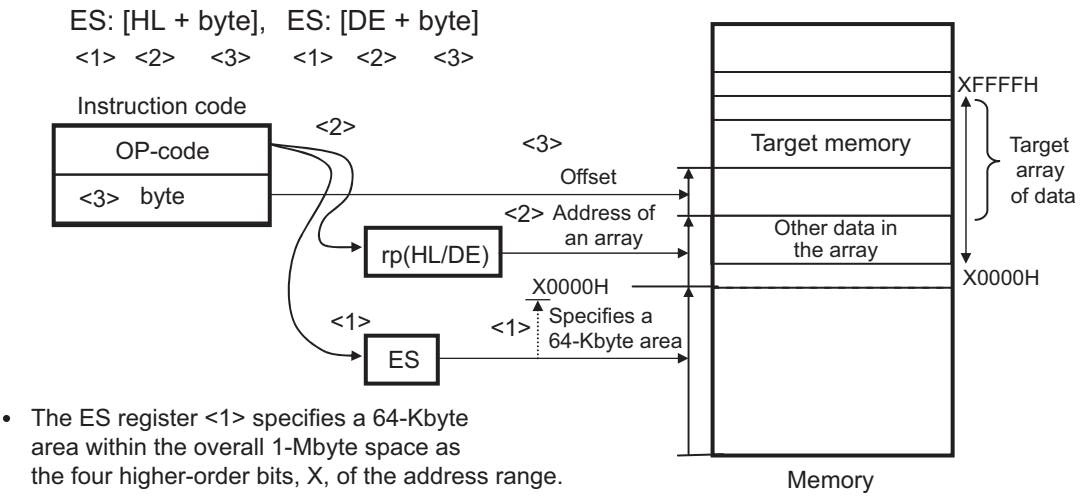

- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

**Figure 3-58. Example of word[BC]**



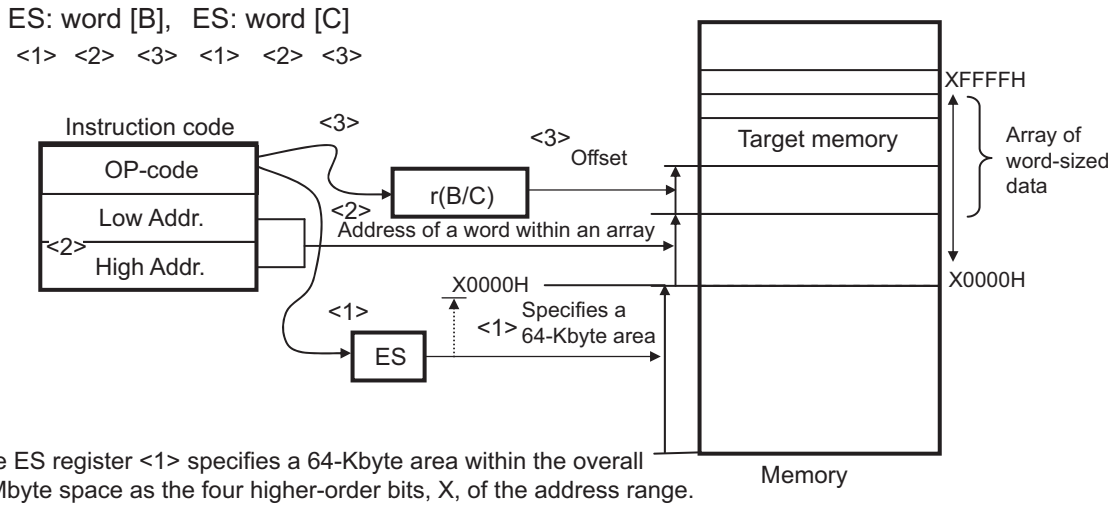

- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

**Figure 3-59. Example of ES:[HL + byte], ES:[DE + byte]**



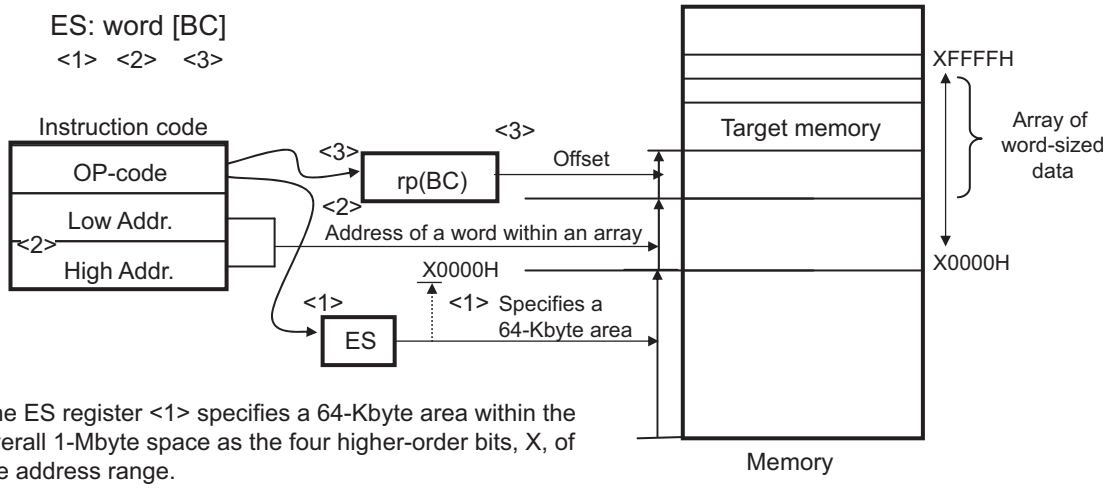

- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- "byte" <3> specifies an offset within the array to the target location in memory.

**Figure 3-60. Example of ES:word[B], ES:word[C]**




- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- "word" <2> specifies the address where the target array of word-sized data starts in the 64-Kbyte area specified in the ES register <1>.
- Either register <3> specifies an offset within the array to the target location in memory.

**Figure 3-61. Example of ES:word[BC]**




- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- "word" <2> specifies the address where the target array of word-sized data starts in the 64-Kbyte area specified in the ES register <1>.
- A pair of registers <3> specifies an offset within the array to the target location in memory.

### 3.4.8 Based indexed addressing

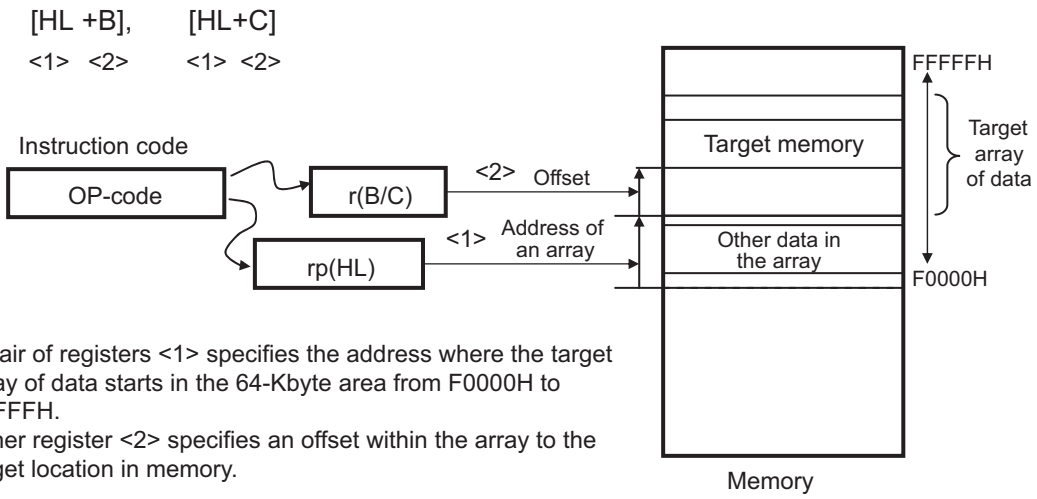
**[Function]**

Based indexed addressing uses the contents of a register pair specified with the instruction word as the base address, and the content of the B register or C register similarly specified with the instruction word as offset address. The sum of these values is used to specify the target address.

**[Operand format]**

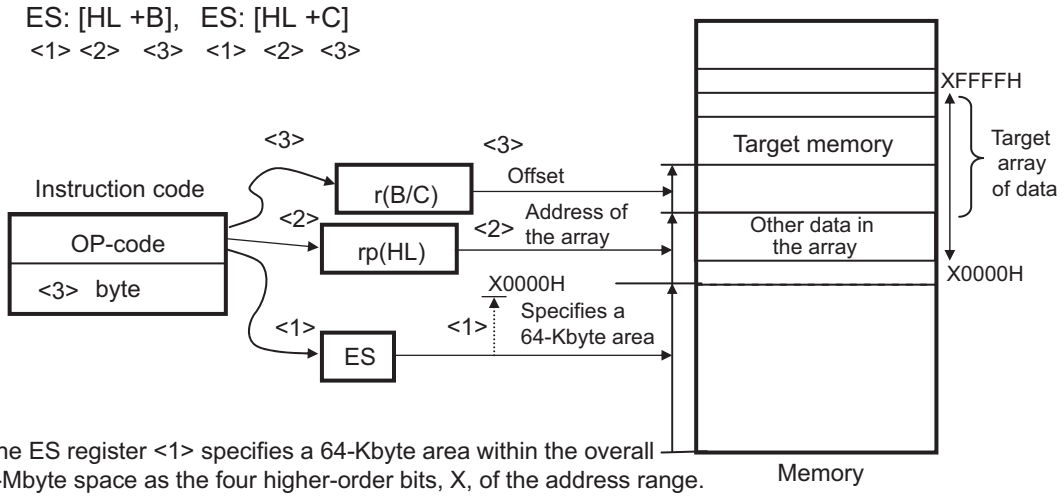
| Identifier | Description |
|---|---|
| – | [HL+B], [HL+C] (only the space from F0000H to FFFFFH is specifiable) |
| – | ES:[HL+B], ES:[HL+C] (higher 4-bit addresses are specified by the ES register) |

**Figure 3-62. Example of [HL+B], [HL+C]**




- A pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

**Figure 3-63. Example of ES:[HL+B], ES:[HL+C]**




- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- A pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- Either register <3> specifies an offset within the array to the target location in memory.

### 3.4.9 Stack addressing

**[Function]**

The stack area is indirectly addressed with the stack pointer (SP) contents. This addressing is automatically employed when the PUSH, POP, subroutine call, and return instructions are executed or the register is saved/restored upon generation of an interrupt request.

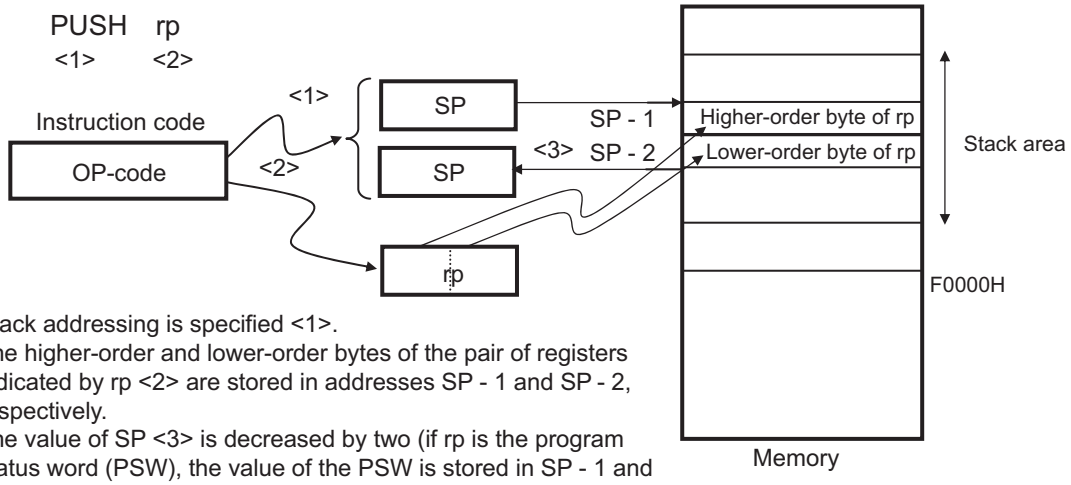
Only the internal RAM area can be set as the stack area.

**[Operand format]**

| Identifier | Description |
|---|---|
| – | PUSH PSW AX/BC/DE/HL<br>POP PSW AX/BC/DE/HL<br>CALL/CALLT<br>RET<br>BRK<br>RETB<br>(Interrupt request generated)<br>RETI |

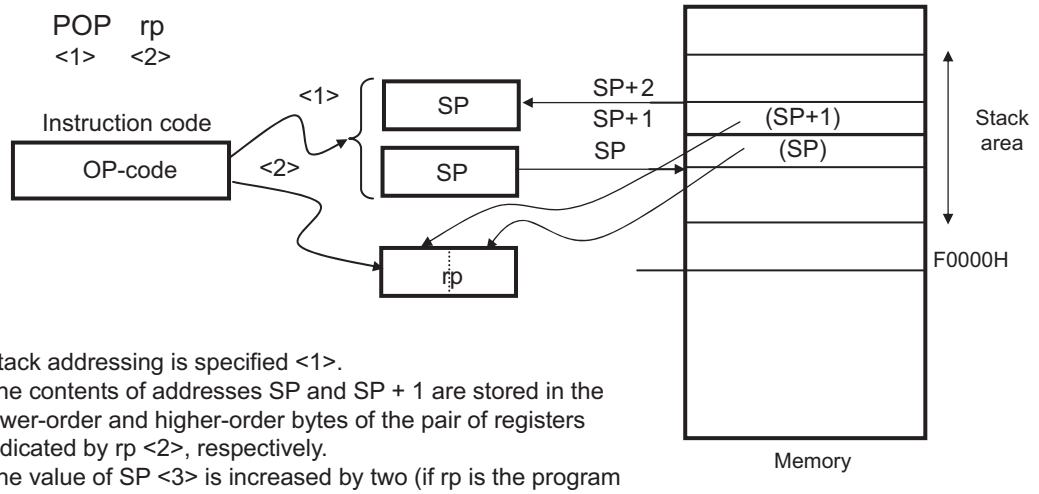
Each stack operation saves or restores data as shown in **Figures 3-64** to **3-69**.

**Figure 3-64. Example of PUSH rp**



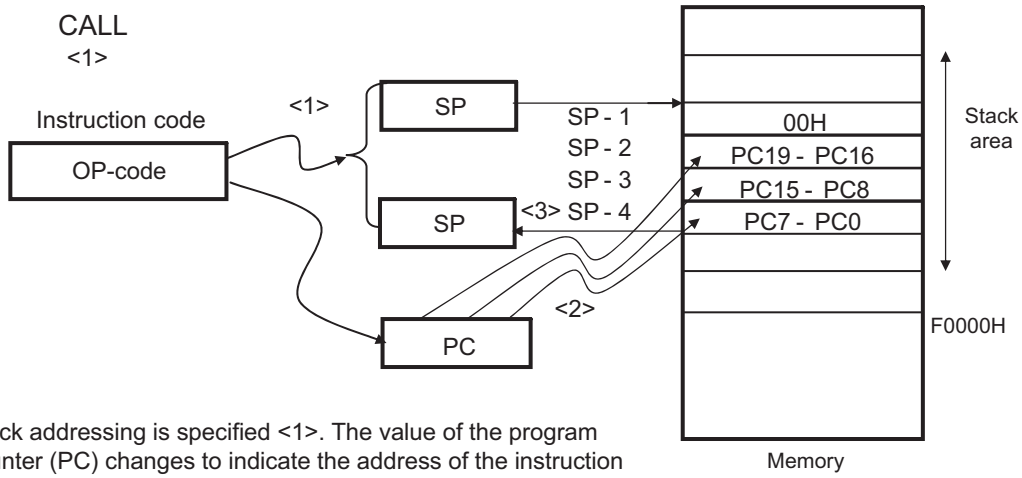

- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP - 1 and SP - 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP - 1 and 0 is stored in SP - 2).

**Figure 3-65. Example of POP**



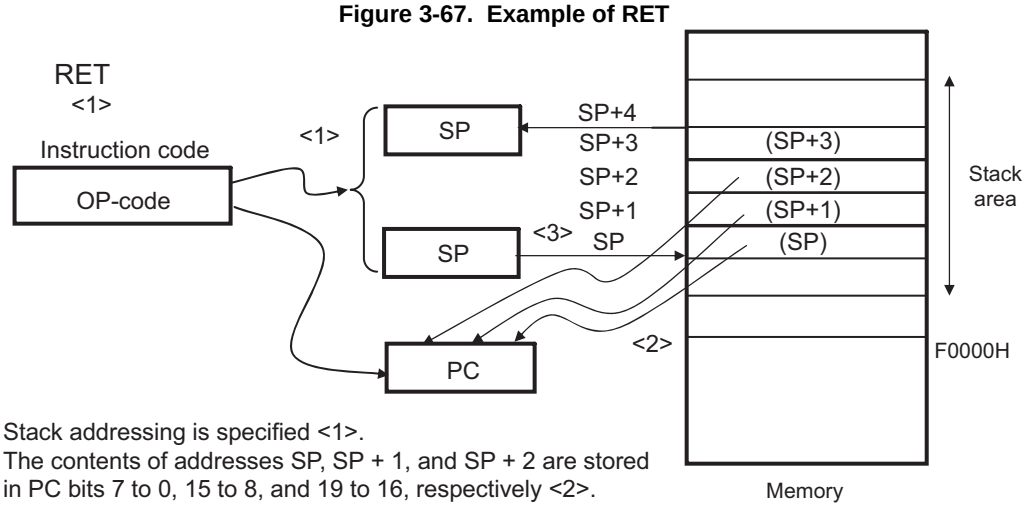

- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP + 1 is stored in the PSW).

**Figure 3-66. Example of CALL, CALLT**



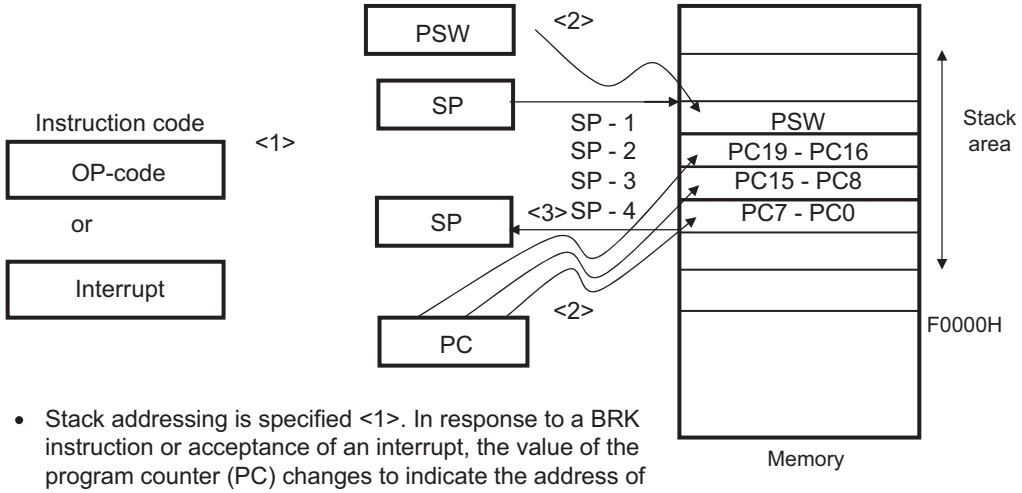

- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP - 1, SP - 2, SP - 3, and SP - 4, respectively <2>.
- The value of the SP <3> is decreased by 4.

**Figure 3-67. Example of RET**



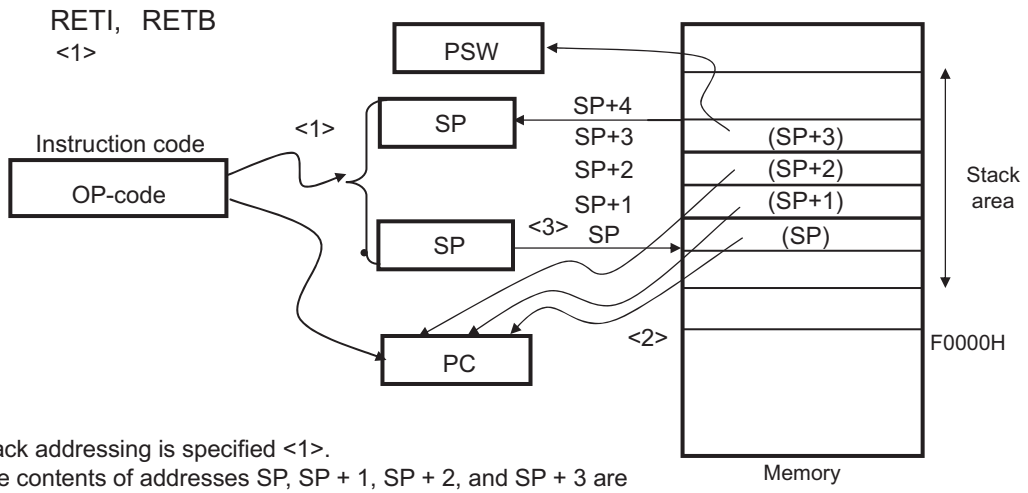

- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

**Figure 3-68. Example of Interrupt, BRK**




- Stack addressing is specified <1>. In response to a BRK instruction or acceptance of an interrupt, the value of the program counter (PC) changes to indicate the address of the next instruction.
- The values of the PSW, PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP - 1, SP - 2, SP - 3, and SP - 4, respectively <2>.
- The value of the SP <3> is decreased by 4.

**Figure 3-69. Example of RETI, RETB**




- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, SP + 2, and SP + 3 are stored in PC bits 7 to 0, 15 to 8, 19 to 16, and the PSW, respectively <2>.
- The value of SP <3> is increased by four.

# CHAPTER 4 PORT FUNCTIONS

## 4.1 Port Functions

Pin I/O buffer power supplies depend on the product. The relationship between power supplies and the pins is shown in table 4-1. $EV_{DD}$ indicates $EV_{DD0}$ and $EV_{DD1}$.

**Table 4-1. Pin I/O Buffer Power Supplies**

**(1) 20-pin, 30-pin, 32-pin, 48-pin products**

| Power Supply | Corresponding Pins |
|---|---|
| $V_{DD}$ | All pins |

**(2) 64-pin products**

| Power Supply | Corresponding Pins |
|---|---|
| $EV_{DD0}$ | Port pins other than P33, P34, P80 to P87, P90 to P96 **Note**, P121 to P124, and P137 |
| $V_{DD}$ | • P33, P34, P80 to P87, P90 to P96 **Note**, P121 to P124, and P137<br>• Pins other than port pins |

**Note** In R5F10PLE, R5F10PLF, R5F10BLC, R5F10BLD, R5F10BLE, R5F10BLF, R5F10BLG, R5F10ALF, and R5F10ALG, the power supply for P96 is $EV_{DD0}$. In R5F10ALC, R5F10ALD, and R5F10ALE, the power supply for P92 to P97 is $EV_{DD0}$.

**(3) 80-pin products**

| Power Supply | Corresponding Pins |
|---|---|
| $EV_{DD0}$ | Port pins other than P33, P34, P80 to P87, P90 to P97 **Note**, P121 to P124, and P137 |
| $V_{DD}$ | • P33, P34, P80 to P87, P90 to P97 **Note**, P121 to P124, and P137<br>• Pins other than port pins |

**Note** In R5F10PME, R5F10PMF, R5F10BME, R5F10BMF, R5F10BMG, R5F10AME, R5F10AMF, and R5F10AMG, the power supply for P96 and P97 is $EV_{DD0}$.

**(4) 100-pin products**

| Power Supply | Corresponding Pins |
|---|---|
| $EV_{DD0}$, $EV_{DD1}$ | Port pins other than P33, P34, P80 to P87, P90 to P97, P100 to P105, P121 to P124, and P137 |
| $V_{DD}$ | • P33, P34, P80 to P87, P90 to P97, P100 to P105, P121 to P124, and P137<br>• Pins other than port pins |

The products are classified into the following five groups according to the product type, pin count, and code flash memory size.

Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory

Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory

Group C: RL78/F13 (CAN and LIN incorporated) products with 30, 32, 48, 64, or 80 pins and 32 Kbytes to 128 Kbytes of code flash memory

Group D: RL78/F14 products with 30, 32, 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory

Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

The RL78/F13 and RL78/F14 microcontrollers are provided with digital I/O ports, which enable variety of control operations.

In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, see **CHAPTER 2 PIN FUNCTIONS**.

## 4.2 Port Configuration

Ports include the following hardware.

**Table 4-2. Port Configuration**

| Item | Configuration |
|---|---|
| Control registers | Port mode registers (PM0, PM1, PM3 to PM10, PM12, PM14, PM15)<br>Port registers (P0, P1, P3 to P10, P12 to P15)<br>Pull-up resistor option registers (PU0, PU1, PU3 to PU7, PU9, PU10, PU12, PU14, PU15)<br>Port input mode registers (PIM1, PIM3, PIM5 to PIM7, PIM12)<br>Port output mode registers (POM1, POM6, POM7, POM12)<br>Port mode control registers (PMC7, PMC9, PMC12)<br>A/D port configuration register (ADPC)<br>Peripheral I/O redirection registers (PIOR0 to PIOR8)<br>Port input threshold control registers (PITHL1, PITHL3 to PITHL7, PITHL10, PITHL12, PITHL15)<br>Port output slew rate select register (PSRSEL)<br>SNOOZE status output control registers 0 to 3 (PSNZCNT0 to PSNZCNT3)<br>Port mode select register (PMS) |
| Port | • 20-pin products<br>Total: 16 (CMOS I/O: 13, CMOS input: 3)<br>• 30-pin products<br>Total: 26 (CMOS I/O: 23, CMOS input: 3)<br>• 32-pin products<br>Total: 28 (CMOS I/O: 25, CMOS input: 3)<br>• 48-pin products<br>Total: 44 (CMOS I/O: 38, CMOS input: 5, CMOS output: 1)<br>• 64-pin products<br>Total: 58 (CMOS I/O: 52, CMOS input: 5, CMOS output: 1)<br>• 80-pin products<br>Total: 74 (CMOS I/O: 68, CMOS input: 5, CMOS output: 1)<br>• 100-pin products<br>Total: 92 (CMOS I/O: 86, CMOS input: 5, CMOS output: 1) |

**Caution Most of the following descriptions in this chapter use the 100-pin products as an example. In addition, the products are classified into the following five groups according to the product type, pin count, and code flash memory size.**

**Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory**

**Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory**

**Group C: RL78/F13 (CAN and LIN incorporated) products with 30, 32, 48, 64, or 80 pins and 32 Kbytes to 128 Kbytes of code flash memory**

**Group D: RL78/F14 products with 30, 32, 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory**

**Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory**

**Caution Most of the following descriptions in this chapter use the 100-pin products as an example.**

### 4.2.1 Port 0

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P03 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

This port can also be used for timer I/O, real-time clock correction clock output, and external interrupt request input.

Reset signal generation sets this port to input mode.

**Table 4-3. Settings of Registers When Using Port 0**

| Pin Name | | PM0x | Alternate Function Setting Note 5 | Remark |
|---|---|---|---|---|
| Name | I/O | | | |
| P00 | Input | 1 | × | |
| | Output | 0 | (TO05 output = 0) Note 1, 3 | |
| P01 | Input | 1 | × | |
| | Output | 0 | (TO04 output = 0) Note 1, 3 | |
| P02 | Input | 1 | × | |
| | Output | 0 | (TO06 output = 0) Note 1, 3 | |
| P03 | Input | 1 | × | |
| | Output | 0 | (RTC1HZ output = 0) Note 2, 4 | |

**Notes 1.** When a pin sharing a timer output function of the timer array unit is to be used as a general-purpose port pin, the TOmn bit of the timer output register m (TOm) and the TOEmn bit of the timer output enable register m (TOEm) corresponding to the target unit and channel must have the same setting as in the initial state (m = 0, 1, n = 0 to 7).

**2.** When a pin sharing the output (1-Hz) function of the RTC1HZ pin is to be used as a general-purpose port pin, the RCLOE1 bit of the real-time clock control register 0 (RTCC0) must have the same setting as its initial value.

**3.** When a pin sharing a timer output function of the timer array unit is to be used as a general-purpose port pin, operation of the corresponding timer output must be stopped.

**4.** The RCLOE1 bit of the real-time clock control register 0 (RTCC0) must have the same setting as its initial value.

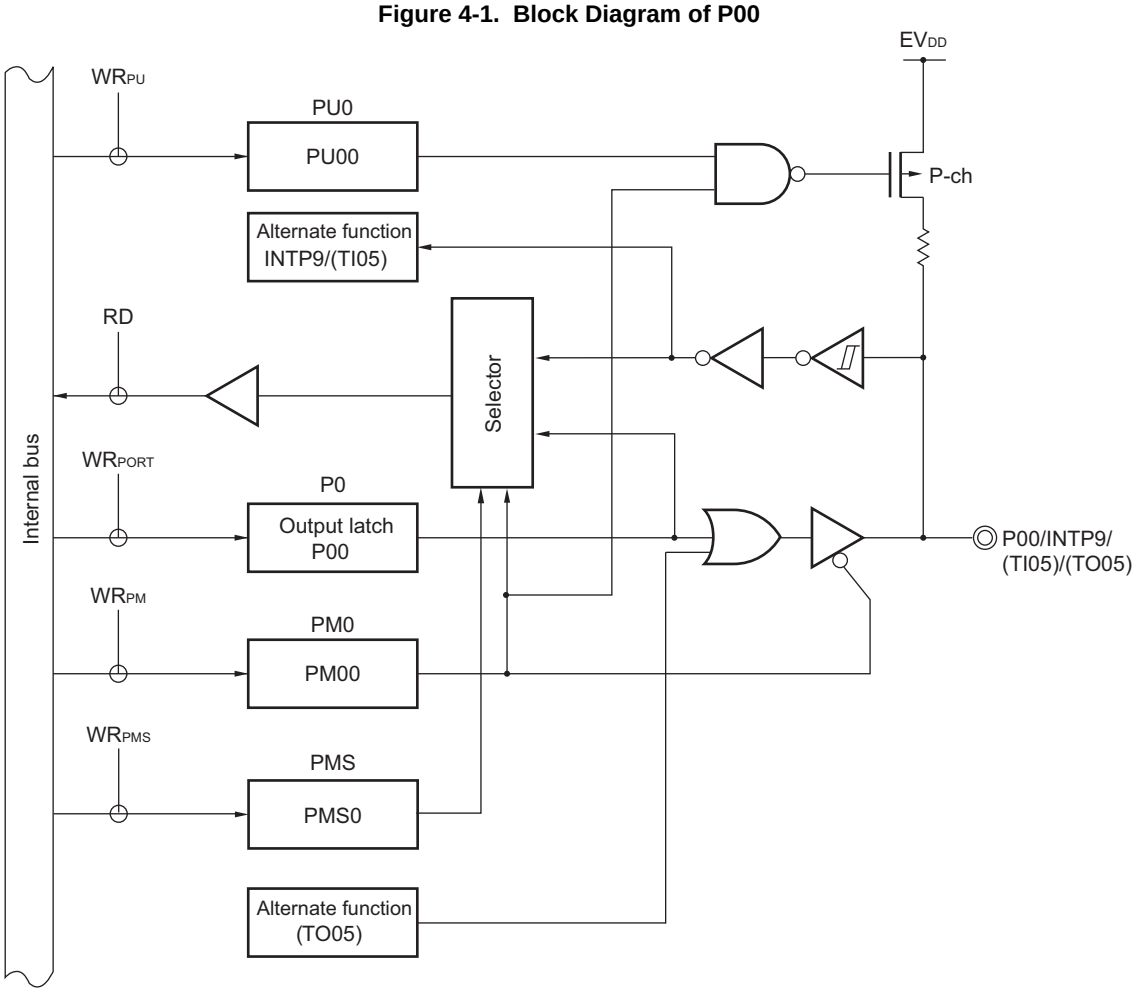
**5.** Functions in parentheses can be assigned via settings in the peripheral I/O redirection registers 1, 8 (PIOR1, PIOR8).

**Remark** ×: Don't care

PM0x: Port mode register 0

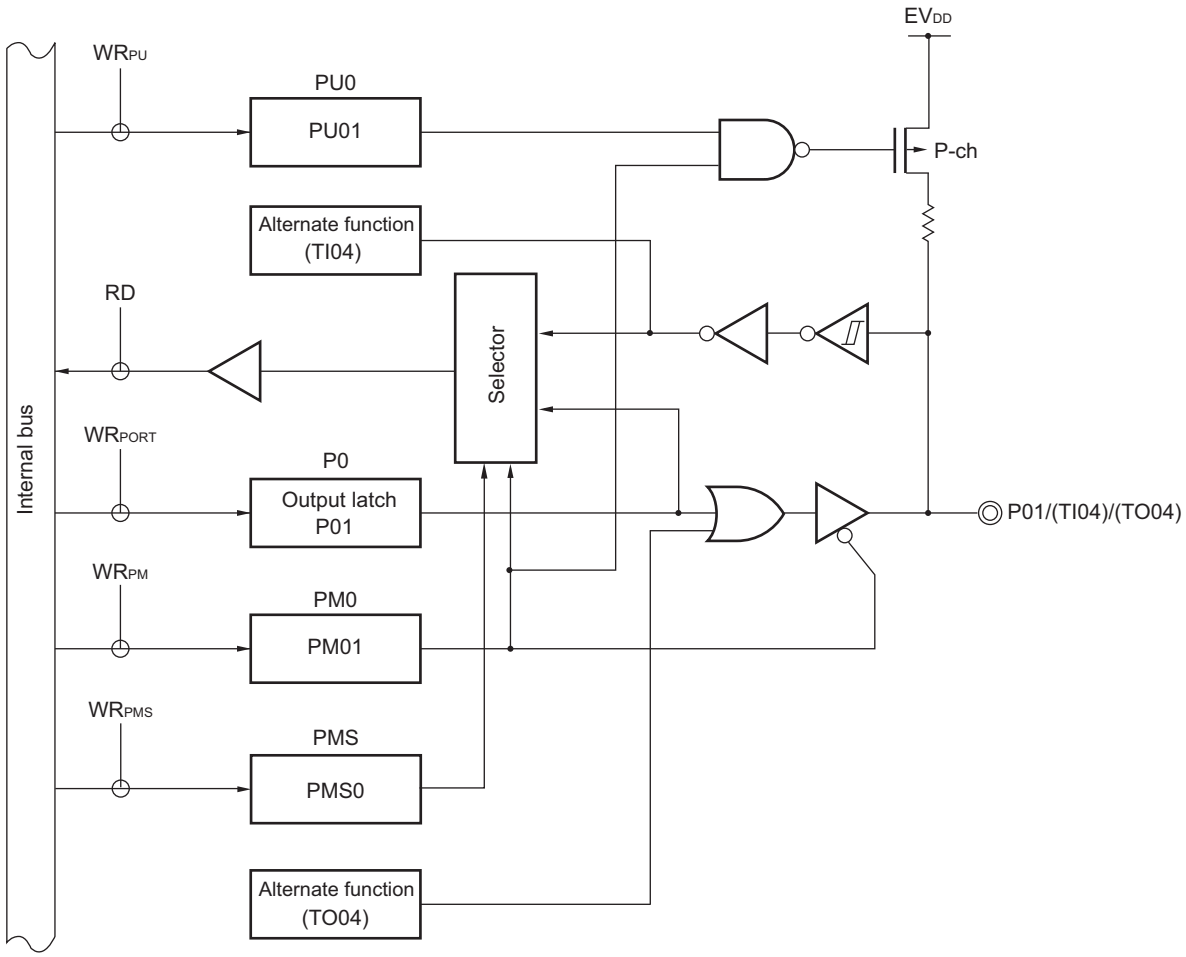
For example, figures 4-1 and 4-4 show block diagrams of port 0 for 100-pin products.

**Figure 4-1. Block Diagram of P00**



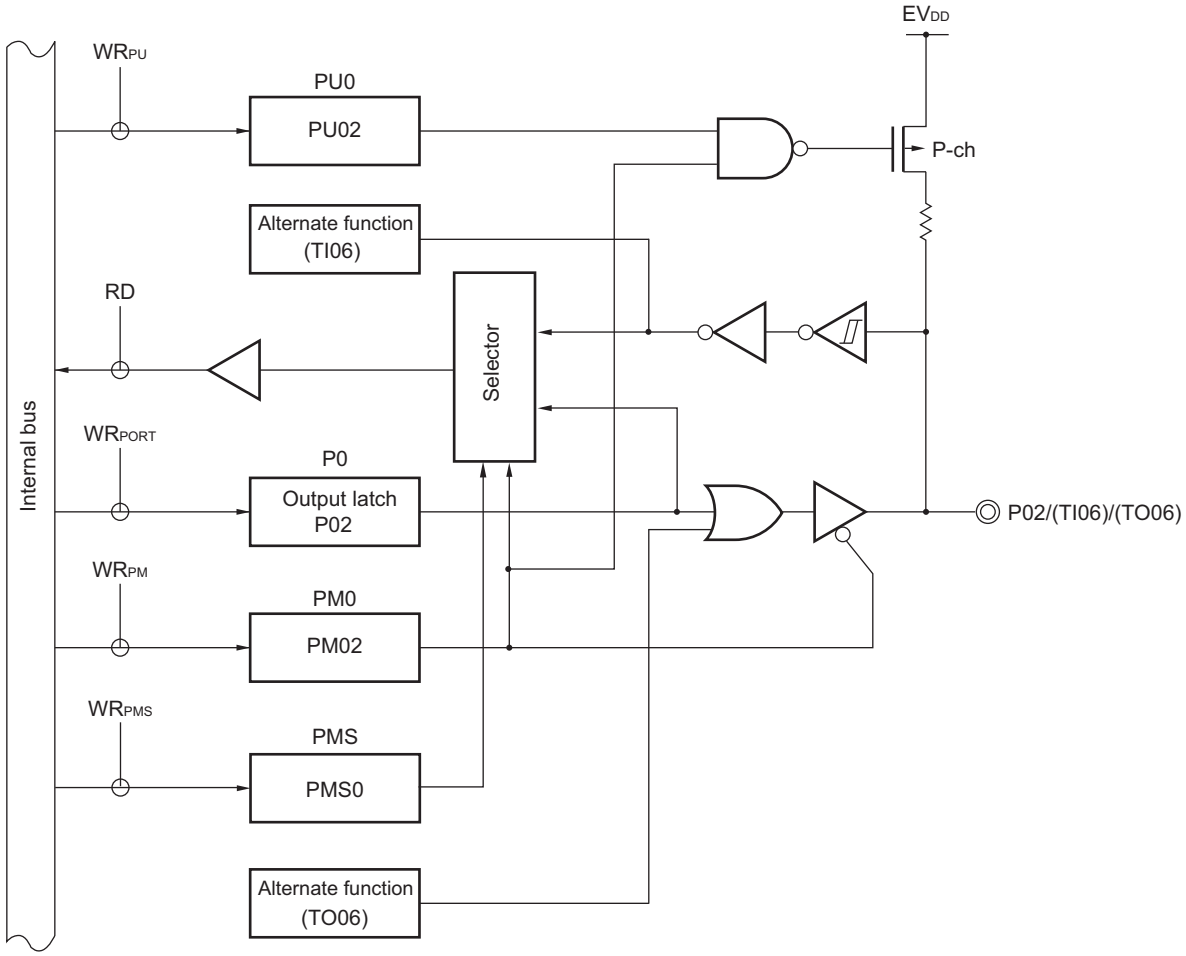

P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-2. Block Diagram of P01**



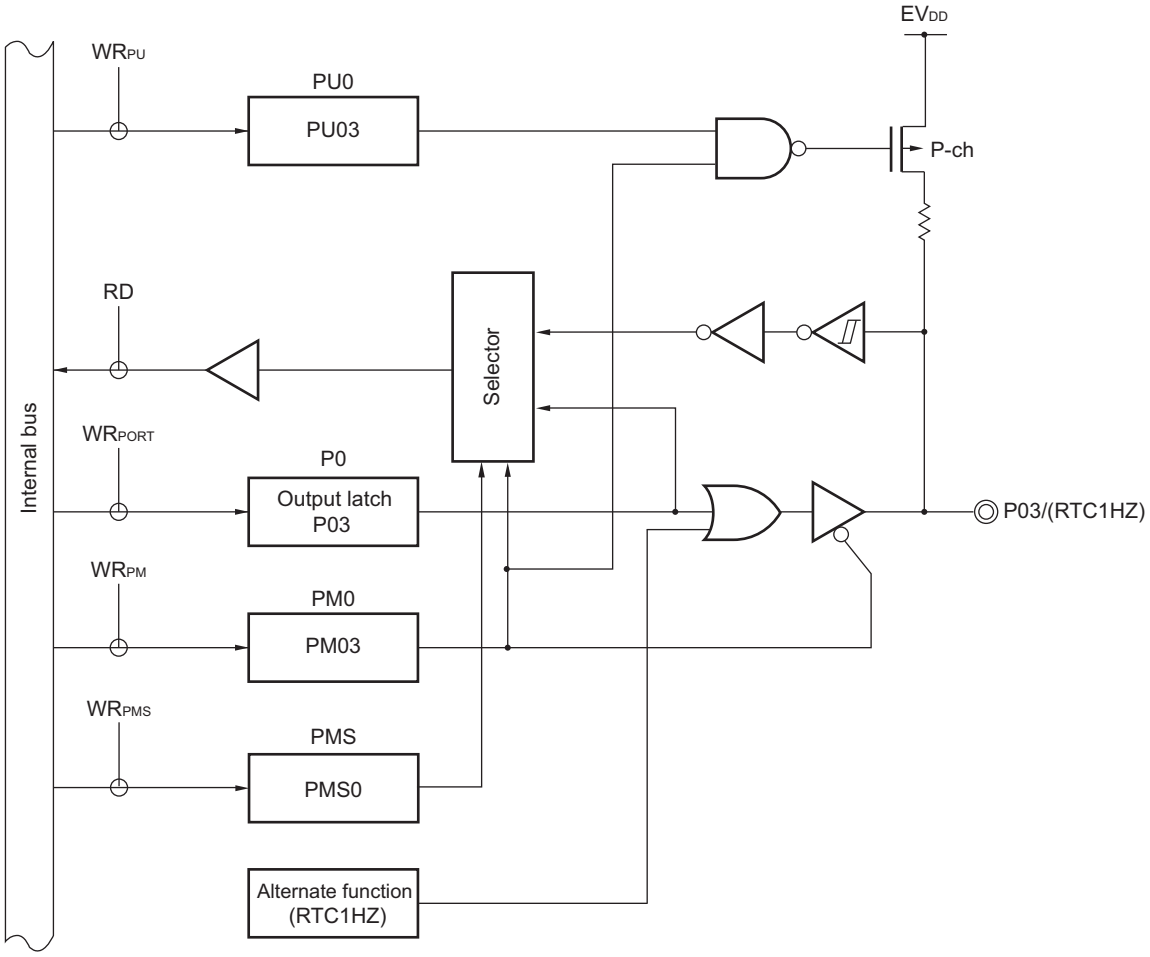

P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-3. Block Diagram of P02**




P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-4. Block Diagram of P03**




P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

### 4.2.2 Port 1

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

Input to the P10, P11, P13, P14, P16, and P17 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 1 (PIM1).

Output from the P10 to P17 pins can be specified as N-ch open-drain output ($EV_{DD}$ tolerance) in 1-bit units using port output mode register 1 (POM1).

Input to the P10, P11, P13, P14, P16, and P17 pins can be specified through an input buffer in 1-bit units using the port input threshold control register 1 (PITHL1).

This port can also be used for data I/O and clock I/O for serial interfaces (simplified IIC, CSI, and UART), serial data I/O for LIN, serial data I/O for CAN, real-time clock correction clock output, programming UART I/O, timer I/O, external interrupt request input, and SNOOZE status output.

Reset signal generation sets this port to input mode.

**Table 4-4. Settings of Registers When Using Port 1 (1/3)**

| Pin name | | PM1x | PIM1x | POM1x | PITHL1x | Alternate Function Setting Note 11 | Remark |
|---|---|---|---|---|---|---|---|
| Name | I/O | | | | | | |
| P10 | Input | 1 | 0 | × | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | 1 | | CMOS input (Schmitt3 input) |
| | | | 1 | × | × | × | TTL input |
| | Output | 0 | × | 0 | × | $\overline{SCK10}$/SCL10 output = 1 Note 1 | CMOS output |
| | | | × | 1 | × | TO13 output = 0 Note 2<br>TRJO0 output = 0 Note 3<br>LTXD1 output = 1 Note 8<br>CTXD0 output = 1 Note 9 | N-ch O.D output |
| P11 | Input | 1 | 0 | × | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | 1 | | CMOS input (Schmitt3 input) |
| | | | 1 | × | × | × | TTL input |
| | Output | 0 | × | 0 | × | SDA10 output = 1 Note4 | CMOS output |
| | | | × | 1 | × | TO12 output = 0 Note2<br>(TRDIOB0 output = 0) Note5 | N-ch O.D output |
| P12 | Input | 1 | – | × | – | × | |
| | Output | 0 | – | 0 | – | TO11 output = 0 Note 2 | CMOS output |
| | | | – | 1 | – | SO10/TxD1 output = 1 Note 4<br>SNZOUT3 output = 0 Note 7<br>(TRDIOD0 output = 0) Note 5 | N-ch O.D output |

(**Notes** and **Remark** are listed on the bottom of Table 4-4 Settings of Registers When Using Port 1 (3/3).)

**Table 4-4. Settings of Registers When Using Port 1 (2/3)**

| Pin name | | PM1x | PIM1x | POM1x | PITHL1x | Alternate Function Setting Note 11 | Remark |
|---|---|---|---|---|---|---|---|
| Name | I/O | | | | | | |
| P13 | Input | 1 | 0 | × | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | 1 | | CMOS input (Schmitt3 input) |
| | | | 1 | × | × | × | TTL input |
| | Output | 0 | × | 0 | × | TRDIOA0 output = 0 Note 5 | CMOS output |
| | | | × | 1 | × | SDA01 output = 1 Note 4<br>TO04 output = 0 Note 2<br>LTXD0 output = 1 Note 8 | N-ch O.D output |
| P14 | Input | 1 | 0 | × | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | 1 | | CMOS input (Schmitt3 input) |
| | | | 1 | × | × | × | TTL input |
| | Output | 0 | × | 0 | × | TRDIOC0 output = 0 Notes 5, 10 | CMOS output |
| | | | × | 1 | × | $\overline{\text{SCK01}}$/SCL01 output = 1 Note 1<br>TO06 output = 0 Note 2 | N-ch O.D output |
| P15 | Input | 1 | – | × | – | × | |
| | Output | 0 | – | 0 | – | TRDIOA1 output = 0 Note 5 | CMOS output |
| | | | – | 1 | – | TO05 output = 0 Note 2<br>SO00 output/TXD0 output = 1 Note 4<br>RTC1HZ output = 0 Note 6<br>(TRDIOA0 output = 0) Note 5 | N-ch O.D output |
| P16 | Input | 1 | 0 | × | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | 1 | | CMOS input (Schmitt3 input) |
| | | | 1 | × | × | × | TTL input |
| | Output | 0 | × | 0 | × | SDA00 output = 1 Note 4 | CMOS output |
| | | | × | 1 | × | TRDIOC1 output = 0 Note 5<br>TO02 output = 0 Note 2 | N-ch O.D output |

(**Notes** and **Remark** are listed on the next page.)

**Table 4-4. Settings of Registers When Using Port 1 (3/3)**

| Pin name | | PM1x | PIM1x | POM1x | PITHL1x | Alternate Function Setting [Note 7] | Remark |
|---|---|---|---|---|---|---|---|
| Name | I/O | | | | | | |
| P17 | Input | 1 | 0 | × | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | 1 | | CMOS input (Schmitt3 input) |
| | | 1 | 1 | × | × | × | TTL input |
| | Output | 0 | × | 0 | × | TRDIOB1 output = 0 [Note 5] | CMOS output |
| | | | × | 1 | × | $\overline{SCK00}$/SCL00 output = 1 [Note 1]<br>TO00 output = 0 [Note 2] | N-ch O.D output |

**Notes 1.** When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the CKOmn bit of the serial output register m (SOm), the SOEmn bit of the serial output enable register m (SOEm), and the SEmn bit of the serial channel enable status register m (SEm) corresponding to the target unit and channel must have the same setting as its initial value (m = 0, 1, n = 0, 1).

**2.** When a pin sharing a timer output function of the timer array unit is to be used as a general-purpose port pin, the TOmn bit of the timer output register m (TOm) and the TOEmn bit of the timer output enable register m (TOEm) corresponding to the target unit and channel must have the same setting as in the initial state (m = 0, 1, n = 0 to 7).

**3.** When a pin sharing the timer output function of the timer RJ is to be used as a general-purpose port pin, the bit 2 (TOENA) of the timer RJ I/O control register 0 (TRJIOC0) must have the same setting as its initial value.

**4.** When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn bit of the serial output register m (SOm), the SOEmn bit of the serial output enable register m (SOEm), and the SEmn bit of the serial channel enable status register m (SEm) corresponding to the target unit and channel must have the same setting as its initial value (m = 0, 1, n = 0, 1).

**5.** When a pin sharing a timer RD function is to be used as a general-purpose port pin, the target bit for TRDIOij pin output control in the timer RD output master enable register 1 (TRDOER1) must have the same setting as its initial value (i = A, B, C, D, j = 0, 1)

**6.** The RCLOE1 bit of the real-time clock control register 0 (RTCC0) must have the same setting as its initial value.

**7.** When a pin sharing the SNOOZE status output function is to be used as a general-purpose port pin, the OUTEN0 to OUTEN7 bits of the SNOOZE status output control registers 0, 1, 2, 3 (PSNZCNT0, 1, 2, 3) must have the same setting as its initial value.

**8.** When a pin sharing the serial data output function of the LIN is to be used as a general-purpose port pin, operation of the corresponding LIN must be stopped.

**9.** When a pin sharing the serial data output function of the CAN is to be used as a general-purpose port pin, operation of the corresponding CAN must be stopped.

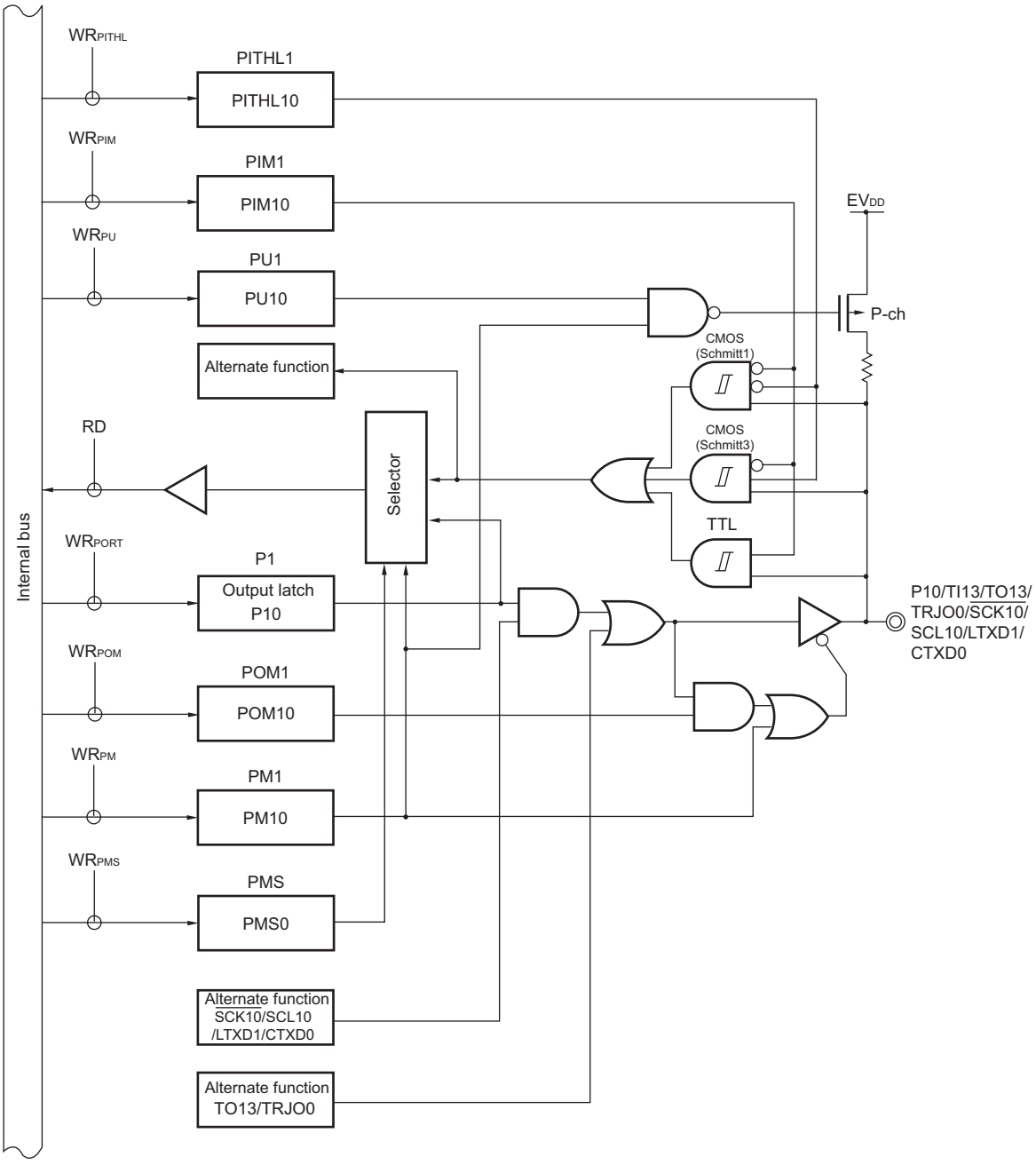
**10.** When the SNOOZE status output is in use, output from TRDIOC0 is stopped.

**11.** Functions in parentheses can be assigned via settings in the peripheral I/O redirection register 7 (PIOR7).

**Remark** ×: Don't care
PM1x: Port mode register 1
PIM1x: Port input mode register 1
POM1x: Port output mode register 1
PITHL1X: Port input threshold control register 1

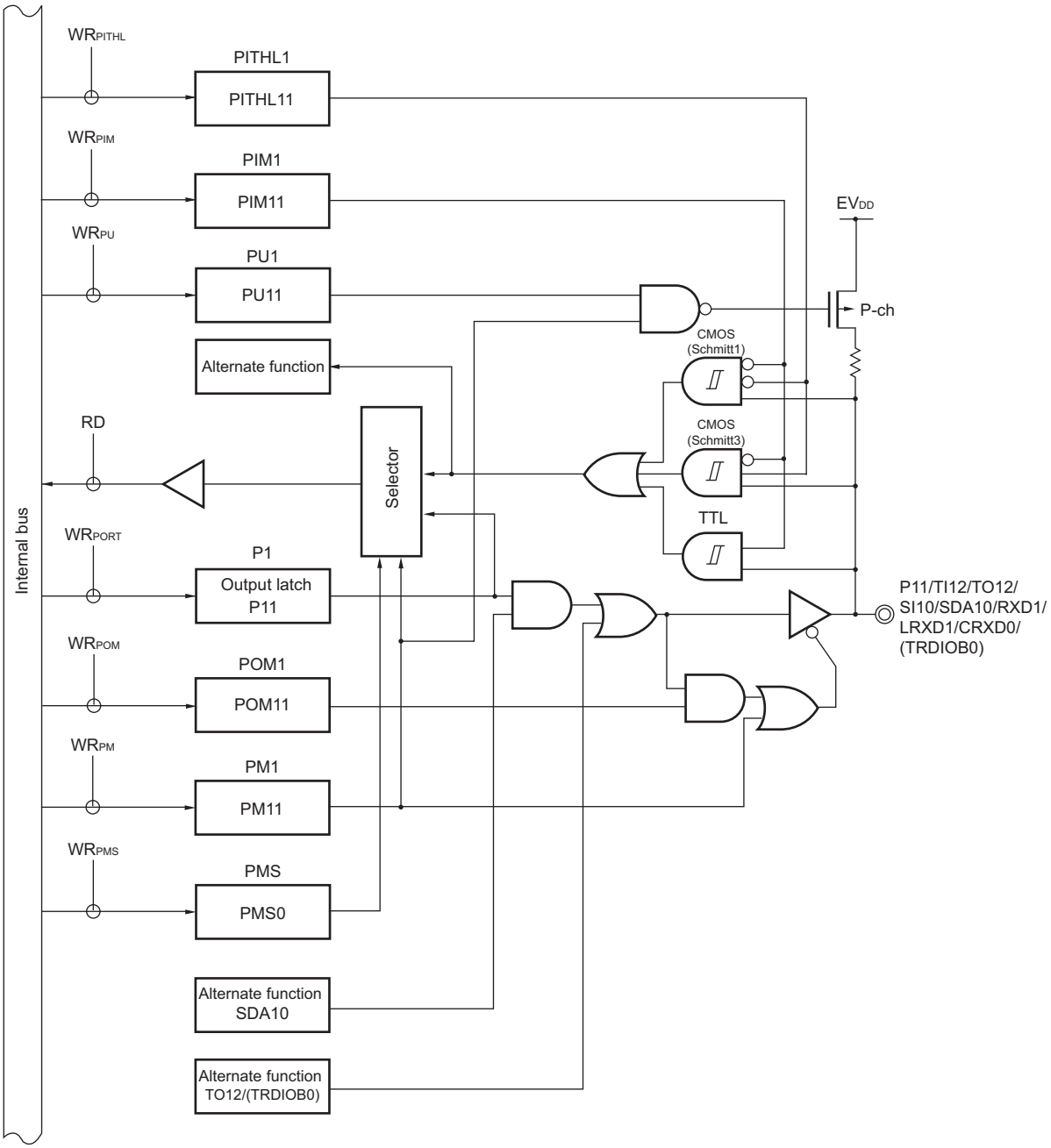
Figures 4-5 to 4-12 show block diagrams of port 1 for 100-pin products.

**Figure 4-5. Block Diagram of P10**



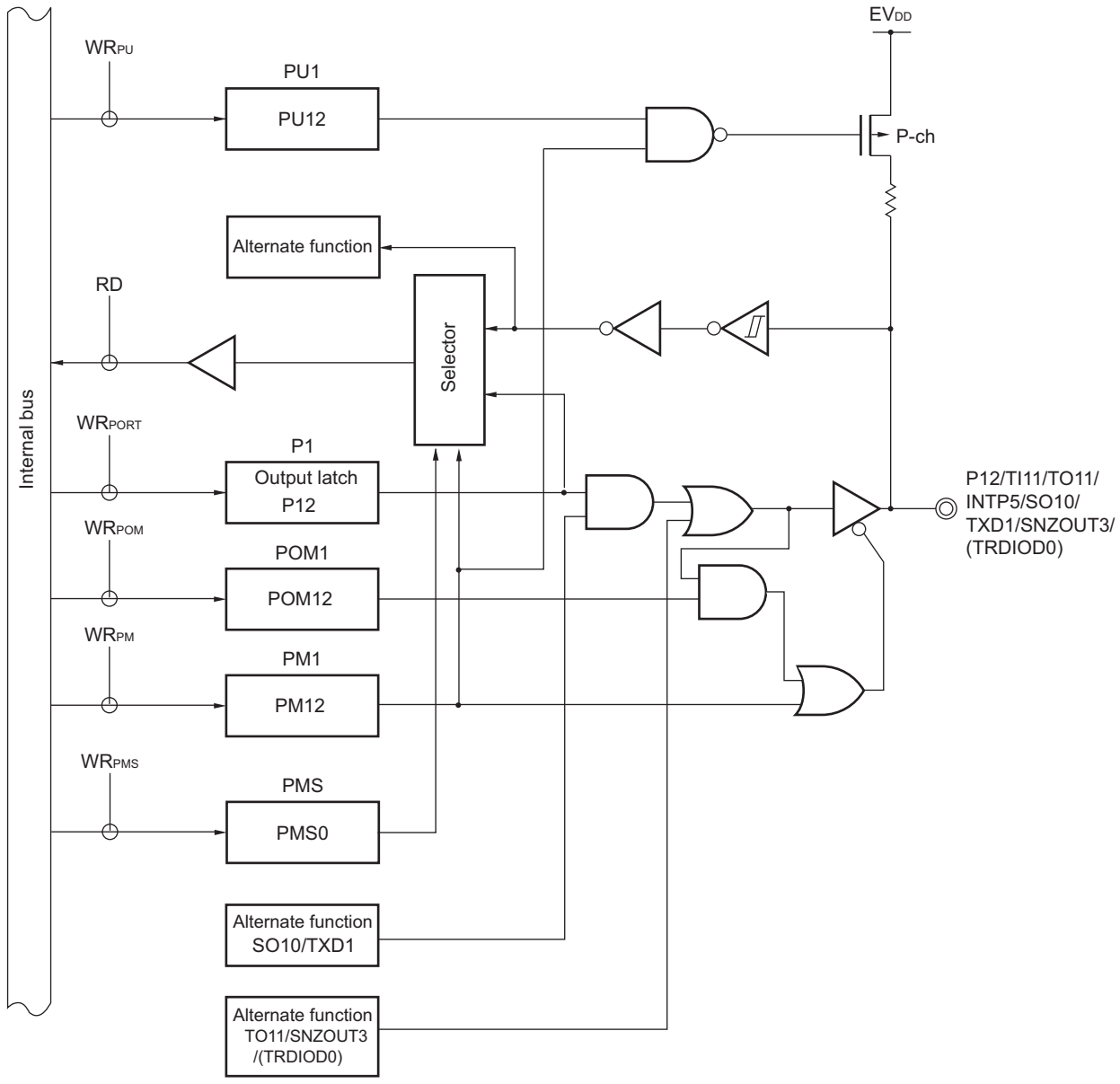

P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
PMS: Port mode select register
PITHL1: Port input threshold control register 1
RD: Read signal
WRxx: Write signal

**Figure 4-6. Block Diagram of P11**



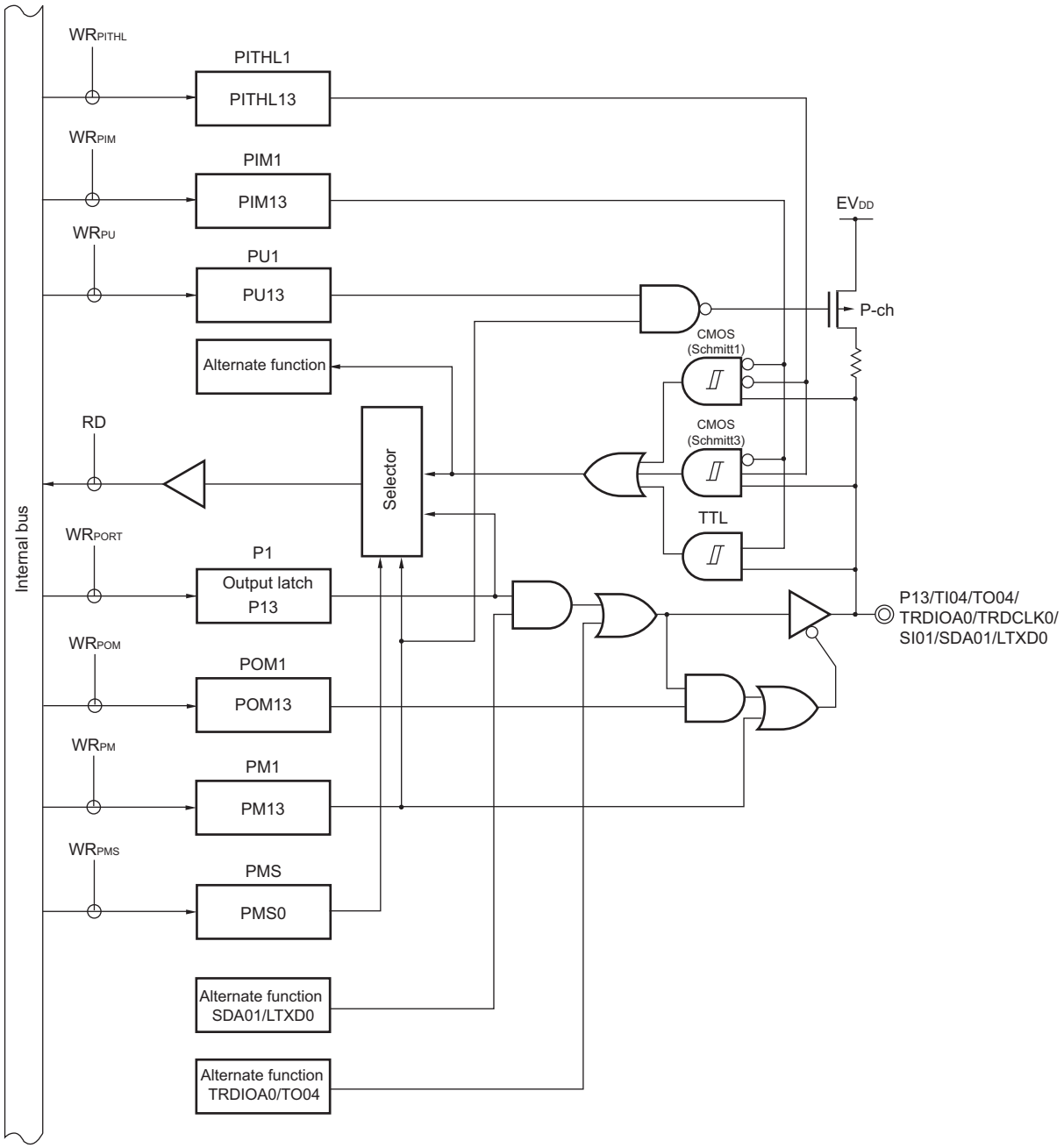

P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
PMS: Port mode select register
PITHL1: Port input threshold control register 1
RD: Read signal
WRxx: Write signal

**Figure 4-7. Block Diagram of P12**



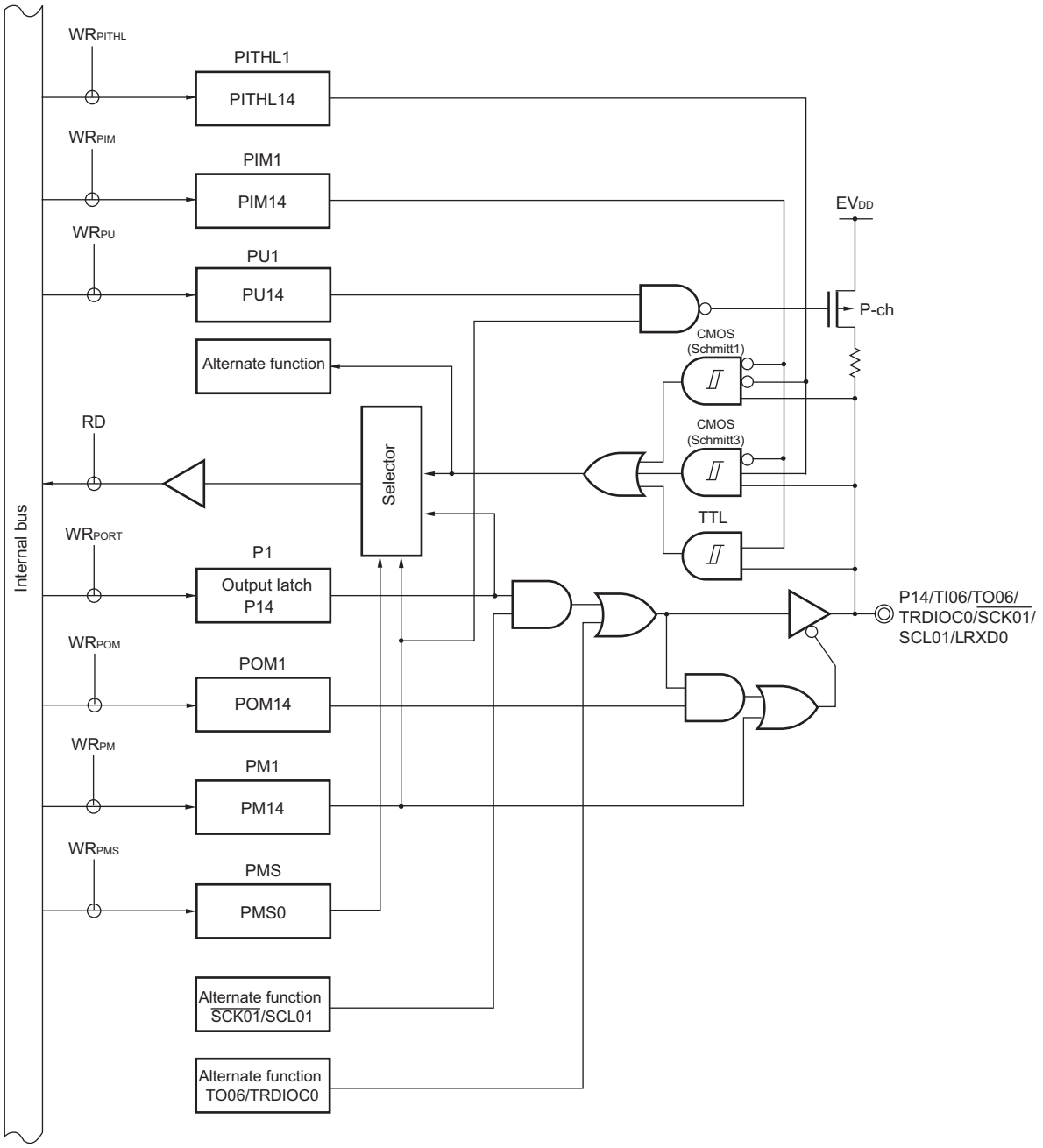

P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
POM1: Port output mode register 1
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-8. Block Diagram of P13**



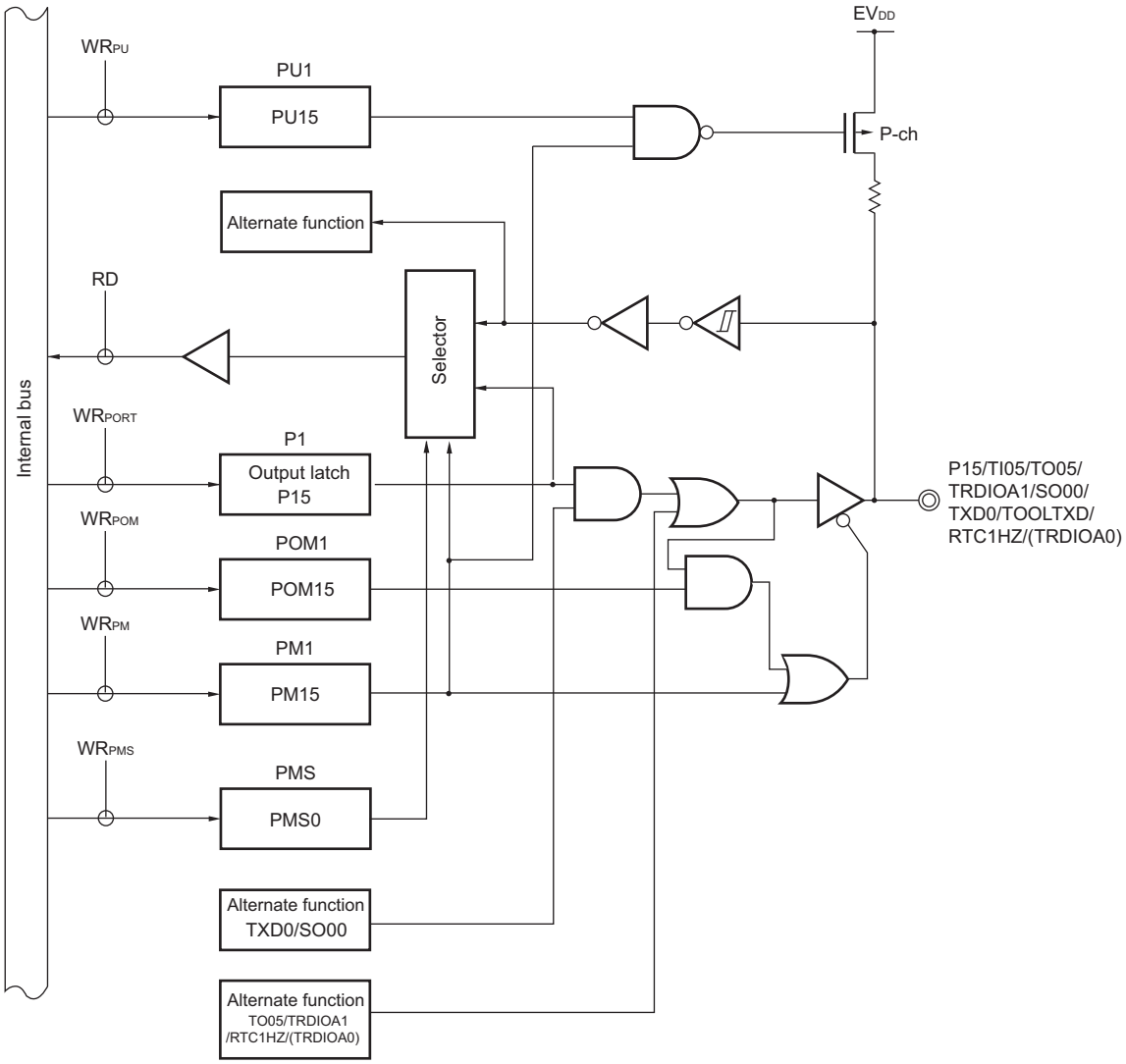

P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
PMS: Port mode select register
PITHL1: Port input threshold control register 1
RD: Read signal
WRxx: Write signal

**Figure 4-9. Block Diagram of P14**



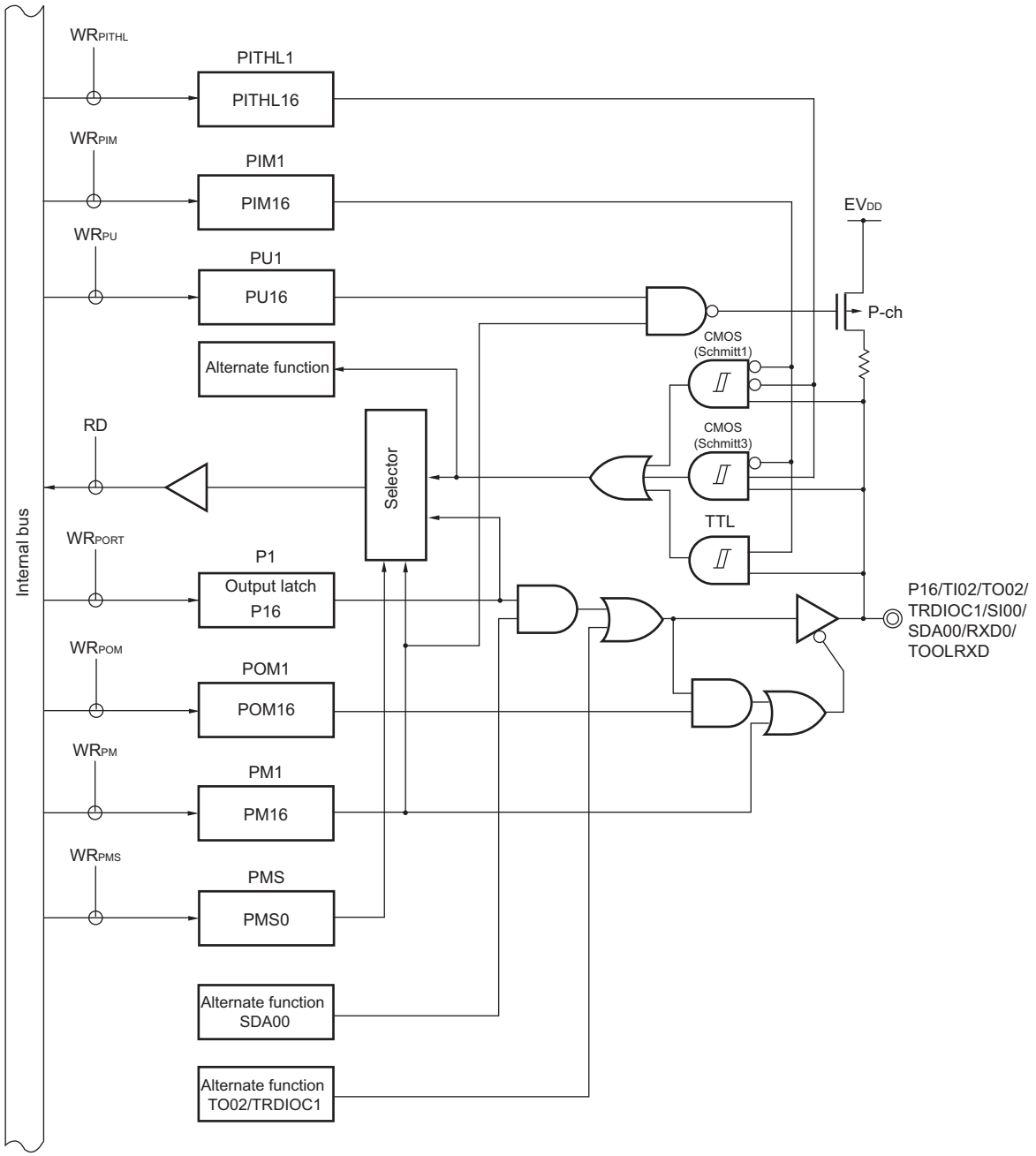

P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
PMS: Port mode select register
PITHL1: Port input threshold control register 1
RD: Read signal
WRxx: Write signal

**Figure 4-10. Block Diagram of P15**



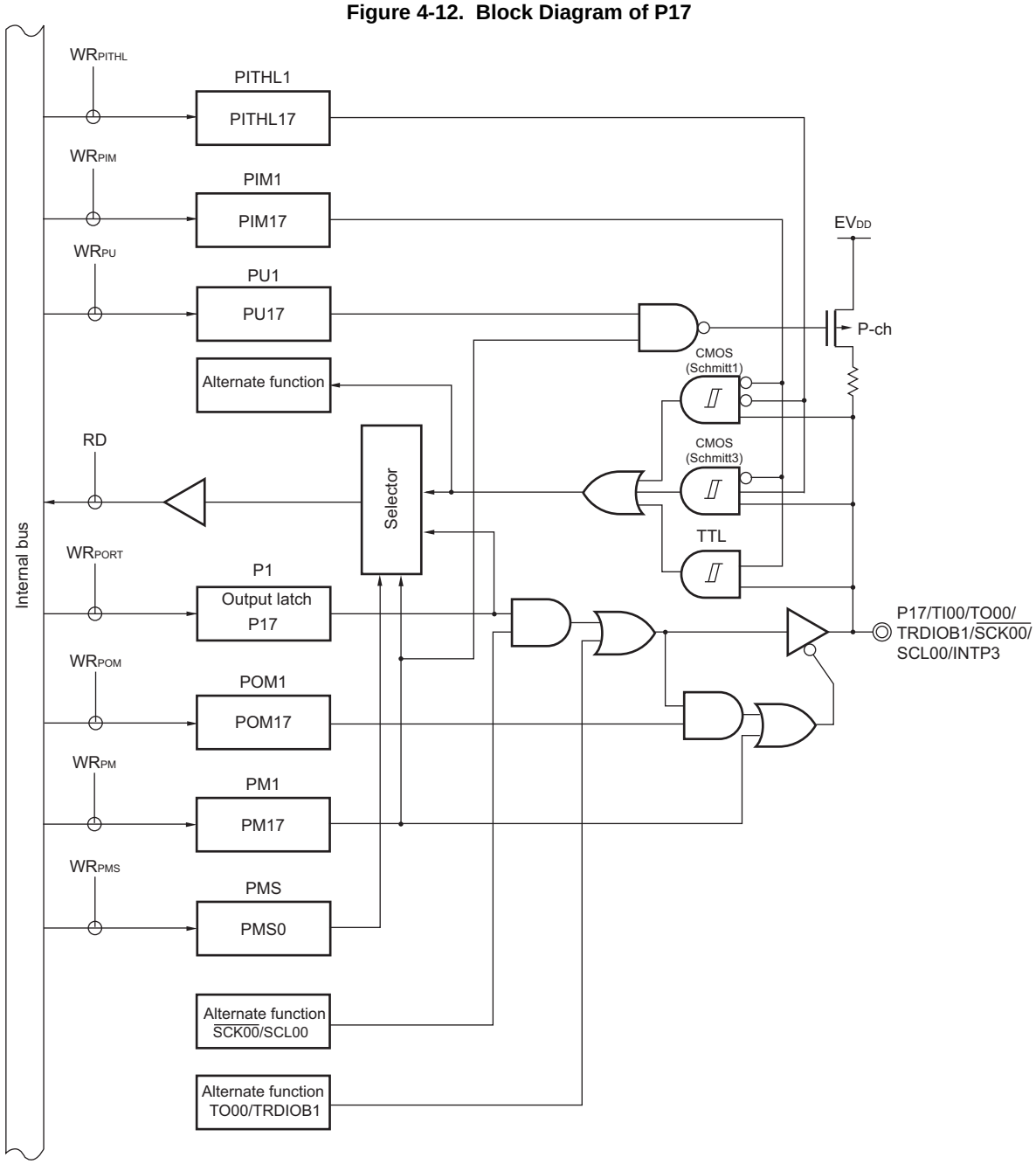

P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
POM1: Port output mode register 1
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-11. Block Diagram of P16**




P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
PMS: Port mode select register
PITHL1: Port input threshold control register 1
RD: Read signal
WRxx: Write signal

**Figure 4-12. Block Diagram of P17**




P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
PMS: Port mode select register
PITHL1: Port input threshold control register 1
RD: Read signal
WRxx: Write signal

### 4.2.3 Port 3

Port 3 is an I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 to P32 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

Input to the P30 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 3 (PIM3). For the P30 pin input, the threshold of the input buffer can be specified in 1-bit units using the port input threshold control register 3 (PITHL3).

This port can also be used for external interrupt request input, timer I/O, serial interface slave select input, SNOOZE status output, and STOP status output.

P33 and P34 can also be used for A/D converter analog input and reference voltage input (+side and – side).

To use P33/ANI0 and P34/ANI1 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM3 register. Use these pins starting from the upper bit.

To use P33/ANI0 and P34/ANI1 as digital output pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the output mode by using the PM3 register. Use these pins starting from the upper bit.

To use P33/ANI0 and P34/ANI1 as analog I/O pins, set them in the analog I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM3 register. Use these pins starting from the lower bit.

Reset signal generation sets P30 to P32 to input mode and P33/ANI0 and P34/ANI1 to analog input mode.

**Table 4-5. Settings of Registers When Using Port 3 (P30 to P32) (1)**

| Pin name | | PM3x | PIM3x | PITHL3x | Alternate Function Setting Note 5 | Remark |
|---|---|---|---|---|---|---|
| Name | I/O | | | | | |
| P30 | Input | 1 | 0 | 0 | × | CMOS input (Schmitt1 input) |
| | | | | 1 | | CMOS input (Schmitt3 input) |
| | | | 1 | × | × | TTL input |
| | Output | 0 | × | × | TRDIOD1 output = 0 Note 1<br>TO01 output = 0 Note 2<br>SNZOUT0 output = 0 Note 3 | |
| P31 | Input | 1 | – | – | × | |
| | Output | 0 | – | – | TO14 output = 0 Note 2<br>STOPST output = 0 Note 4 | |
| P32 | Input | 1 | – | – | × | |
| | Output | 0 | – | – | TO16 output = 0 Note 2 | |

**Notes 1.** When a pin sharing a timer RD function is to be used as a general-purpose port pin, the target bit for TRDIOij pin output control in the timer RD output master enable register 1 (TRDOER1) must have the same setting as its initial value (i = A, B, C, D, j = 0, 1)

**2.** When a pin sharing a timer output function of the timer array unit is to be used as a general-purpose port pin, the TOmn bit of the timer output register m (TOm) and the TOEmn bit of the timer output enable register m (TOEm) corresponding to the target unit and channel must have the same setting as in the initial state (m = 0, 1, n = 0 to 7).

**3.** When a pin sharing the SNOOZE status output function is to be used as a general-purpose port pin, the OUTEN0 to OUTEN7 bits of the SNOOZE status output control registers 0, 1, 2, 3 (PSNZCNT0, 1, 2, 3) must have the same setting as its initial value.

**4.** When a pin sharing the STOP status output function is to be used as a general-purpose port pin, the STPOEN bit of the STOP status output control register (STPSTC) must have the same setting as its initial value.

**5.** Functions in parentheses can be assigned via settings in the peripheral I/O redirection registers 0 to 8 (PIOR0 to PIOR8).

**Remark** ×: Don't care
PM3x: Port mode register 3
PIM3x: Port input mode register 3
PITHL3x: Port input threshold control register 3

**Table 4-6. Settings of Registers When Using Port 3 (P33 and P34) (2)**

| Pin Name | | PM3x | ADPC | Alternate Function Setting | Remark |
|---|---|---|---|---|---|
| Name | I/O | | | | |
| P3n | Input | 1 | 01 to n-2H | — | To use P3n as a port, use these pins from the upper bit. |
| | Output | 0 | 01 to n-2H | | |

**Remarks 1.** PM3x: Port mode register 3
ADPC: A/D port configuration register
**2.** n = 3 or 4

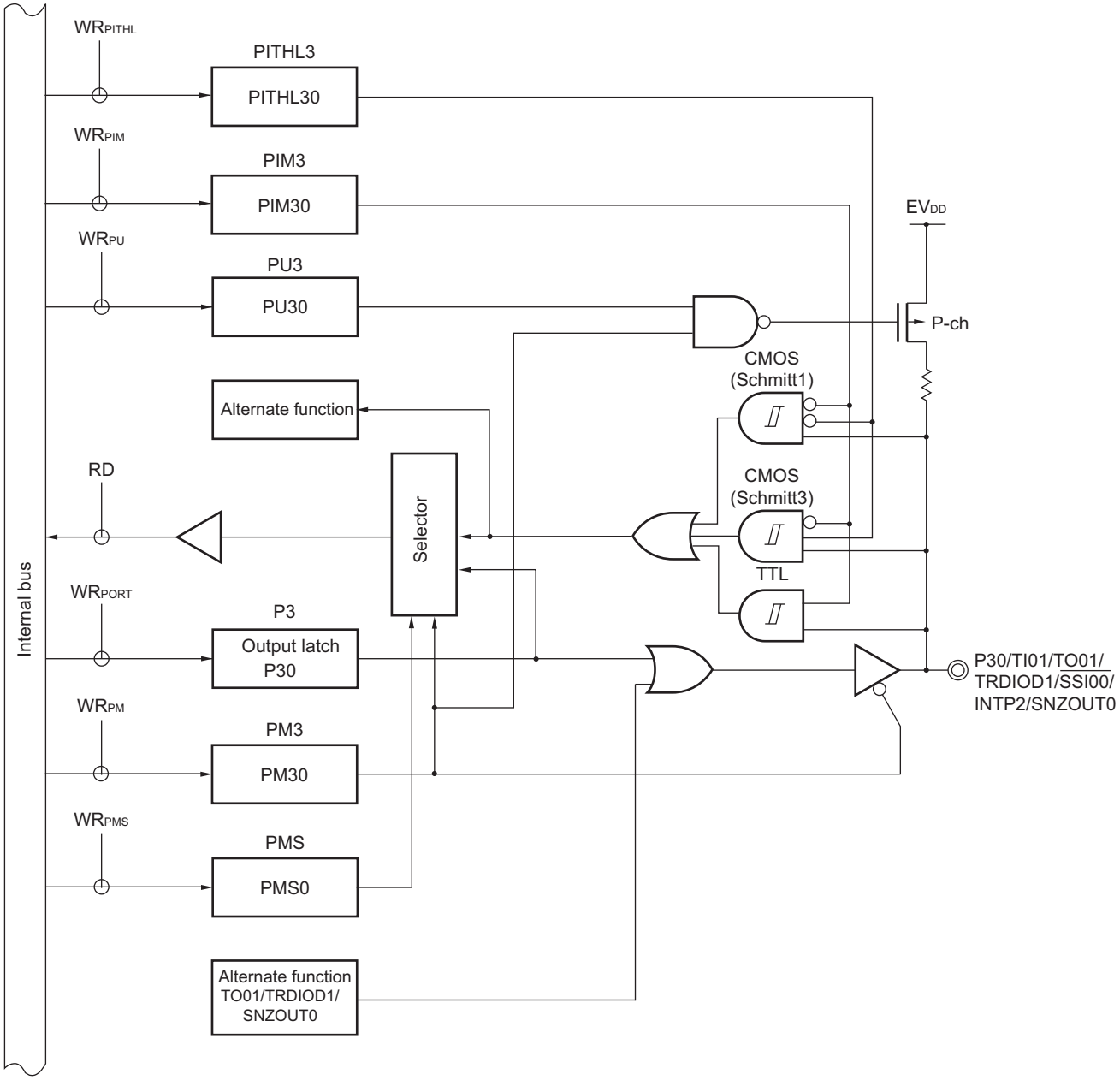
**Table 4-7. Setting Functions of P33/ANI0 and P34/ANI1 Pins**

| ADPC Register | PM3 Register | ADS Register | P33/ANI0 and P34/ANI1 Pins |
|---|---|---|---|
| Digital I/O selection | Input mode | — | Digital input |
| | Output mode | — | Digital output |
| Analog input selection | Input mode | Selects ANI. | Analog input (to be converted) |
| | | Does not select ANI. | Analog input (not to be converted) |
| | Output mode | Selects ANI. | Setting prohibited |
| | | Does not select ANI. | |

Reset signal generation sets P33/ANI0 and P34/ANI1 to analog input mode.

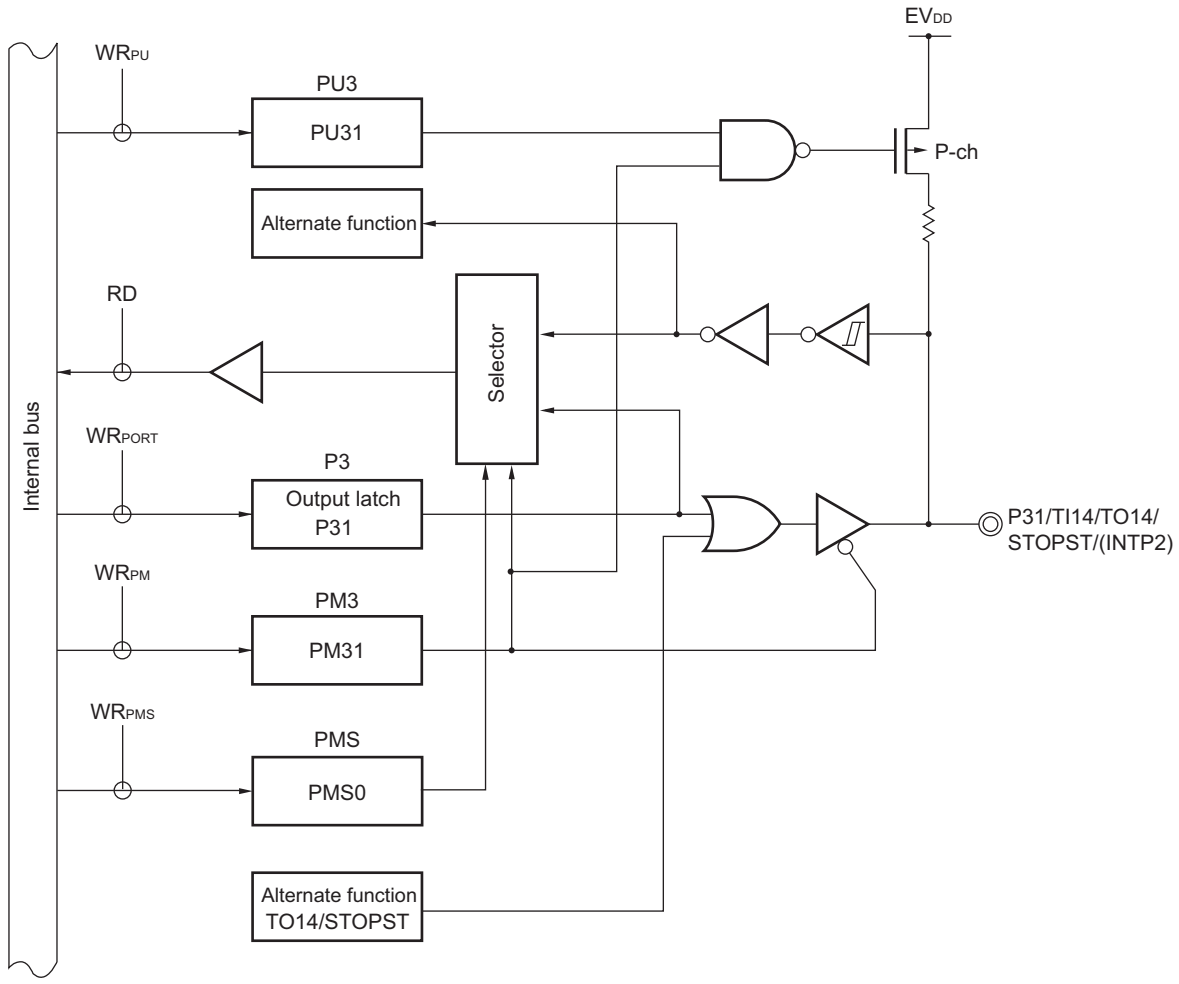
Figures 4-13 to 4-16 show block diagrams of port 3 for 100-pin products.

**Figure 4-13. Block Diagram of P30**



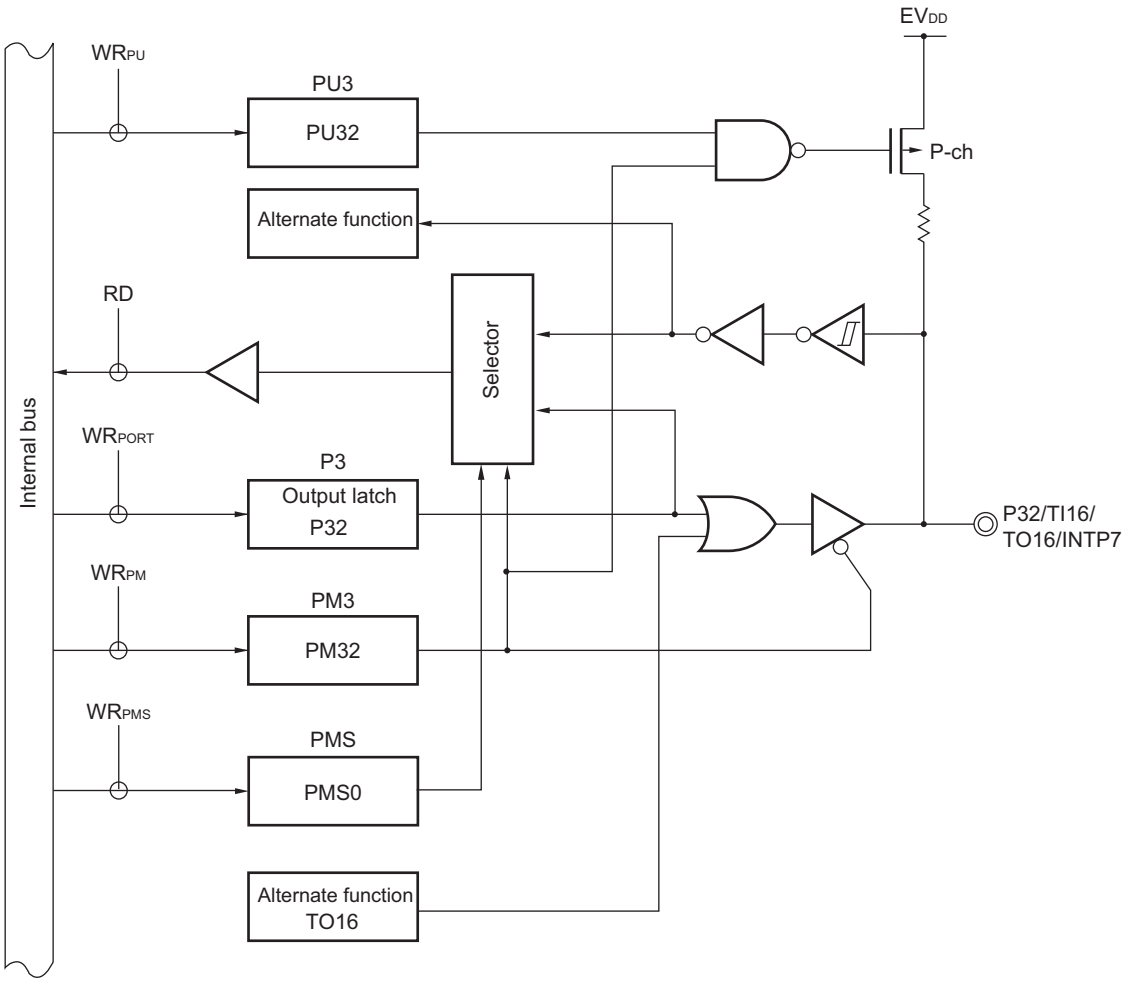

P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
PIM3: Port input mode register 3
PMS: Port mode select register
PITHL3: Port input threshold control register 3
RD: Read signal
WRxx: Write signal

**Figure 4-14. Block Diagram of P31**



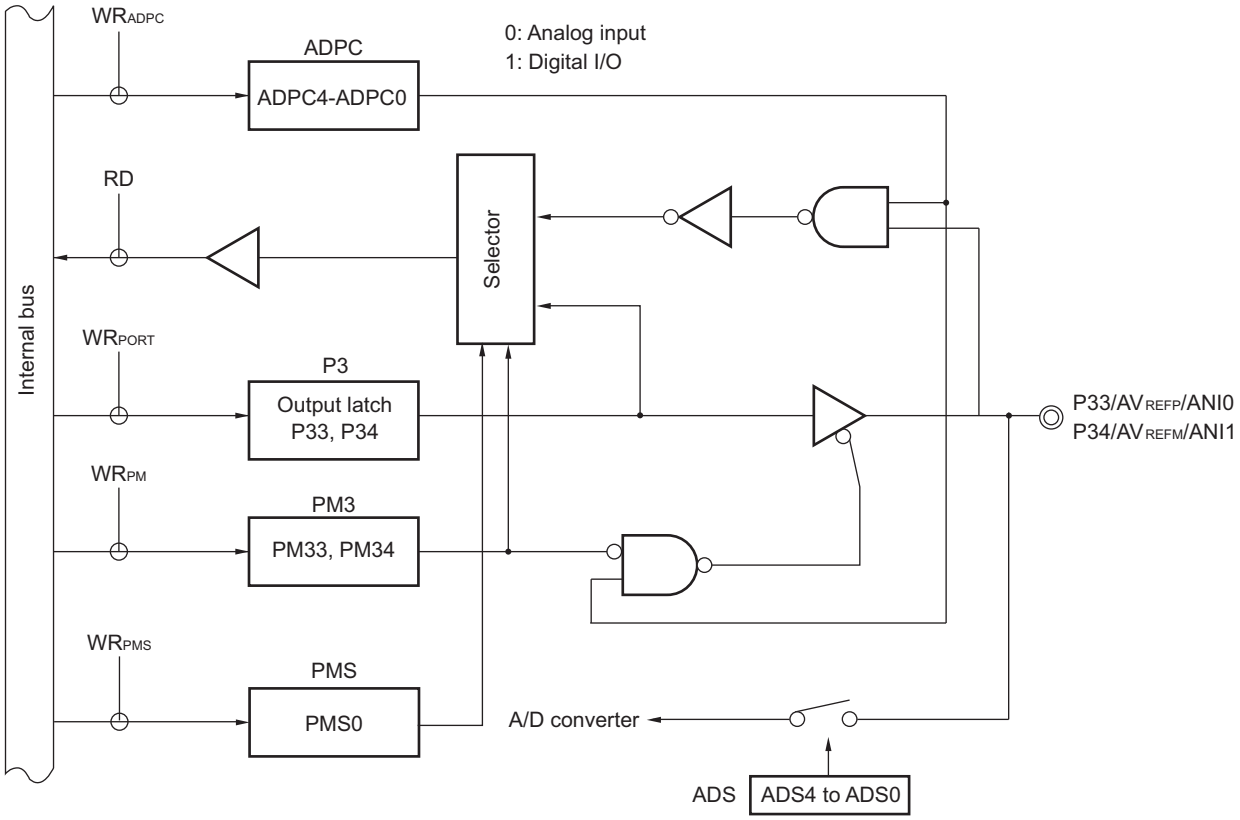

P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-15. Block Diagram of P32**




P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-16. Block Diagram of P33 and P34**




P3: Port register 3
PM3: Port mode register 3
PMS: Port mode select register
ADPC: A/D Port configuration register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

### 4.2.4 Port 4

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 to P47 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).

For the P43 pin input, the threshold of the input buffer can be specified in 1-bit units using the port input threshold control register 4 (PITHL4).

This port can also be used for external interrupt request input, timer I/O, comparator output, SNOOZE status output, LIN serial data I/O, and data I/O for a flash memory programmer/debugger.

Reset signal generation sets this port to input mode.

**Table 4-8. Settings of Registers When Using Port 4**

| Pin name | | PM4x | PITHL4x | Alternate Function Setting Note 6 | Remark |
|---|---|---|---|---|---|
| Name | I/O | | | | |
| P40 | Input | 1 | – | × | |
| | Output | 0 | – | × | |
| P41 | Input | 1 | – | × | |
| | Output | 0 | – | TRJIO0 output = 0 Note 1<br>TO10 output = 0 Note 2<br>VCOUT0 output = 0 Note 3<br>SNZOUT2 output = 0 Note 4 | |
| P42 | Input | 1 | – | × | |
| | Output | 0 | – | (LTXD0 = 1) Note 5 | |
| P43 | Input | 1 | 0 | × | CMOS input<br>(Schmitt1 input) |
| | | | 1 | × | CMOS input<br>(Schmitt3 input) |
| | Output | 0 | × | × | |
| P44 | Input | 1 | – | × | |
| | Output | 0 | – | (TO07 output = 0) Note 2 | |
| P45 | Input | 1 | – | × | |
| | Output | 0 | – | (TO10 output = 0) Note 2 | |
| P46 | Input | 1 | – | × | |
| | Output | 0 | – | (TO12 output = 0) Note 2 | |
| P47 | Input | 1 | – | × | |
| | Output | 0 | – | × | |

**Notes 1.** When a pin sharing a timer input/output function of the timer RJ is to be used as a general-purpose port pin, the TMOD2 to TMOD0 bits of the timer RJ mode register 0 (TRJMR0) must have the same setting as their initial value or have a setting other than 001B.

**2.** When a pin sharing a timer output function of the timer array unit is to be used as a general-purpose port pin, the TOmn bit of the timer output register m (TOm) and the TOEmn bit of the timer output enable register m (TOEm) corresponding to the target unit and channel must have the same setting as in the initial state (m = 0, 1, n = 0 to 7).

**3.** When a pin sharing the comparator output function is to be used as a general-purpose port pin, the COE bit of the comparator control register (CMPCTL) must have the same setting as its initial value.

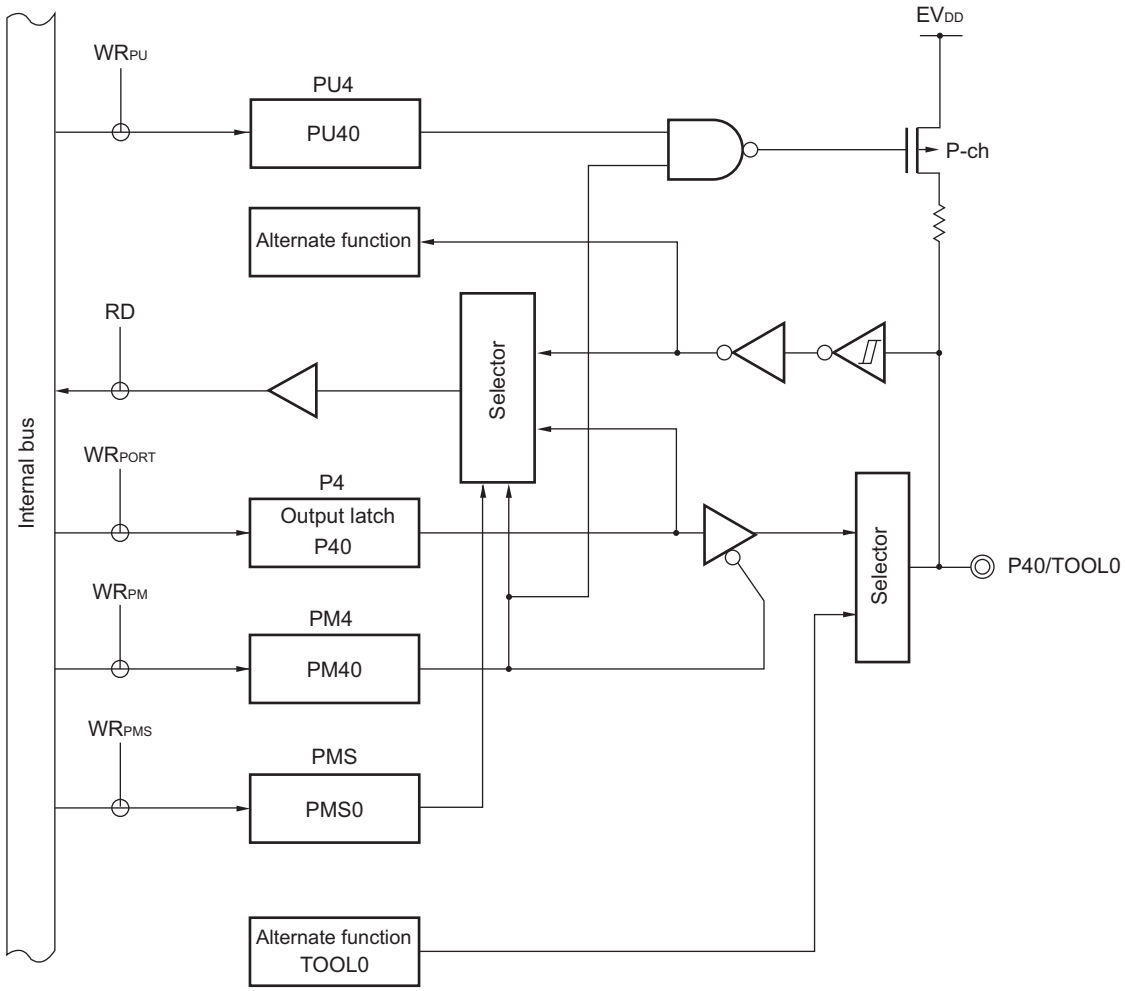
**4.** When a pin sharing the SNOOZE status output function is to be used as a general-purpose port pin, the OUTEN0 to OUTEN7 bits of the SNOOZE status output control registers 0, 1, 2, 3 (PSNZCNT0, 1, 2, 3) must have the same setting as its initial value.

5. When a pin sharing the serial data output function of the LIN is to be used as a general-purpose port pin, operation of the corresponding LIN must be stopped.
6. Functions in parentheses can be assigned via settings in the peripheral I/O redirection registers 1, 3, 4 (PIOR1, PIOR3, PIOR4).

**Remark** ×: Don't care
PM4x: Port mode register 4
PITHL4x: Port input threshold control register 4

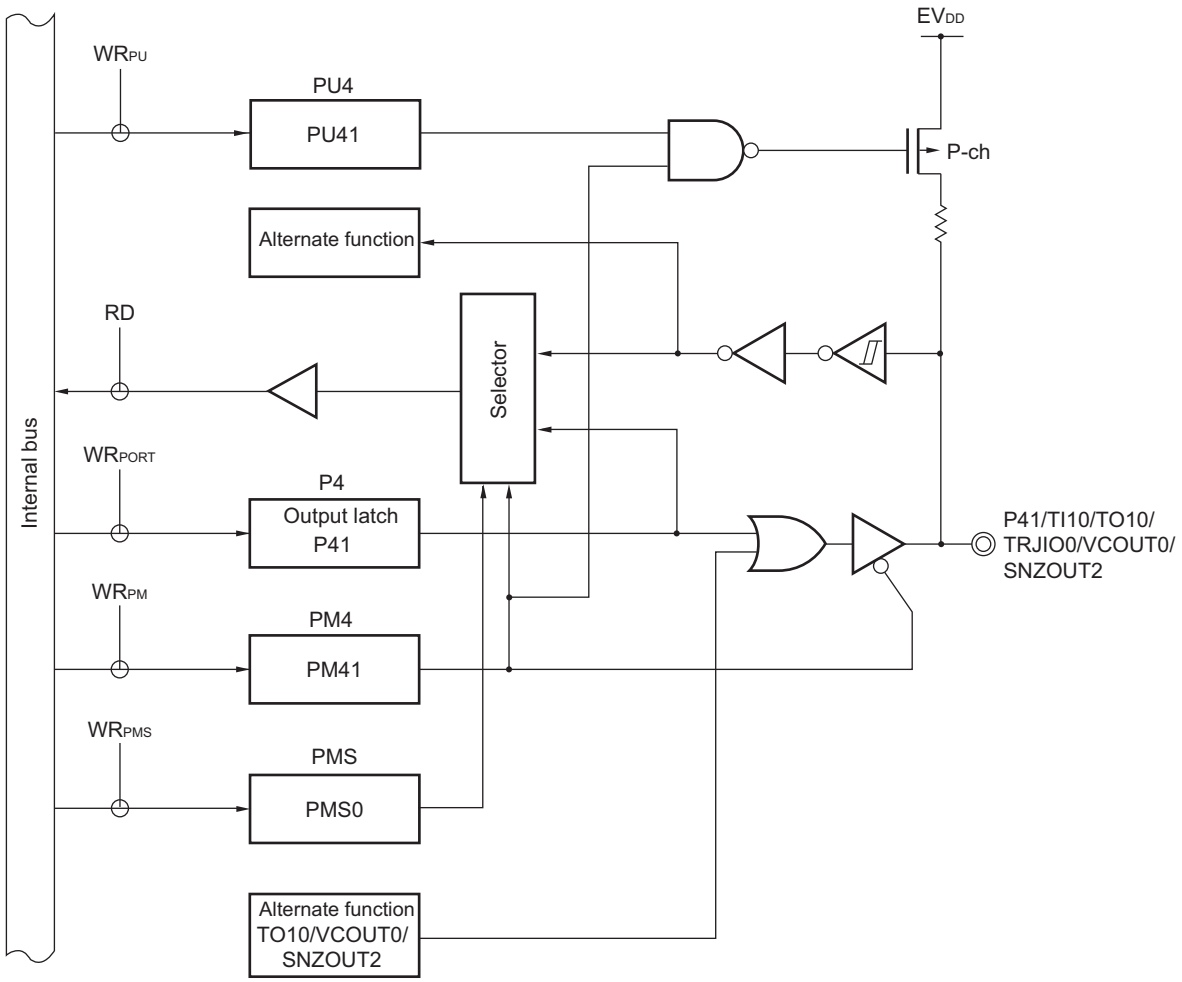
Figures 4-17 to 4-24 show block diagrams of port 4 for 100-pin products.

**Figure 4-17. Block Diagram of P40**



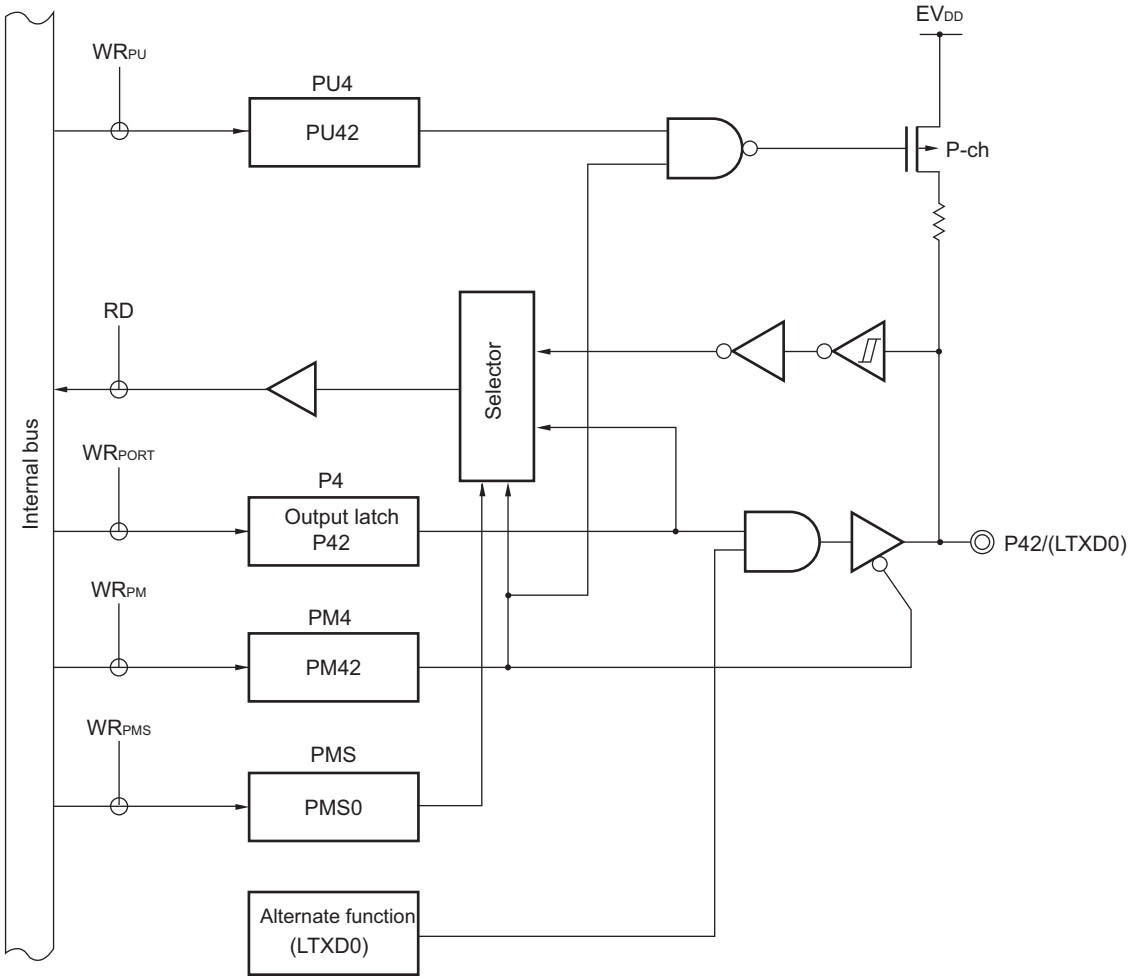

P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-18. Block Diagram of P41**



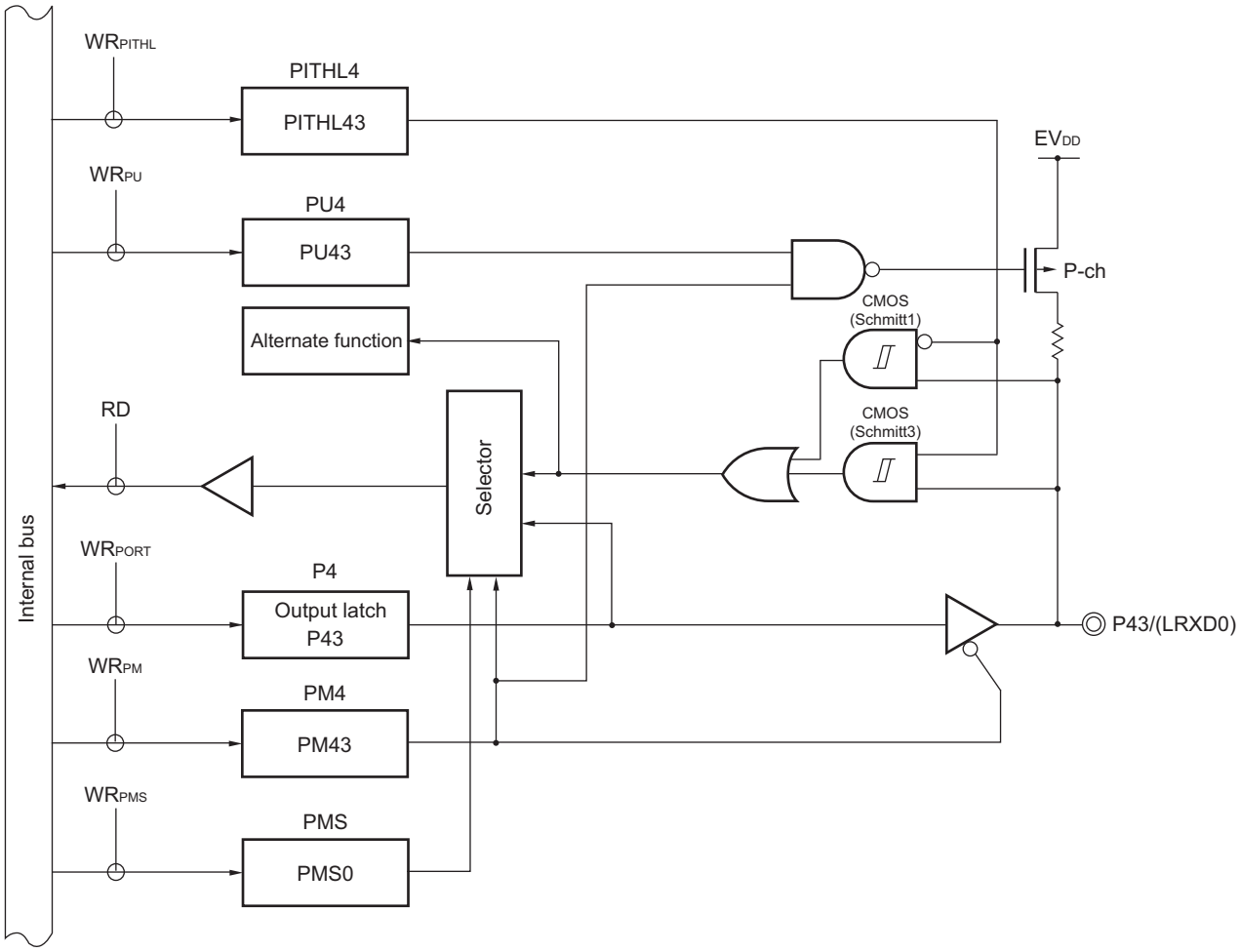

P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-19. Block Diagram of P42**



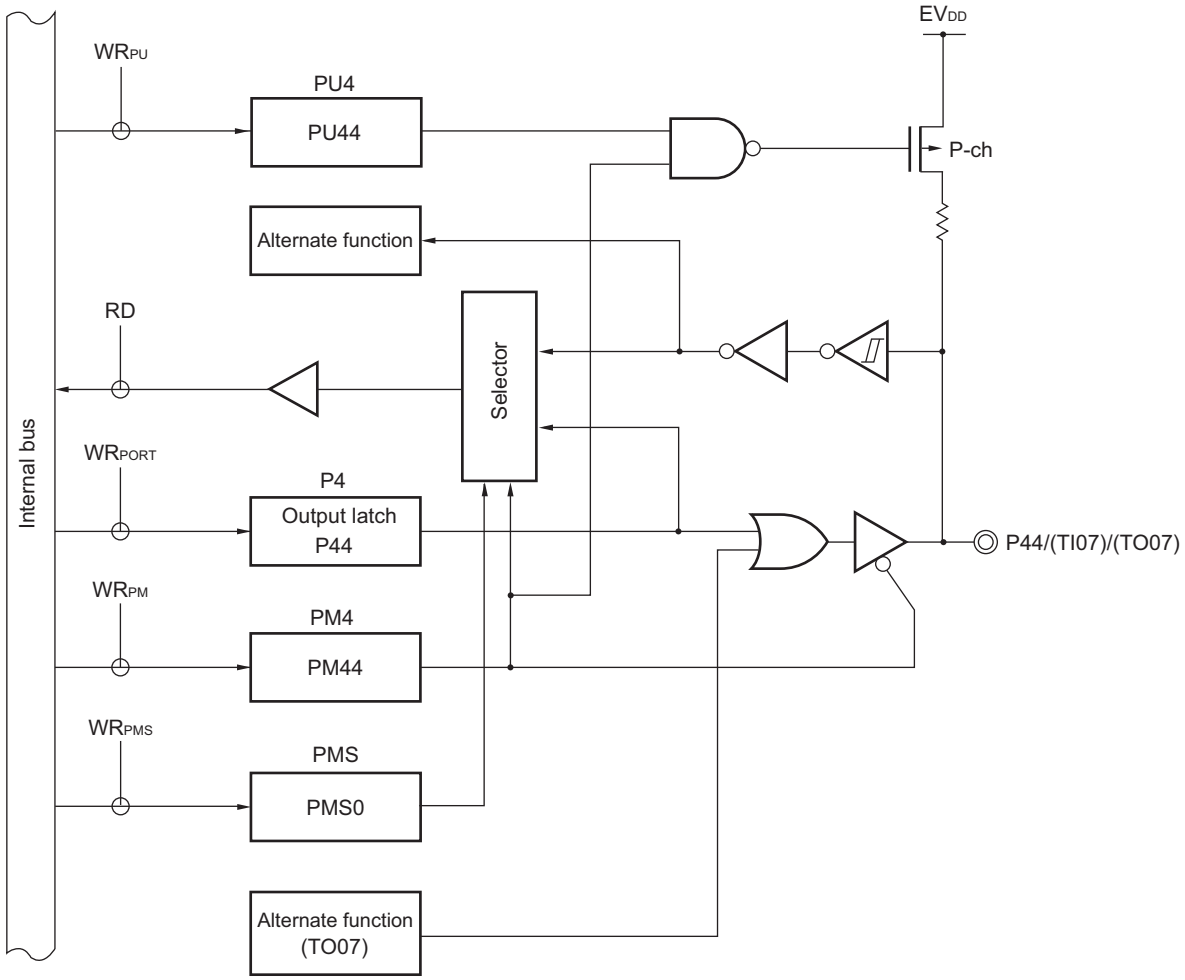

P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-20. Block Diagram of P43**



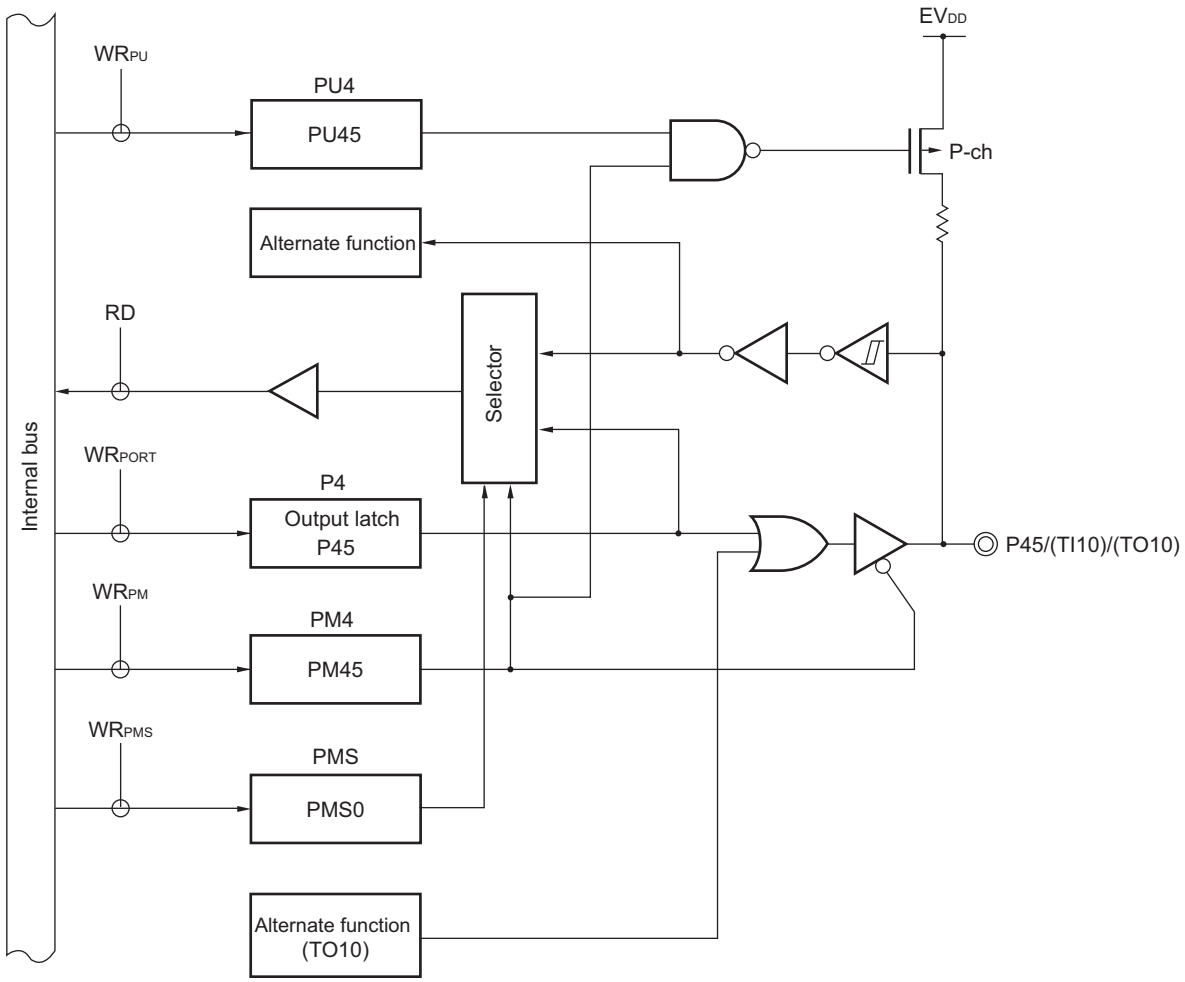

P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
PITHL4: Port input threshold control register 4
RD: Read signal
WRxx: Write signal

**Figure 4-21. Block Diagram of P44**



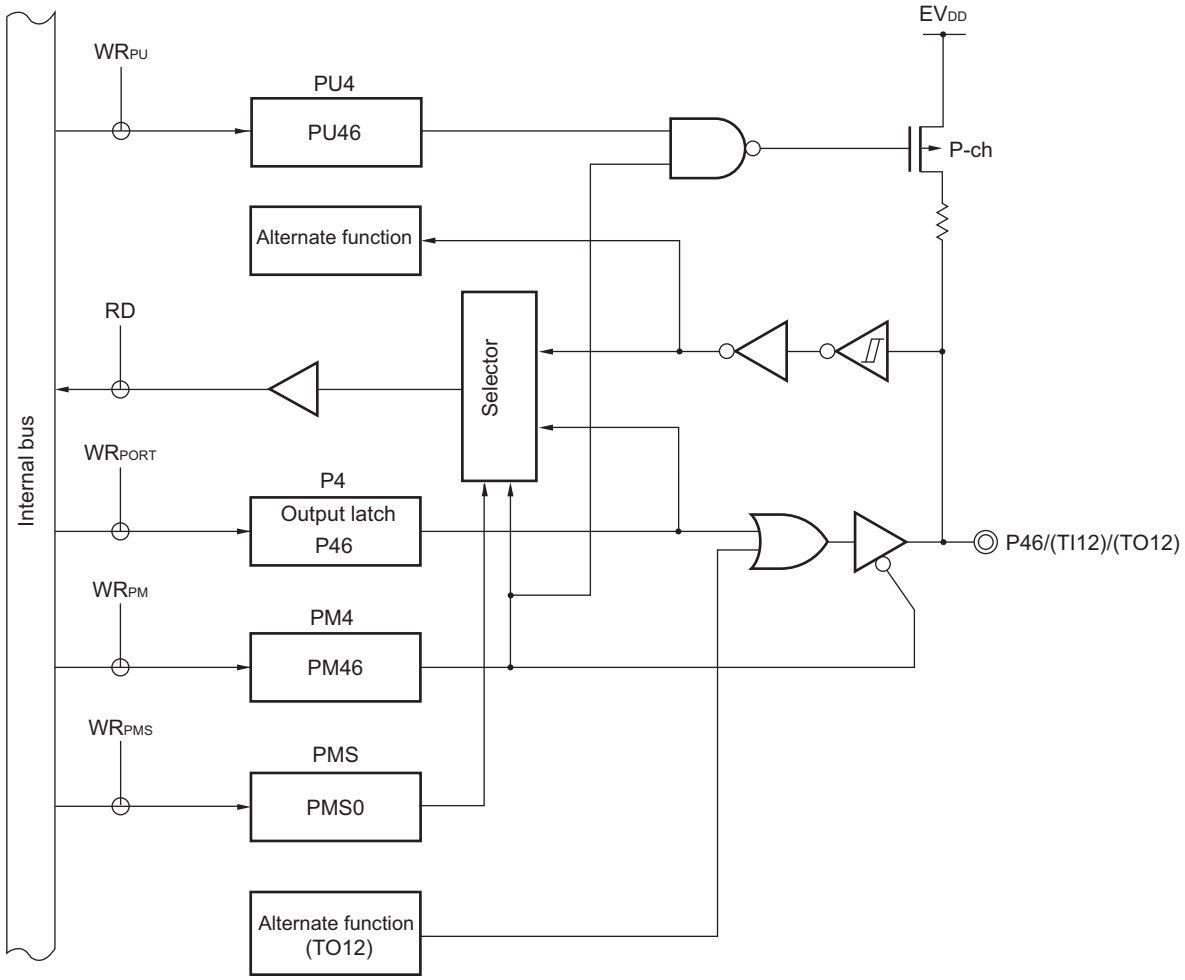

P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-22. Block Diagram of P45**



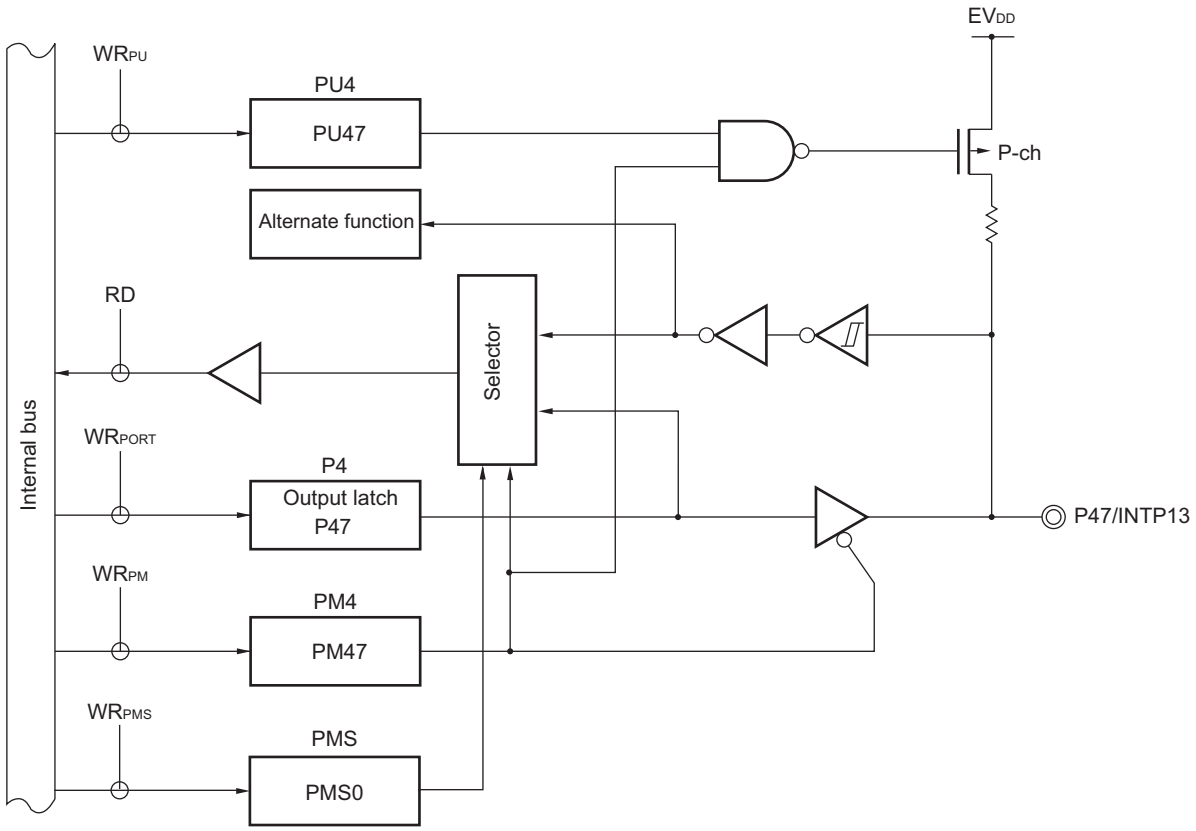

P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-23. Block Diagram of P46**




P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-24. Block Diagram of P47**




P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

### 4.2.5 Port 5

Port 5 is an I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P57 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

Input to the P54 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 5 (PIM5).

For the P50 and P52 to P54 pin input, the threshold of the input buffer can be specified in 1-bit units using the port input threshold control register 5 (PITHL5).

This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, slave select input, timer I/O, STOP status output, and SNOOZE status output.

Reset signal generation sets this port to input mode.

**Table 4-9. Settings of Registers When Using Port 5 (1/2)**

| Pin Name | | PM5x | PIM5x | PITHL5x | Alternate Function Setting Note 6 | Remark |
|---|---|---|---|---|---|---|
| Name | I/O | | | | | |
| P50 | Input | 1 | – | 0 | × | CMOS input (Schmitt1 input) |
| | | | | 1 | | CMOS input (Schmitt3 input) |
| | Output | 0 | – | × | × | |
| P51 | Input | 1 | – | – | × | |
| | Output | 0 | – | – | (SO01 output = 1) Note 1 | |
| P52 | Input | 1 | – | 0 | × | CMOS input (Schmitt1 input) |
| | | | | 1 | | CMOS input (Schmitt3 input) |
| | Output | 0 | – | × | ($\overline{\text{SCK01}}$ output = 1) Note 2<br>(STOPST output = 0) Note 5 | |
| P53 | Input | 1 | – | 0 | × | CMOS input (Schmitt1 input) |
| | | | | 1 | | CMOS input (Schmitt3 input) |
| | Output | 0 | – | × | × | |
| P54 | Input | 1 | 0 | 0 | × | CMOS input (Schmitt1 input) |
| | | | | 1 | | CMOS input (Schmitt3 input) |
| | | | 1 | × | × | TTL input |
| | Output | 0 | × | × | (TO11 output = 0) Note 3 | |

(**Notes** and **Remark** are listed on the next page.)

**Table 4-9. Settings of Registers When Using Port 5 (2/2)**

| Pin Name | | PM5x | PIM5x | PITHL5x | Alternate Function Setting Note 5 | Remark |
|---|---|---|---|---|---|---|
| Name | I/O | | | | | |
| P55 | Input | 1 | – | – | × | |
| | Output | 0 | – | – | (TO13 output = 0) Note 3 | |
| P56 | Input | 1 | – | – | × | |
| | Output | 0 | – | – | (TO15 output = 0) Note 3<br>(SNZOUT1 output = 0) Note 4 | |
| P57 | Input | 1 | – | – | × | |
| | Output | 0 | – | – | (TO17 output = 0) Note 3<br>(SNZOUT0 output = 0) Note 4 | |

**Notes 1.** When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn bit of the serial output register m (SOm), the SOEmn bit of the serial output enable register m (SOEm), and the SEmn bit of the serial channel enable status register m (SEm) corresponding to the target unit and channel must have the same setting as its initial value (m = 0, 1, n = 0, 1).

**2.** When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the CKOmn bit of the serial output register m (SOm), the SOEmn bit of the serial output enable register m (SOEm), and the SEmn bit of the serial channel enable status register m (SEm) corresponding to the target unit and channel must have the same setting as its initial value (m = 0, 1, n = 0, 1).

**3.** When a pin sharing a timer output function of the timer array unit is to be used as a general-purpose port pin, the TOmn bit of the timer output register m (TOm) and the TOEmn bit of the timer output enable register m (TOEm) corresponding to the target unit and channel must have the same setting as in the initial state (m = 0, 1, n = 0 to 7).

**4.** When a pin sharing the SNOOZE status output function is to be used as a general-purpose port pin, the OUTEN0 to OUTEN7 bits of the SNOOZE status output control registers 0, 1, 2, 3 (PSNZCNT0, 1, 2, 3) must have the same setting as its initial value.

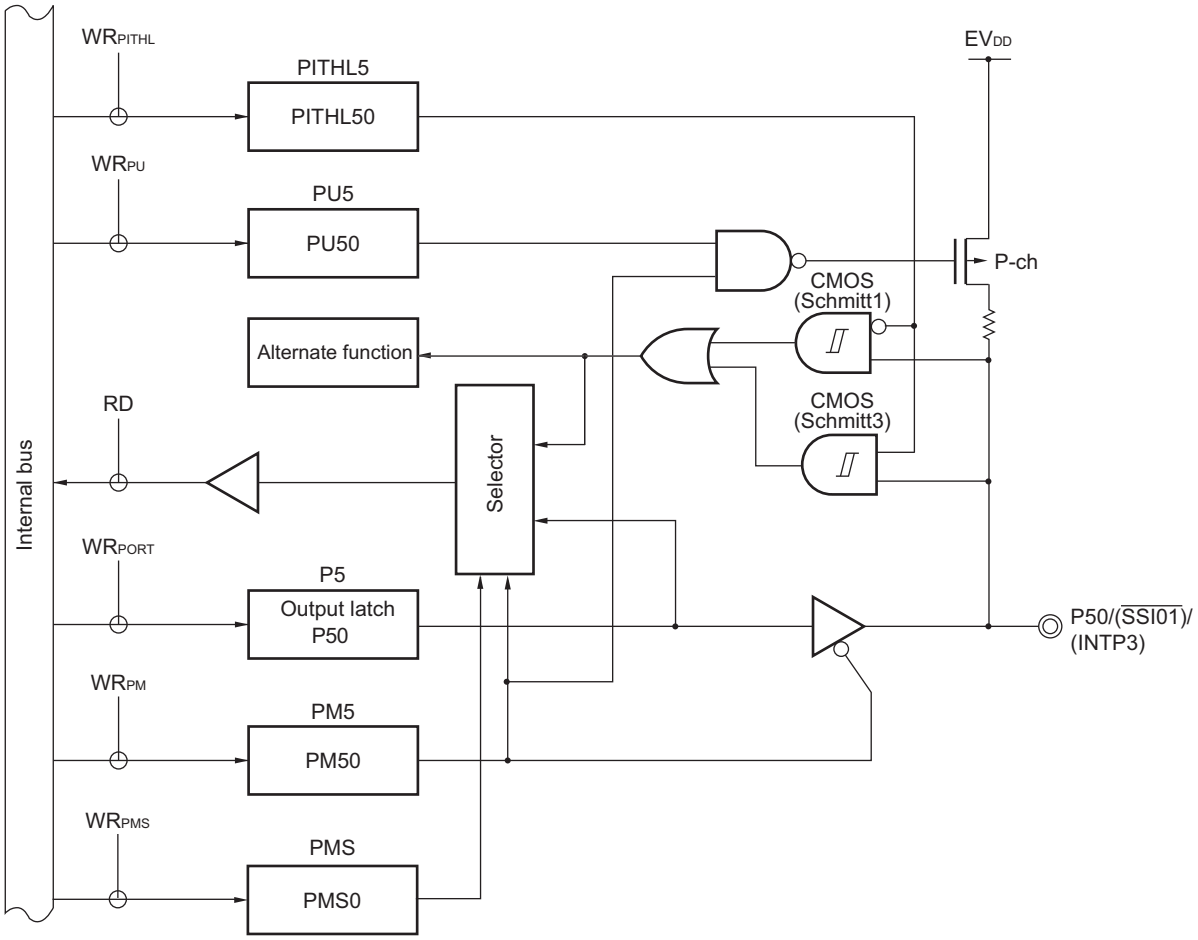
**5.** When a pin sharing the STOP status output function is to be used as a general-purpose port pin, the STPOEN bit of the STOP status output control register (STPSTC) must have the same setting as its initial value.

**6.** Functions in parentheses can be assigned via settings in the peripheral I/O redirection registers 3, 4, 6 (PIOR3, PIOR4, PIOR6). The STOPST function can be assigned via settings in the STOP status output control register (STPSTC).

**Remark** ×: Don't care
PM5x: Port mode register 5
PIM5x: Port input mode register 5
PITHL5x: Port input threshold control register 5

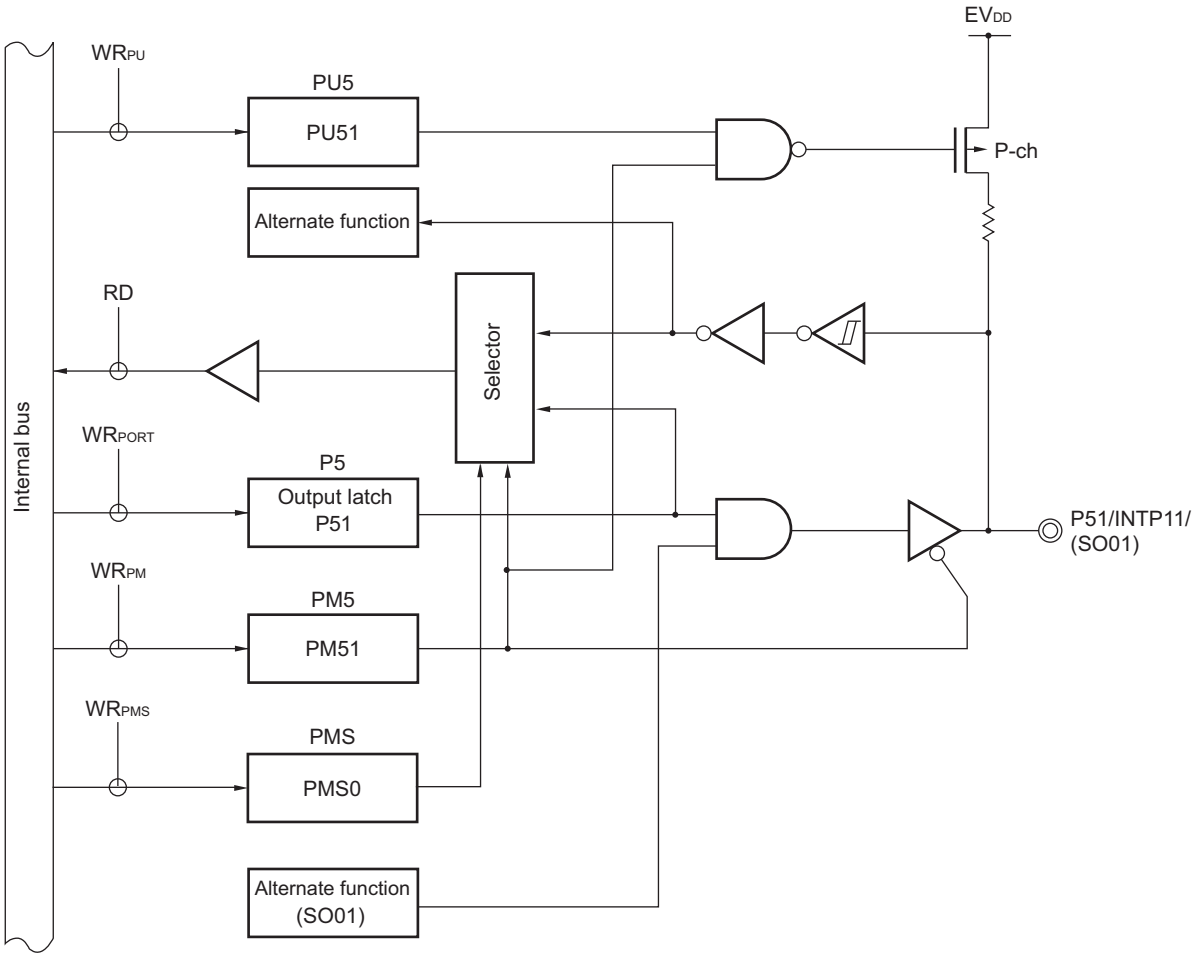
Figures 4-25 to 4-32 show block diagrams of port 5 for 100-pin products.

**Figure 4-25. Block Diagram of P50**



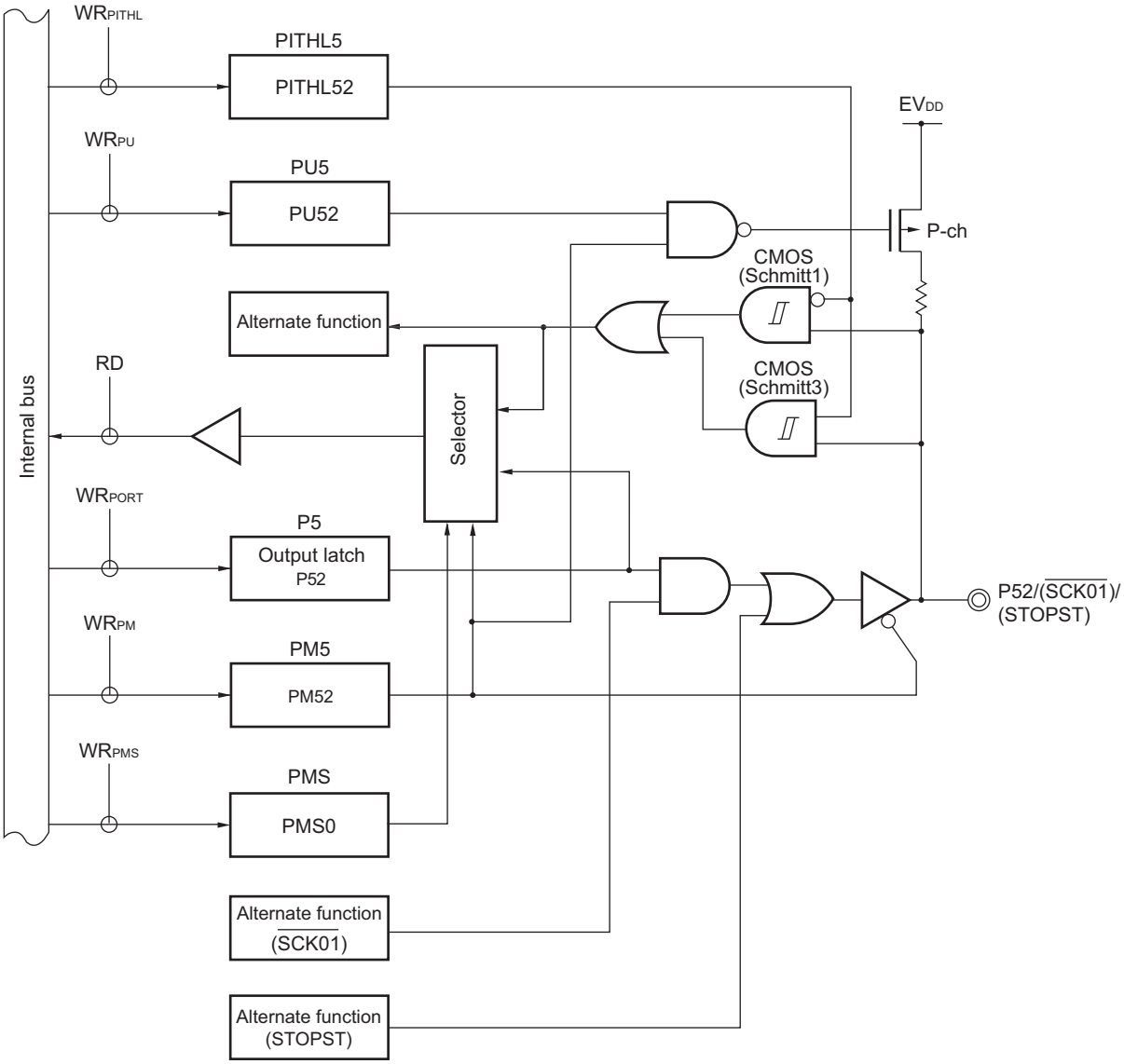

P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PMS: Port mode select register
PITHL5: Port input threshold control register 5
RD: Read signal
WRxx: Write signal

**Figure 4-26. Block Diagram of P51**



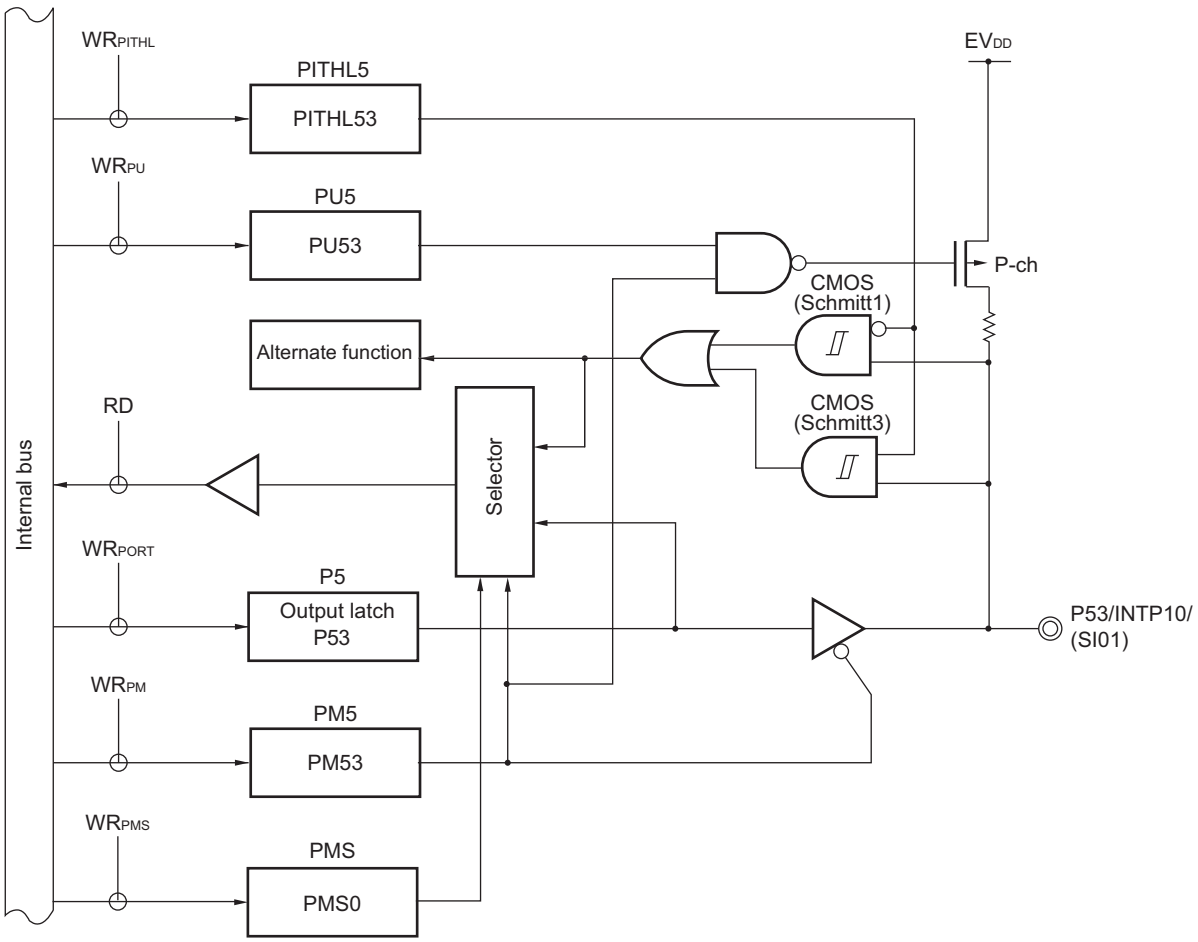

P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-27. Block Diagram of P52**



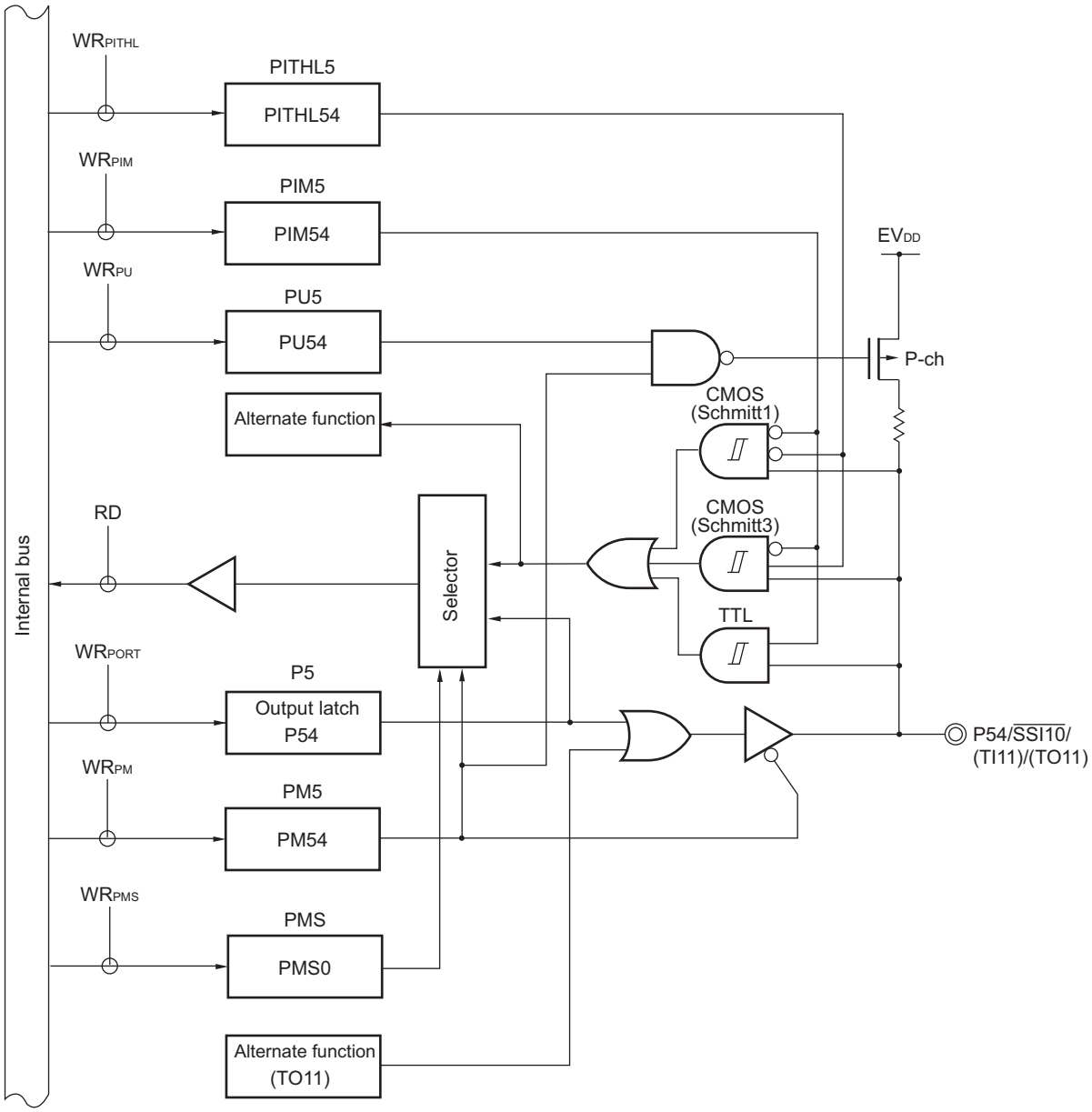

P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PMS: Port mode select register
PITHL5: Port input threshold control register 5
RD: Read signal
WRxx: Write signal

**Figure 4-28. Block Diagram of P53**



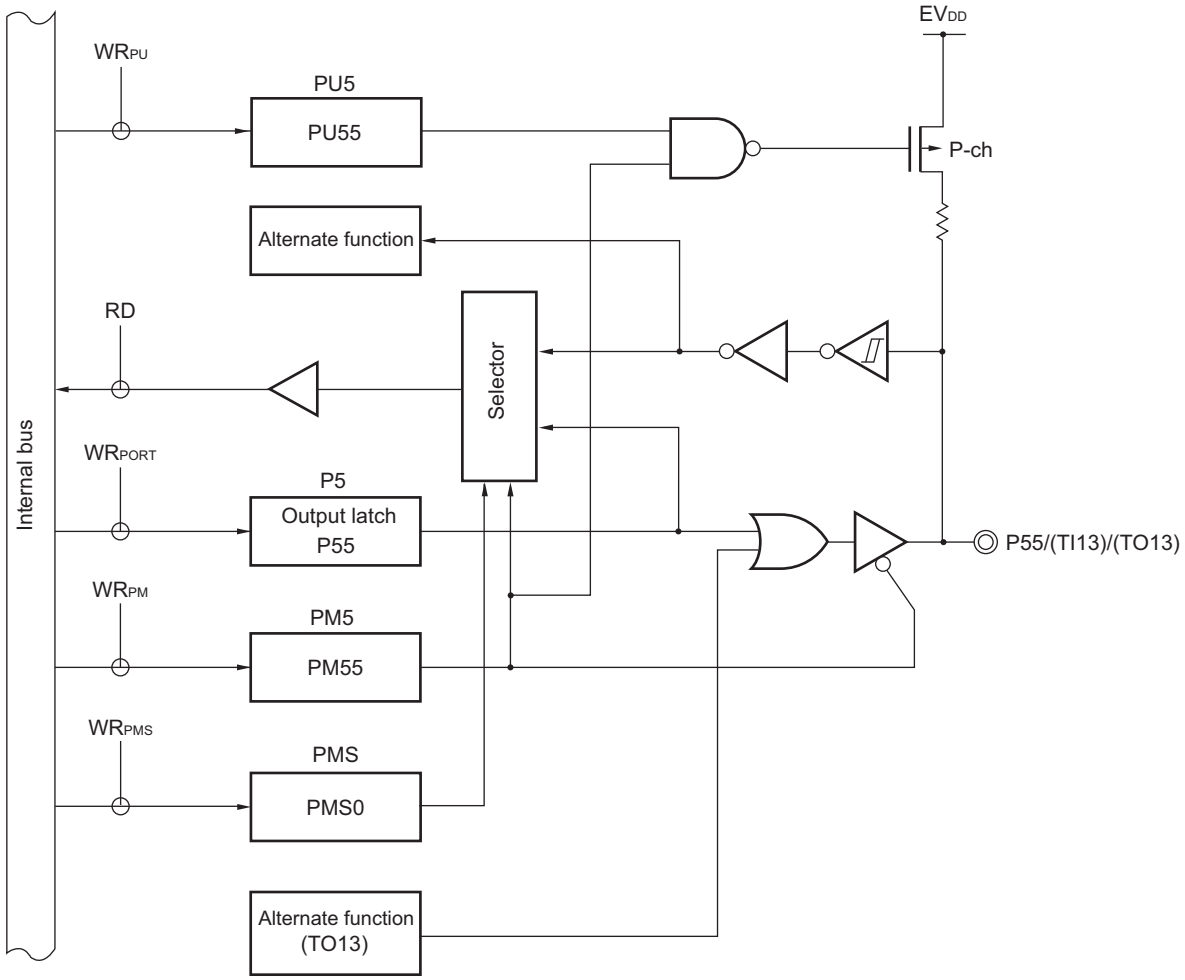

P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PMS: Port mode select register
PITHL5: Port input threshold control register 5
RD: Read signal
WRxx: Write signal

**Figure 4-29. Block Diagram of P54**



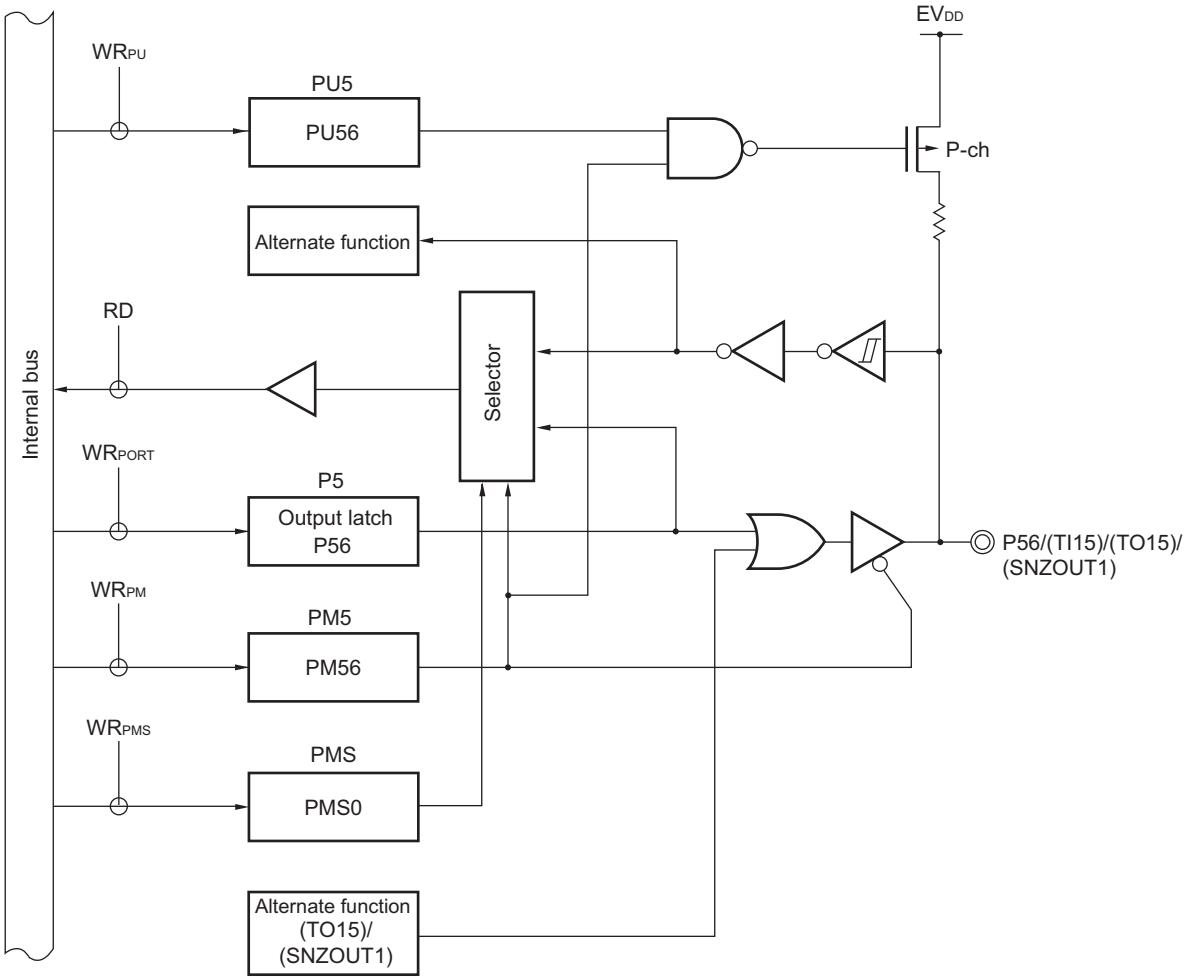

P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PIM5: Port input mode register 5
PMS: Port mode select register
PITHL5: Port input threshold control register 5
RD: Read signal
WRxx: Write signal

**Figure 4-30. Block Diagram of P55**



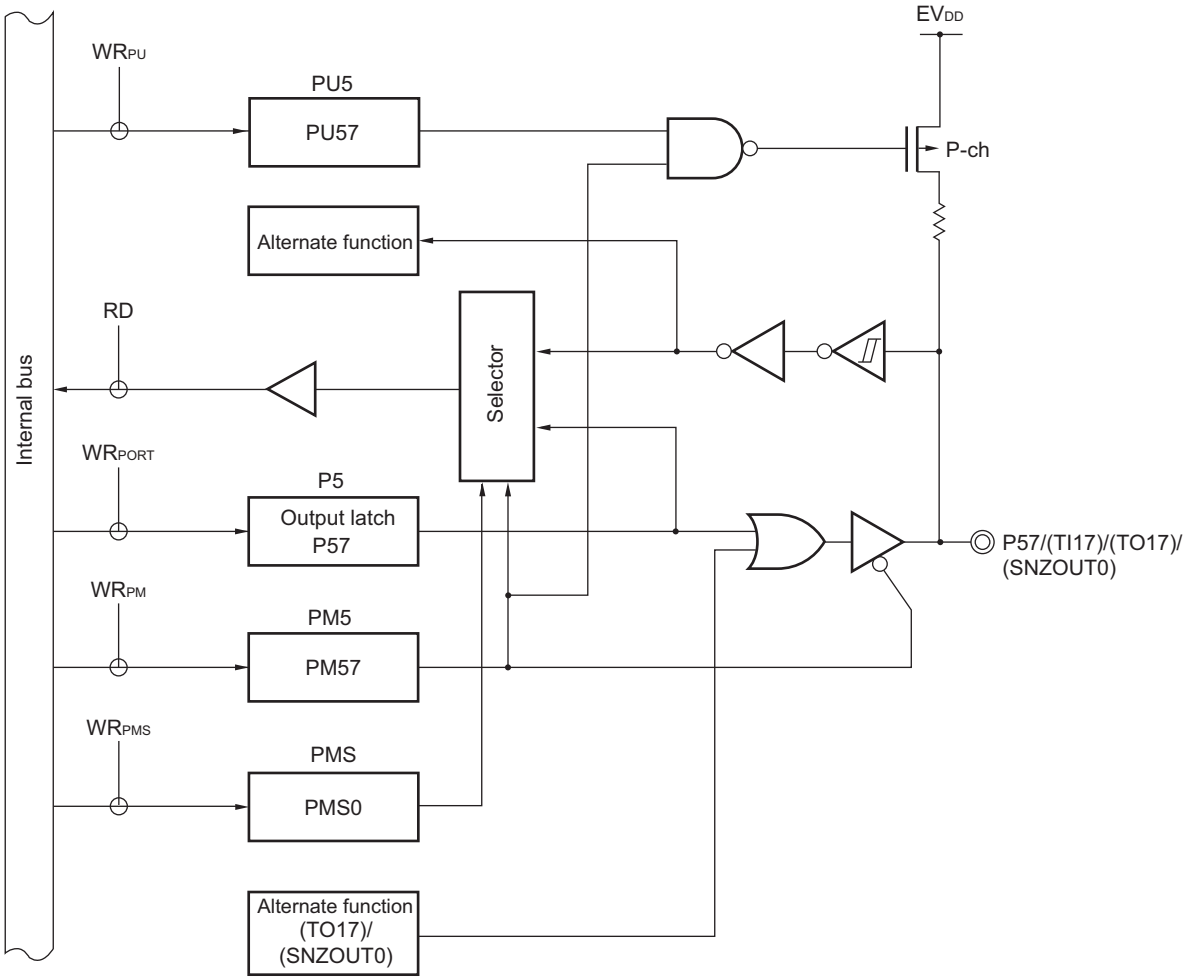

P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-31. Block Diagram of P56**




P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-32. Block Diagram of P57**




P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

### 4.2.6 Port 6

Port 6 is an I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6). Input to the P62 and P63 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 6 (PIM6). When the P60 to P67 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 6 (PU6).

Output from the P60 to P63 pins can be specified as N-ch open-drain output ($EV_{DD}$ tolerance) in 1-bit units using port output mode register 6 (POM6).

For the P60 to P63 pin input, the threshold of the input buffer can be specified in 1-bit units using the port input threshold control register 6 (PITHL6)

This port can also be used for serial interface data I/O and clock I/O, slave select input, timer I/O, and SNOOZE status output.

Reset signal generation sets this port to input mode.

**Table 4-10. Settings of Registers When Using Port 6 (1/2)**

| Pin Name | | PM6x | PIM6x | POM6x | PITHL6x | Alternate Function Setting Note 6 | Remark |
|---|---|---|---|---|---|---|---|
| Name | I/O | | | | | | |
| P60 | Input | 1 | – | × | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | 1 | | CMOS input (Schmitt3 input) |
| | Output | 0 | – | 0 | × | ($\overline{SCK00}$/SCL00 output = 1) Note 1 | CMOS output |
| | | | | 1 | × | | N-ch O.D output |
| P61 | Input | 1 | – | × | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | 1 | | CMOS input (Schmitt3 input) |
| | Output | 0 | – | 0 | × | SDA00 output = 1 Note 2 | CMOS output |
| | | | | 1 | × | | N-ch O.D output |
| P62 | Input | 1 | 0 | × | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | 1 | | CMOS input (Schmitt3 input) |
| | | | 1 | × | × | × | TTL input |
| | Output | 0 | × | 0 | × | SCLA0 output = 0 Note 3 | CMOS output |
| | | | | 1 | × | (SO00/TXD0 output = 1) Note 2 | N-ch O.D output |

(**Notes** and **Remark** are listed on the next page.)

**Table 4-10. Settings of Registers When Using Port 6 (2/2)**

| Pin Name | | PM6x | PIM6x | POM6x | PITHL6x | Alternate Function Setting [Note 6] | Remark |
|---|---|---|---|---|---|---|---|
| Name | I/O | | | | | | |
| P63 | Input | 1 | 0 | × | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | 1 | | CMOS input (Schmitt3 input) |
| | | | 1 | × | × | × | TTL input |
| | Output | 0 | × | 0 | × | SDAA0 output = 0 [Note 3] | CMOS output |
| | | | | 1 | × | | N-ch O.D output |
| P64 | Input | 1 | – | – | – | × | |
| | Output | 0 | – | – | – | (TO14 output = 0) [Note 4] (SNZOUT3 output = 0) [Note 5] | |
| P65 | Input | 1 | – | – | – | × | |
| | Output | 0 | – | – | – | (TO16 output = 0) [Note 4] (SNZOUT2 output = 0) [Note 5] | |
| P66 | Input | 1 | – | – | – | × | |
| | Output | 0 | – | – | – | (TO00 output = 0) [Note 4] | |
| P67 | Input | 1 | – | – | – | × | |
| | Output | 0 | – | – | – | (TO02 output = 0) [Note 4] | |

**Notes 1.** When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the CKOmn bit of the serial output register m (SOm), the SOEmn bit of the serial output enable register m (SOEm), and the SEmn bit of the serial channel enable status register m (SEm) corresponding to the target unit and channel must have the same setting as its initial value (m = 0, 1, n = 0, 1).

**2.** When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn bit of the serial output register m (SOm), the SOEmn bit of the serial output enable register m (SOEm), and the SEmn bit of the serial channel enable status register m (SEm) corresponding to the target unit and channel must have the same setting as its initial value (m = 0, 1, n = 0, 1).

**3.** When a pin sharing the serial interface IICA function is to be used as a general-purpose port pin, operation of the corresponding serial interface IICA must be stopped.

**4.** When a pin sharing a timer output function of the timer array unit is to be used as a general-purpose port pin, the TOmn bit of the timer output register m (TOm) and the TOEmn bit of the timer output enable register m (TOEm) corresponding to the target unit and channel must have the same setting as in the initial state (m = 0, 1, n = 0 to 7).

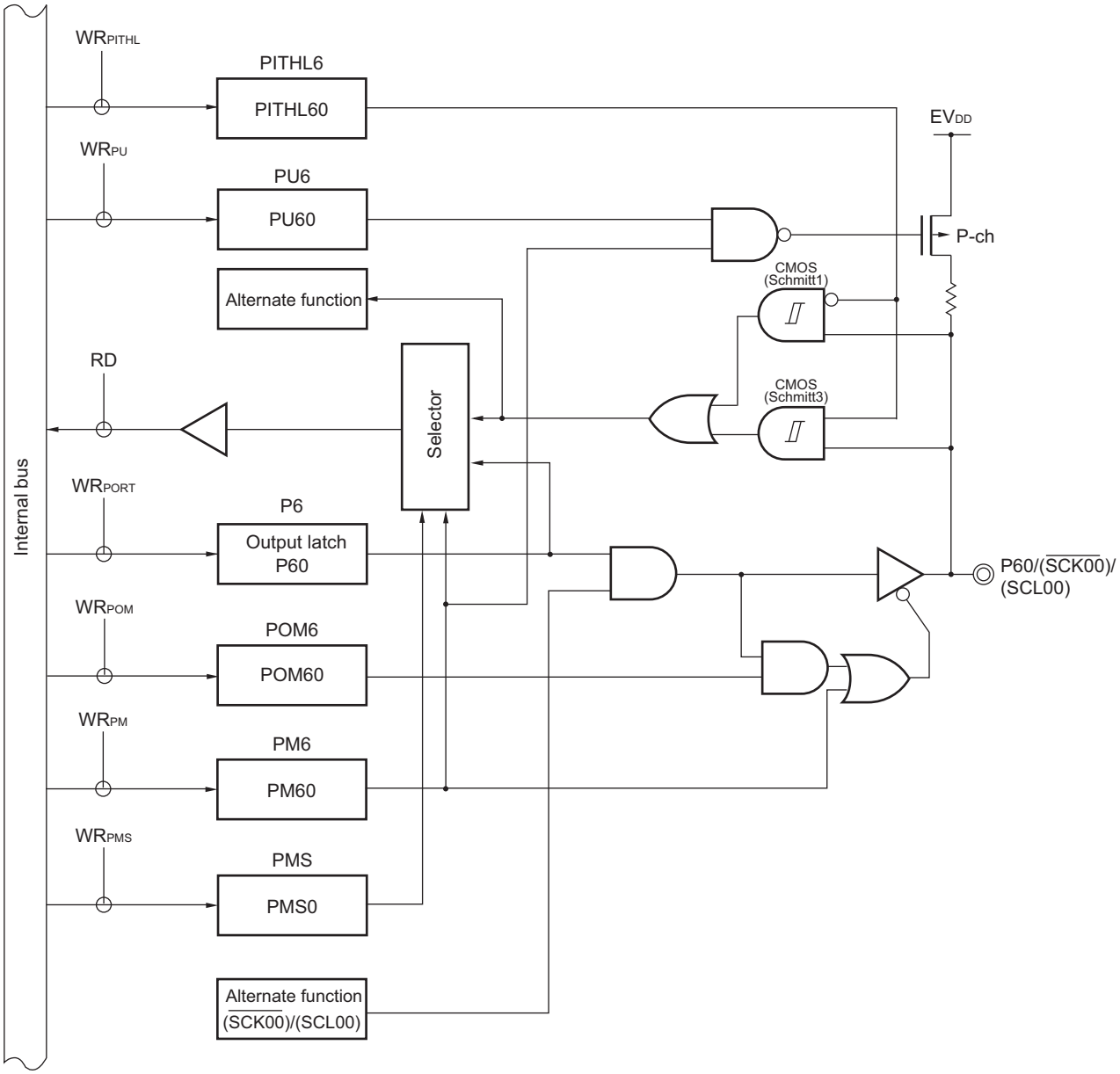
**5.** When a pin sharing the SNOOZE status output function is to be used as a general-purpose port pin, the OUTEN0 to OUTEN7 bits of the SNOOZE status output control registers 0, 1, 2, 3 (PSNZCNT0, 1, 2, 3) must have the same setting as its initial value.

**6.** Functions in parentheses can be assigned via settings in the peripheral I/O redirection registers 1, 3, 4, 6 (PIOR1, PIOR3, PIOR4, PIOR6).

**Remark** ×: Don't care
PM6x: Port mode register 6
PIM6x: Port input mode register 6
POM6x: Port output mode register 6
PITHL6x: Port input threshold control register 6

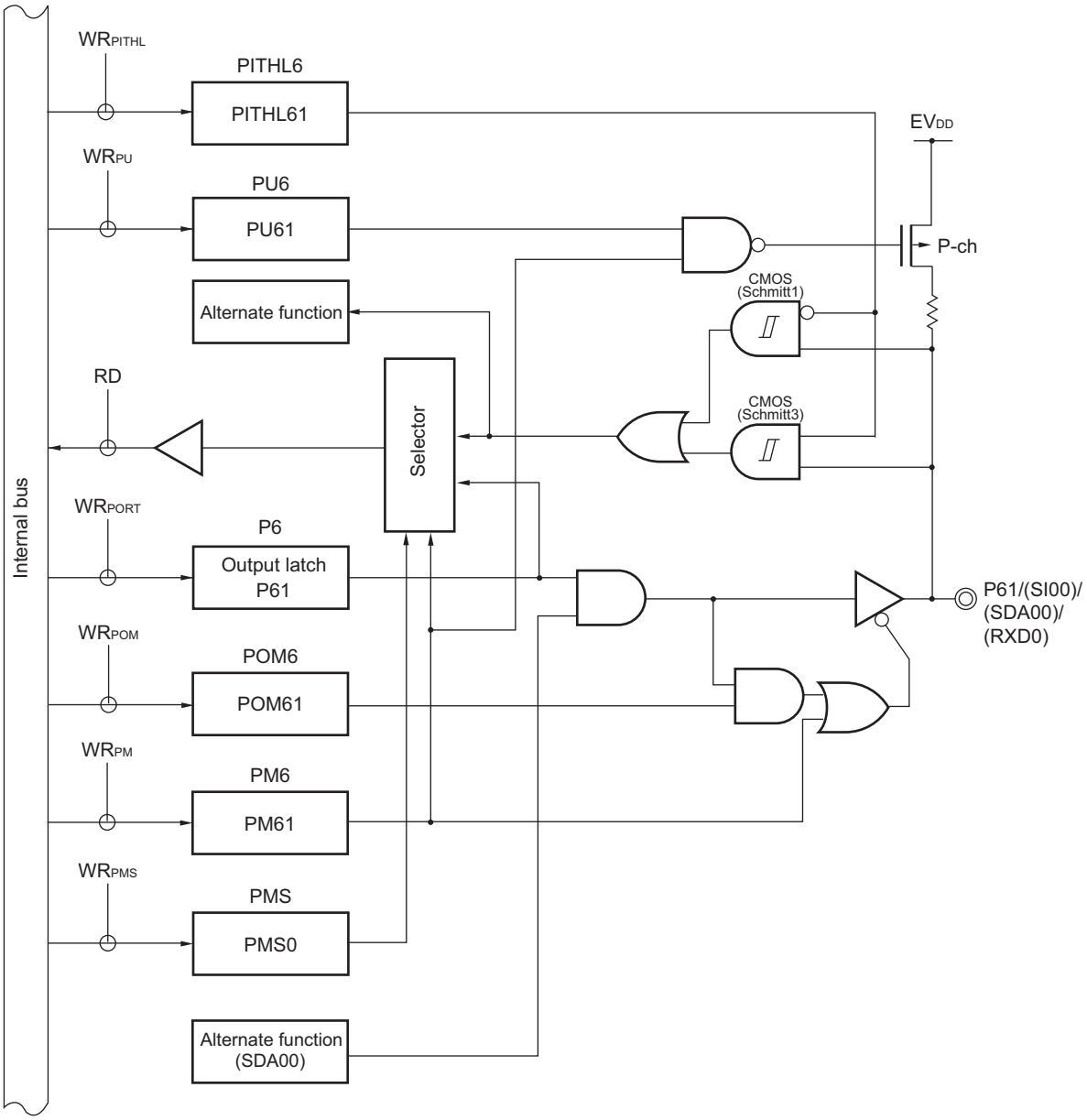
Figures 4-33 to 4-40 show block diagrams of port 6 for 100-pin products.

**Figure 4-33. Block Diagram of P60**



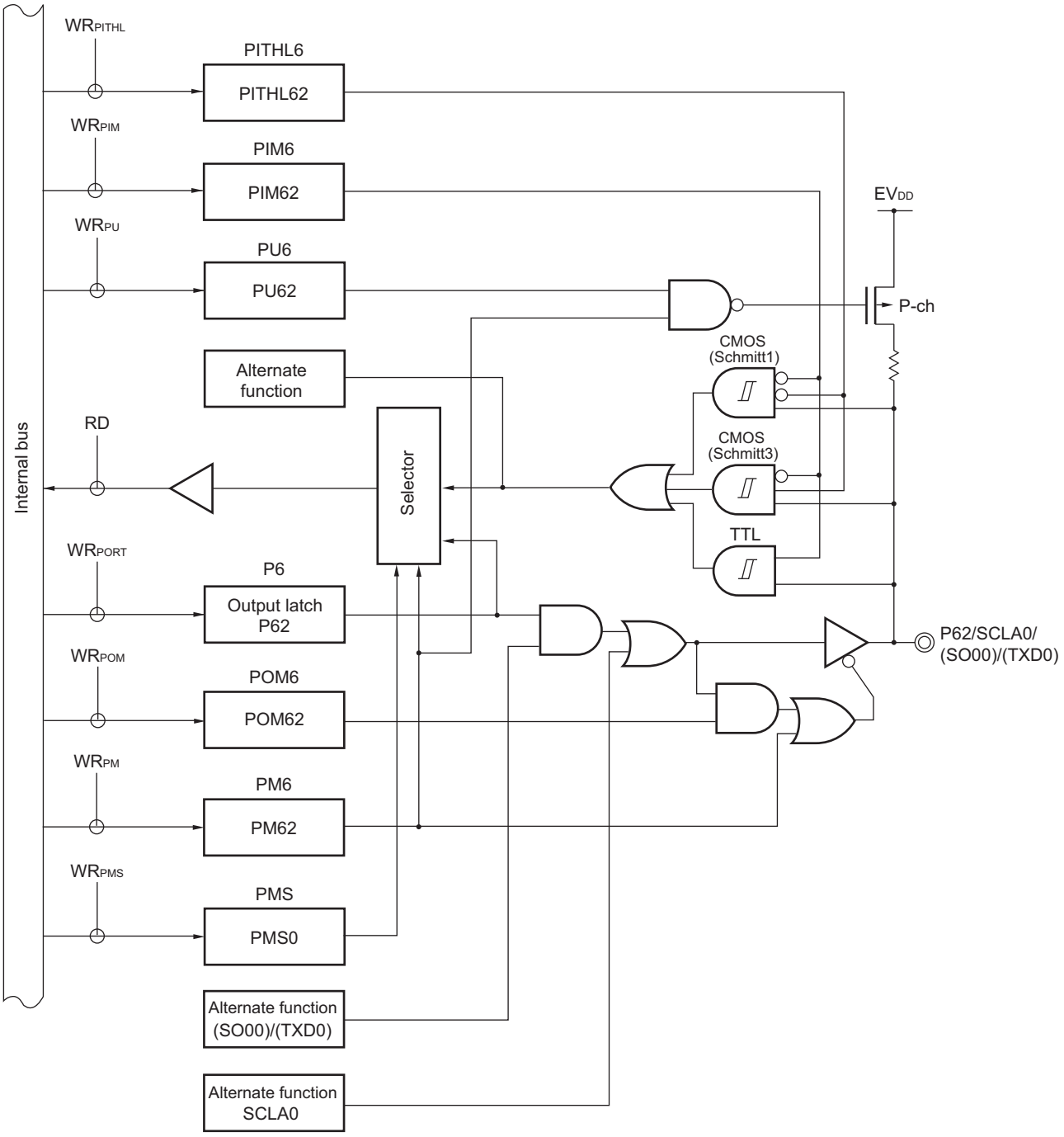

P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
POM6: Port output mode register 6
PMS: Port mode select register
PITHL6: Port input threshold control register 6
RD: Read signal
WRxx: Write signal

**Figure 4-34. Block Diagram of P61**



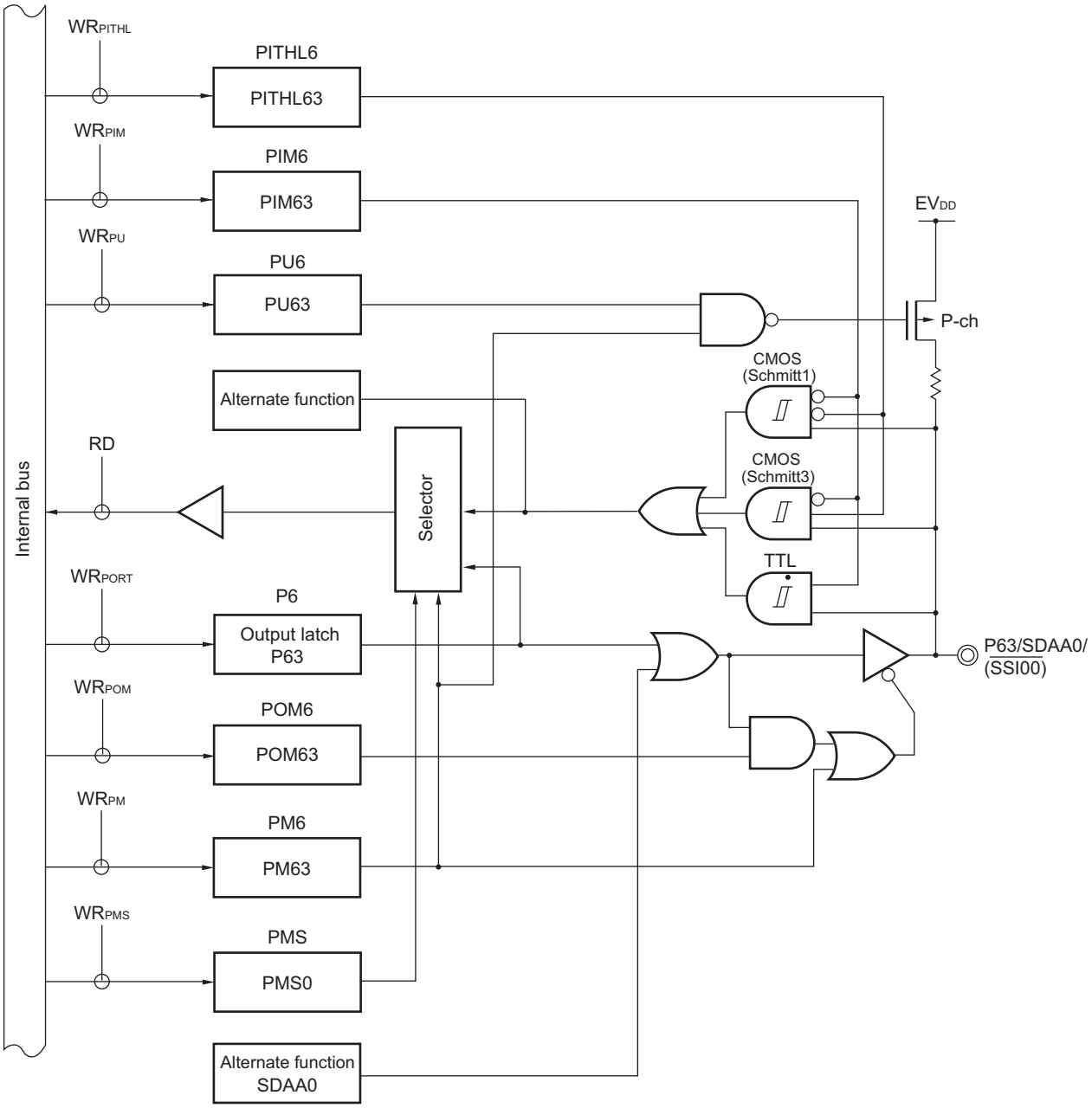

P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
POM6: Port output mode register 6
PMS: Port mode select register
PITHL6: Port input threshold control register 6
RD: Read signal
WRxx: Write signal

**Figure 4-35. Block Diagram of P62**



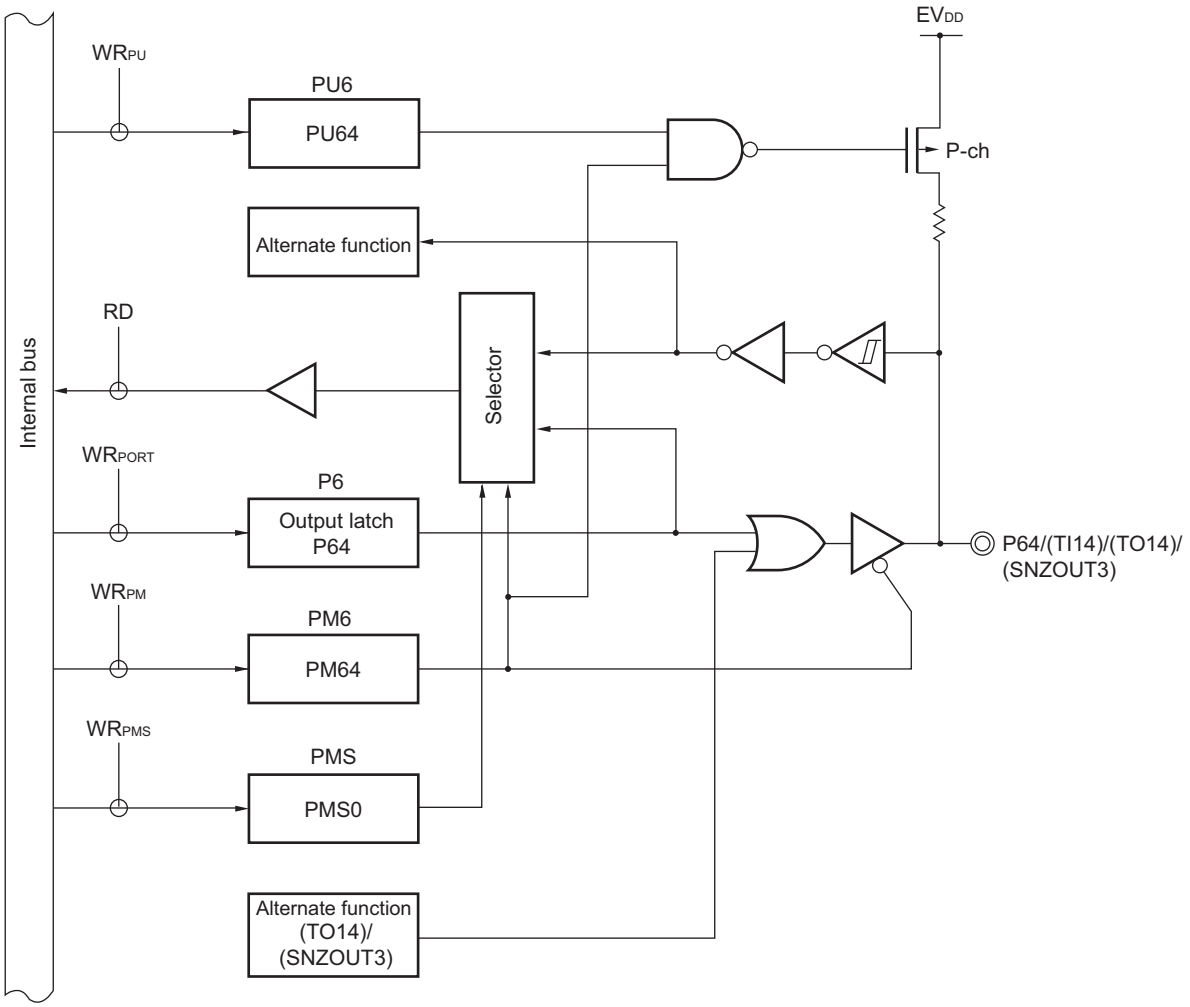

P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
PIM6: Port input mode register 6
POM6: Port output mode register 6
PMS: Port mode select register
PITHL6: Port input threshold control register 6
RD: Read signal
WRxx: Write signal

**Figure 4-36. Block Diagram of P63**



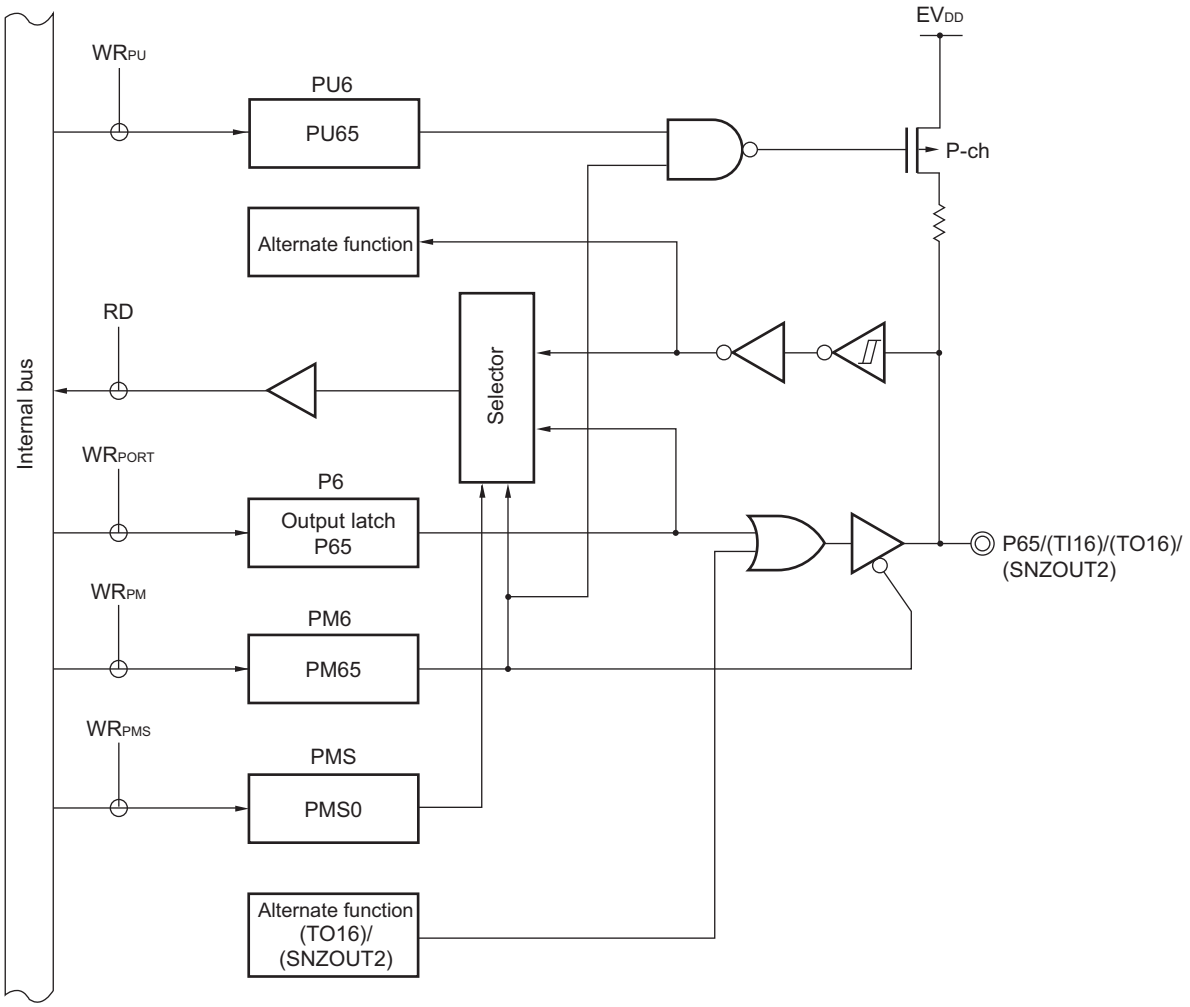

P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
PIM6: Port input mode register 6
POM6: Port output mode register 6
PMS: Port mode select register
PITHL6: Port input threshold control register 6
RD: Read signal
WRxx: Write signal

**Figure 4-37. Block Diagram of P64**



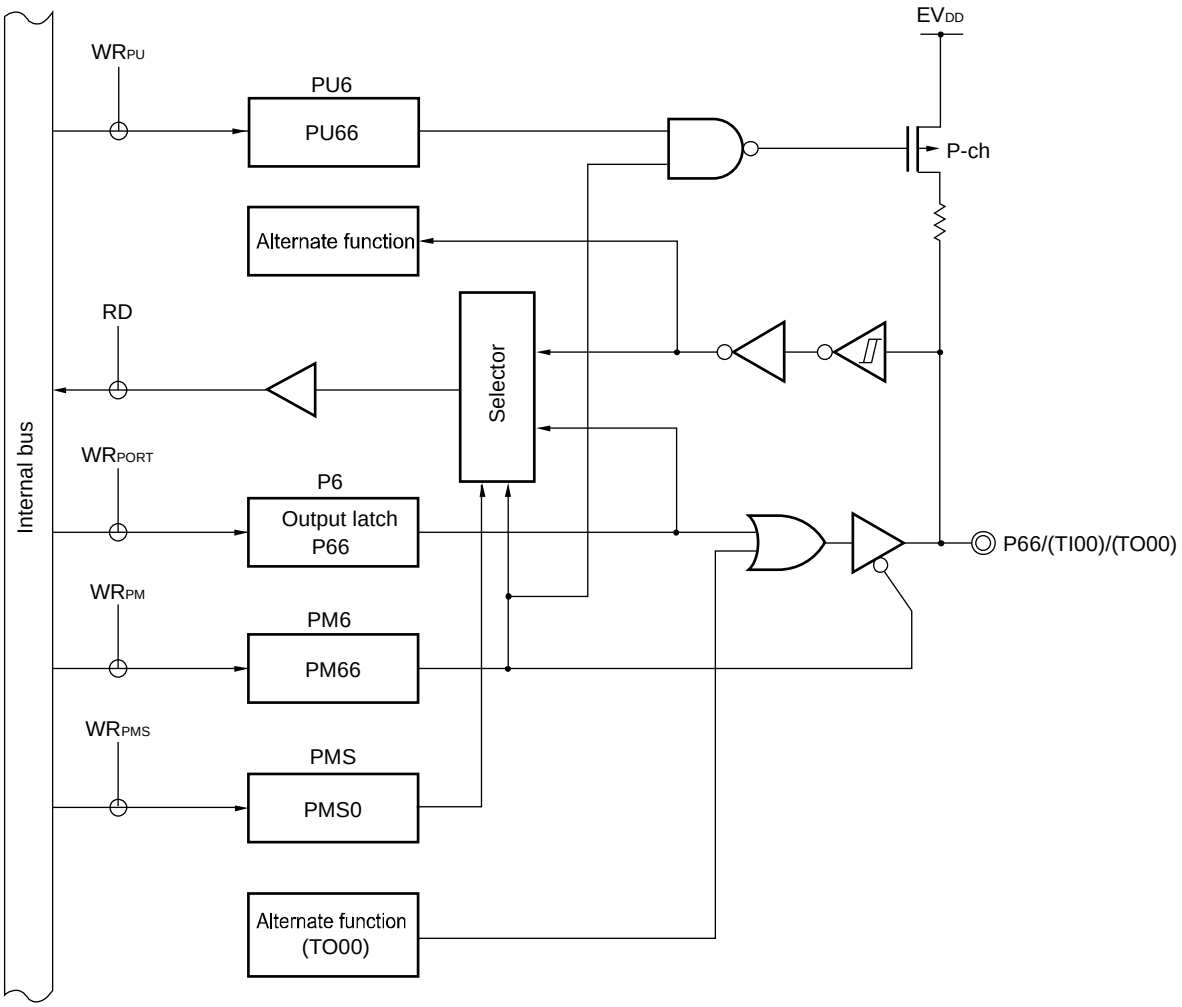

P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-38. Block Diagram of P65**



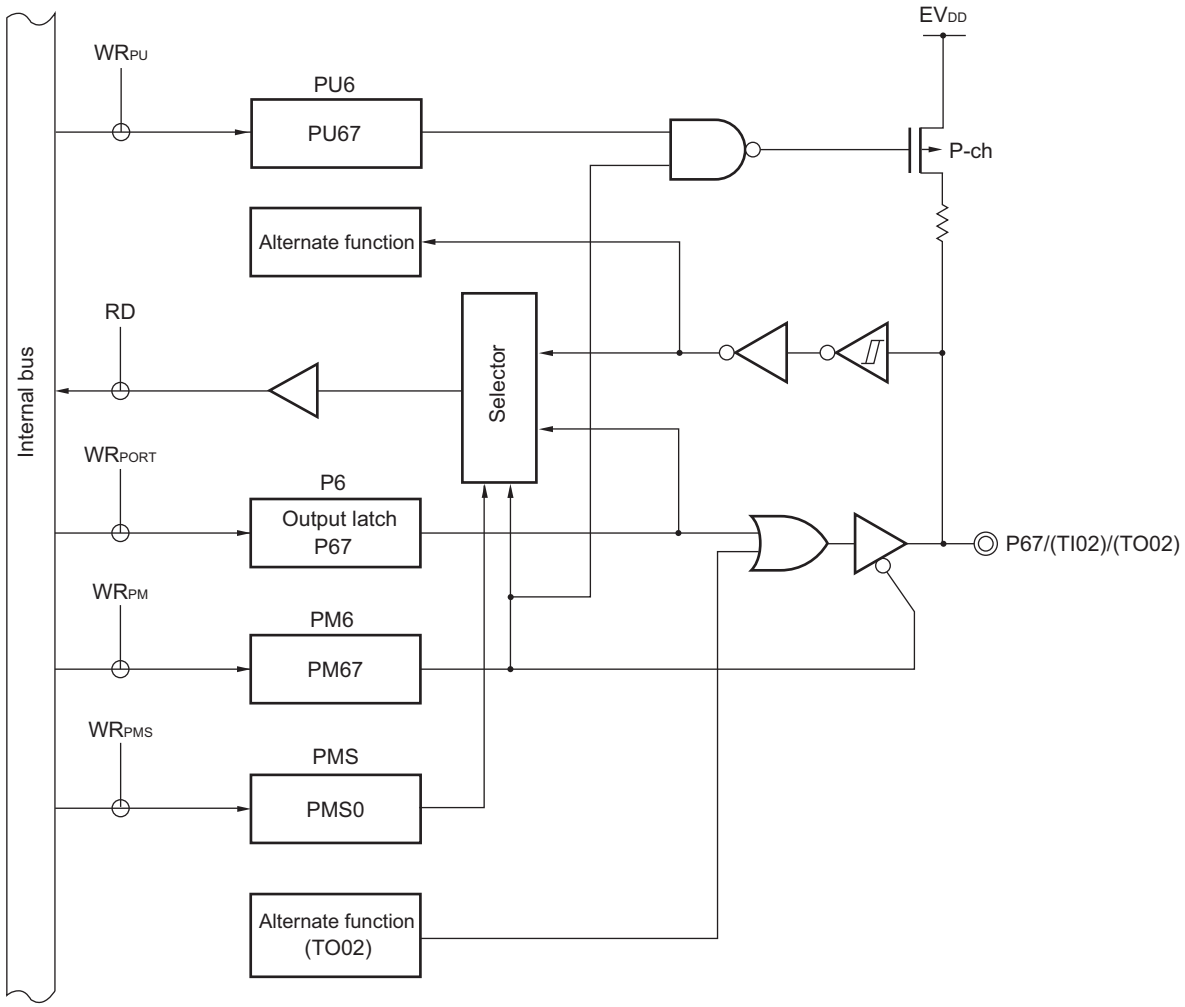

P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-39. Block Diagram of P66**




P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-40. Block Diagram of P67**




P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

### 4.2.7 Port 7

Port 7 is an I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

Input to the P70, P71, and P73 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 7 (PIM7). For the P70, P71, P73, and P75 to P77 pin input, the threshold of the input buffer can be specified in 1-bit units using the port input threshold control register 7 (PITHL7). Output from the P70 to P72 pins can be specified as N-ch open-drain output ($EV_{DD}$ tolerance) in 1-bit units using port output mode register 7 (POM7). To use P70 to P74 as input pins, set them in the digital mode or analog mode in 1-bit units by using the port mode control register 7 (PMC7).

This port can also be used for A/D converter analog input, key interrupt input, data I/O for serial interfaces, clock I/O, slave select input, timer I/O, external interrupt request input, SNOOZE status output, and serial data I/O for CAN.

<R> To use P70/ANI26 to P74/ANI30 as digital input pins, set them in the digital I/O mode by using the port mode control register 7 (PMC7) and in the input mode by using the PM7 register. Use these pins starting from the upper bit.

To use P70/ANI26 to P74/ANI30 as digital output pins, set them in the digital I/O mode by using the port mode control register 7 (PMC7) and in the output mode by using the PM7 register.

To use P70/ANI26 to P74/ANI30 as analog input pins, set them in the analog input mode by using the port mode control register 7 (PMC7) and in the input mode by using the PM7 register. Use these pins starting from the lower bit.

Reset signal generation sets P70 to P74 to analog input mode and P75 to P77 to input mode.

**Table 4-11. Settings of Registers When Using Port 7 (1/2)**

| Pin Name | | PM7x | PIM7x | POM7x | PMC7x | PITHL7x | Alternate Function Setting Note 6 | Remark |
|---|---|---|---|---|---|---|---|---|
| Name | I/O | | | | | | | |
| P70 | Input | 1 | 0 | × | 0 | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | | 1 | | CMOS input (Schmitt3 input) |
| | | | 1 | × | 0 | × | × | TTL input |
| | Output | 0 | × | 0 | 0 | × | SDA11 output = 1 Note 1<br>TO15 output = 0 Note 2<br>SNZOUT4 output = 0 Note 3 | CMOS output |
| | | | × | 1 | 0 | × | | N-ch O.D output |
| P71 | Input | 1 | 0 | × | 0 | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | | 1 | | CMOS input (Schmitt3 input) |
| | | | 1 | × | 0 | × | × | TTL input |
| | Output | 0 | × | 0 | 0 | × | $\overline{\text{SCK11}}$ output = 1 Note 4<br>TO17 output = 0 Note 2<br>SCL11 output = 1 Note 4<br>SNZOUT5 output = 0 Note 3 | CMOS output |
| | | | × | 1 | 0 | × | | N-ch O.D output |
| P72 | Input | 1 | – | × | 0 | – | × | |
| | Output | 0 | – | 0 | 0 | – | SO11 output = 1 Note 1<br>SNZOUT6 output = 0 Note 3<br>(CTXD0 output = 1 Note 5) | CMOS output |
| | | | – | 1 | 0 | – | | N-ch O.D output |
| P73 | Input | 1 | 0 | – | 0 | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | | 1 | × | CMOS input (Schmitt3 input) |
| | | | 1 | – | 0 | × | × | TTL input |
| | Output | 0 | × | – | 0 | × | SNZOUT7 output = 0 Note 3 | CMOS output |
| P74 | Input | 1 | – | – | 0 | – | × | |
| | Output | 0 | – | – | 0 | – | (SO10 output = 1) Note 1<br>(TXD1 output = 0) Note 1 | |

(**Notes** and **Remark** are listed on the next page.)

**Table 4-11. Settings of Registers When Using Port 7 (2/2)**

| Pin Name | | PM7x | PIM7x | POM7x | PMC7x | PITHL7x | Alternate Function Setting | Remark |
|---|---|---|---|---|---|---|---|---|
| Name | I/O | | | | | | | |
| P75 | Input | 1 | – | – | – | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | | 1 | × | CMOS input (Schmitt3 input) |
| | Output | 0 | – | – | – | × | × | |
| P76 | Input | 1 | – | – | – | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | | 1 | × | CMOS input (Schmitt3 input) |
| | Output | 0 | – | – | – | × | ($\overline{\text{SCK10}}$ output = 1 Note 4) | |
| P77 | Input | 1 | – | – | – | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | | 1 | × | CMOS input (Schmitt3 input) |
| | Output | 0 | – | – | – | × | × | |

**Notes 1.** When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn bit of the serial output register m (SOm), the SOEmn bit of the serial output enable register m (SOEm), and the SEmn bit of the serial channel enable status register m (SEm) corresponding to the target unit and channel must have the same setting as its initial value (m = 0, 1, n = 0, 1).

**2.** When a pin sharing a timer output function of the timer array unit is to be used as a general-purpose port pin, the TOmn bit of the timer output register m (TOm) and the TOEmn bit of the timer output enable register m (TOEm) corresponding to the target unit and channel must have the same setting as in the initial state (m = 0, 1, n = 0 to 7).

**3.** When a pin sharing the SNOOZE status output function is to be used as a general-purpose port pin, the OUTEN0 to OUTEN7 bits of the SNOOZE status output control registers 0, 1, 2, 3 (PSNZCNT0, 1, 2, 3) must have the same setting as its initial value.

**4.** When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the CKOmn bit of the serial output register m (SOm), the SOEmn bit of the serial output enable register m (SOEm), and the SEmn bit of the serial channel enable status register m (SEm) corresponding to the target unit and channel must have the same setting as its initial value (m = 0, 1, n = 0, 1).

**5.** When a pin sharing the serial data output function of the CAN is to be used as a general-purpose port pin, operation of the corresponding CAN must be stopped.

**6.** Functions in parentheses can be assigned via settings in the peripheral I/O redirection register 4 (PIOR4).

**Remark** ×: Don't care
PM7x: Port mode register 7
PIM7x: Port input mode register 7
POM7x: Port output mode register 7
PMC7x: Port mode control register 7
PITHL7x: Port input threshold control register 7

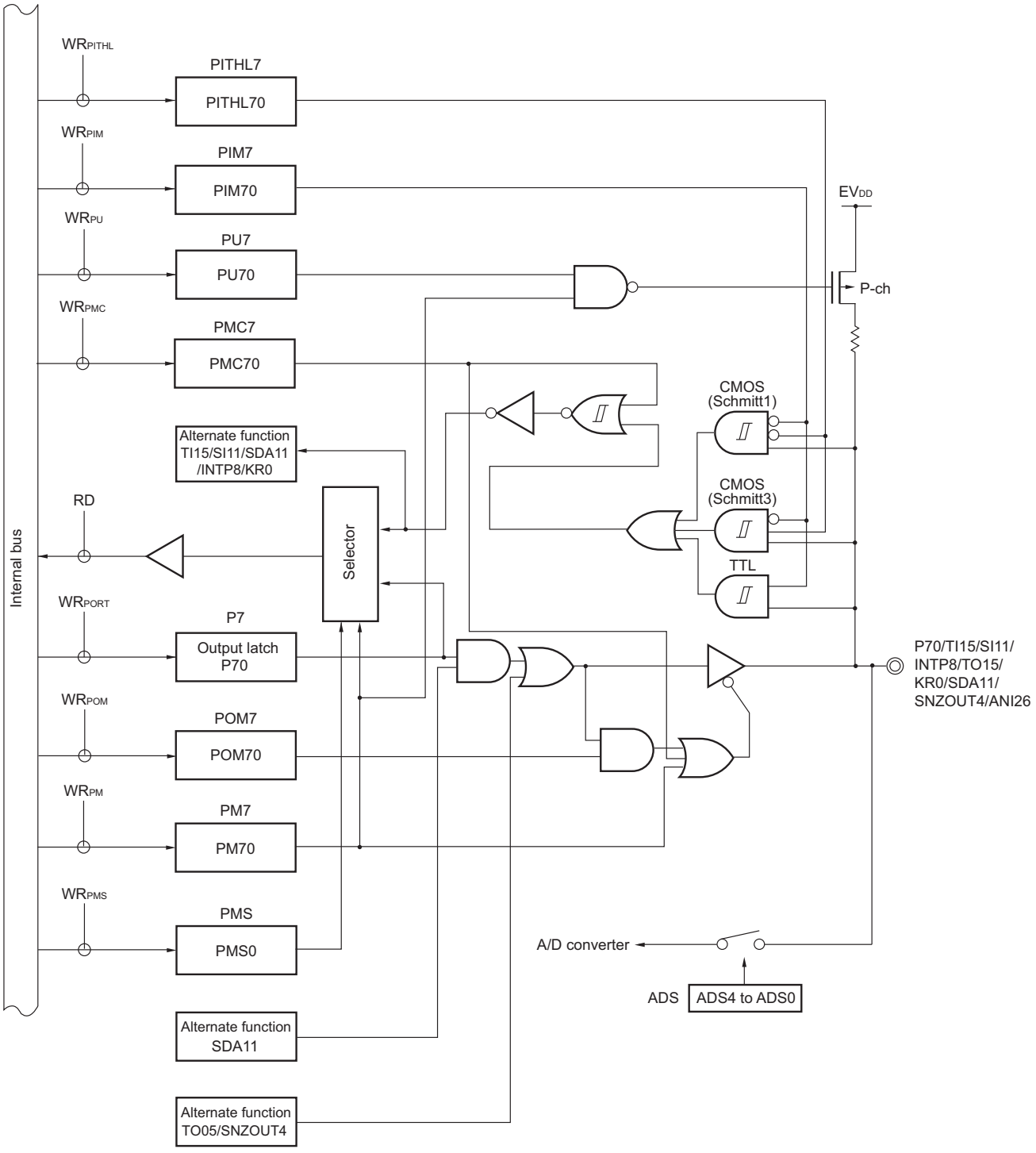
**Table 4-12. Setting Functions of P70/ANI26 to P74/ANI30 Pins**

| ADPC Register | PM7 Register | ADS Register | P70/ANI26 to P74/ANI30 Pins |
|---|---|---|---|
| Digital I/O selection | Input mode | — | Digital input |
| | Output mode | — | Digital output |
| Analog input selection | Input mode | Selects ANI. | Analog input (to be converted) |
| | | Does not select ANI. | Analog input (not to be converted) |
| | Output mode | Selects ANI. | Setting prohibited |
| | | Does not select ANI. | |

Reset signal generation sets P70/ANI26 to P74/ANI30 to analog input mode.

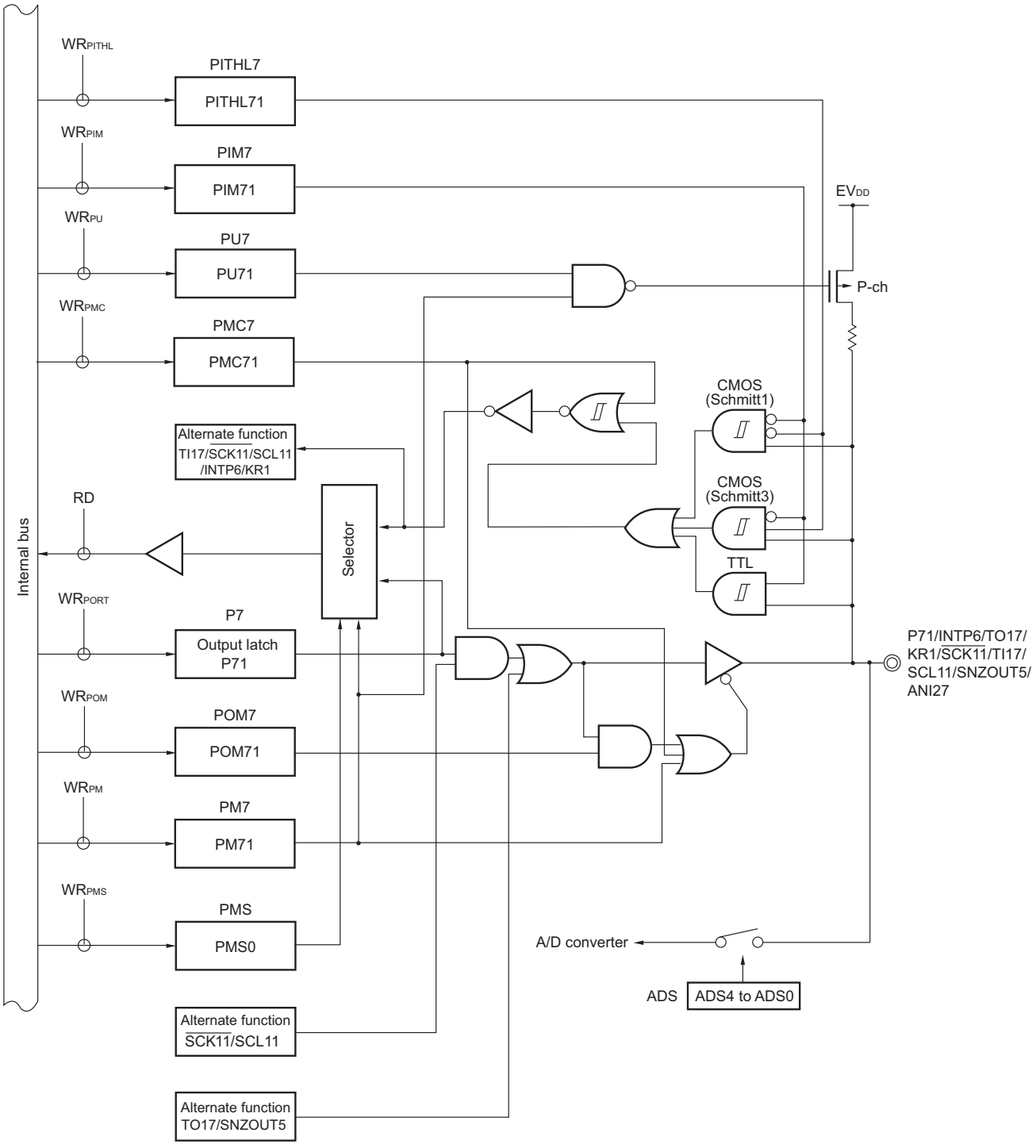
Figures 4-41 to 4-48 show block diagrams of port 7 for 100-pin products.

**Figure 4-41. Block Diagram of P70**



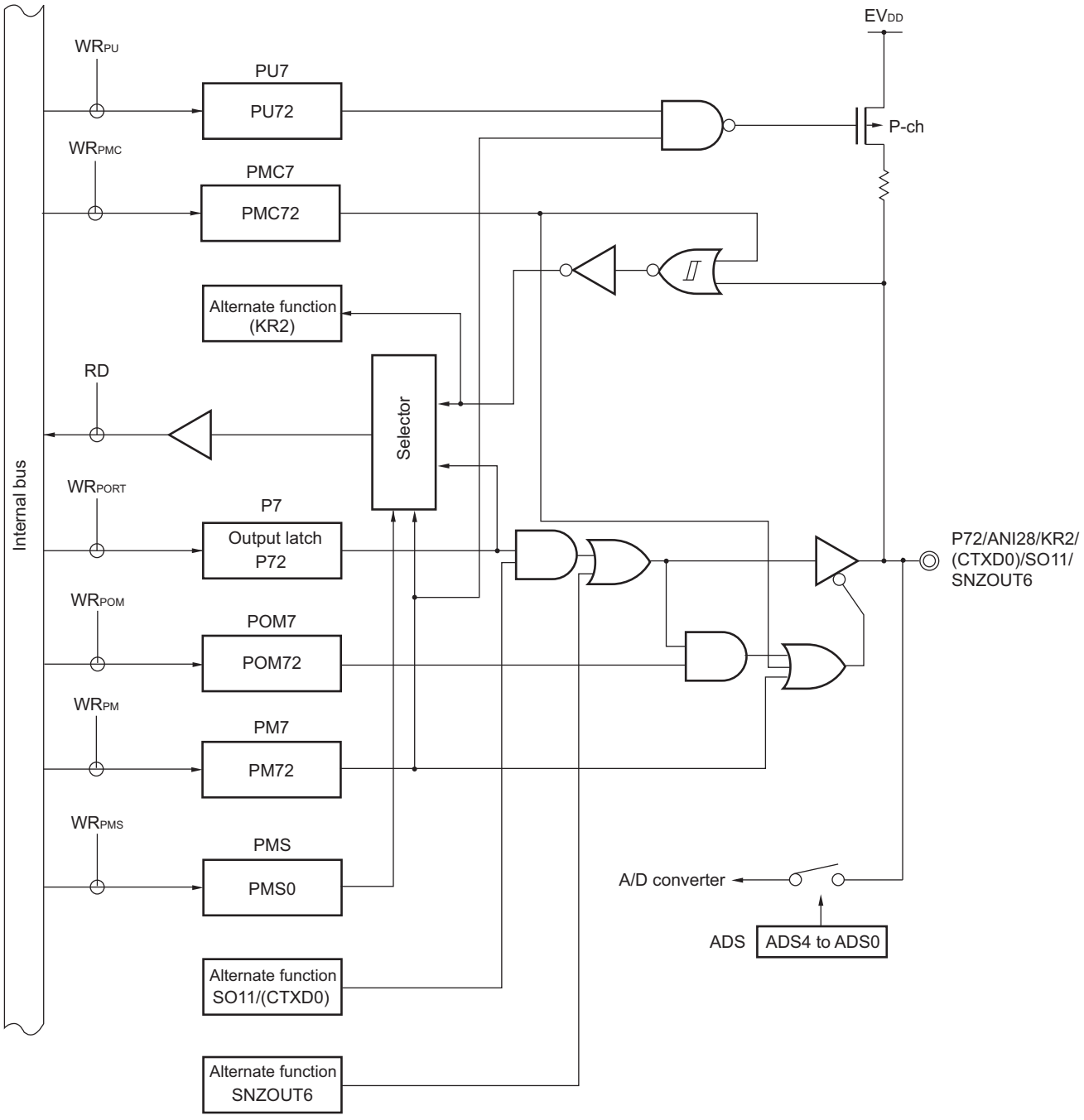
| | |
|---|---|
| P7: | Port register 7 |
| PU7: | Pull-up resistor option register 7 |
| PM7: | Port mode register 7 |
| PIM7: | Port input mode register 7 |
| POM7: | Port output mode register 7 |
| PMC7: | Port mode control register 7 |
| PMS: | Port mode select register |
| PITHL7: | Port input threshold control register 7 |
| ADS: | Analog input channel specification register |
| RD: | Read signal |
| WRxx: | Write signal |

**Figure 4-42. Block Diagram of P71**



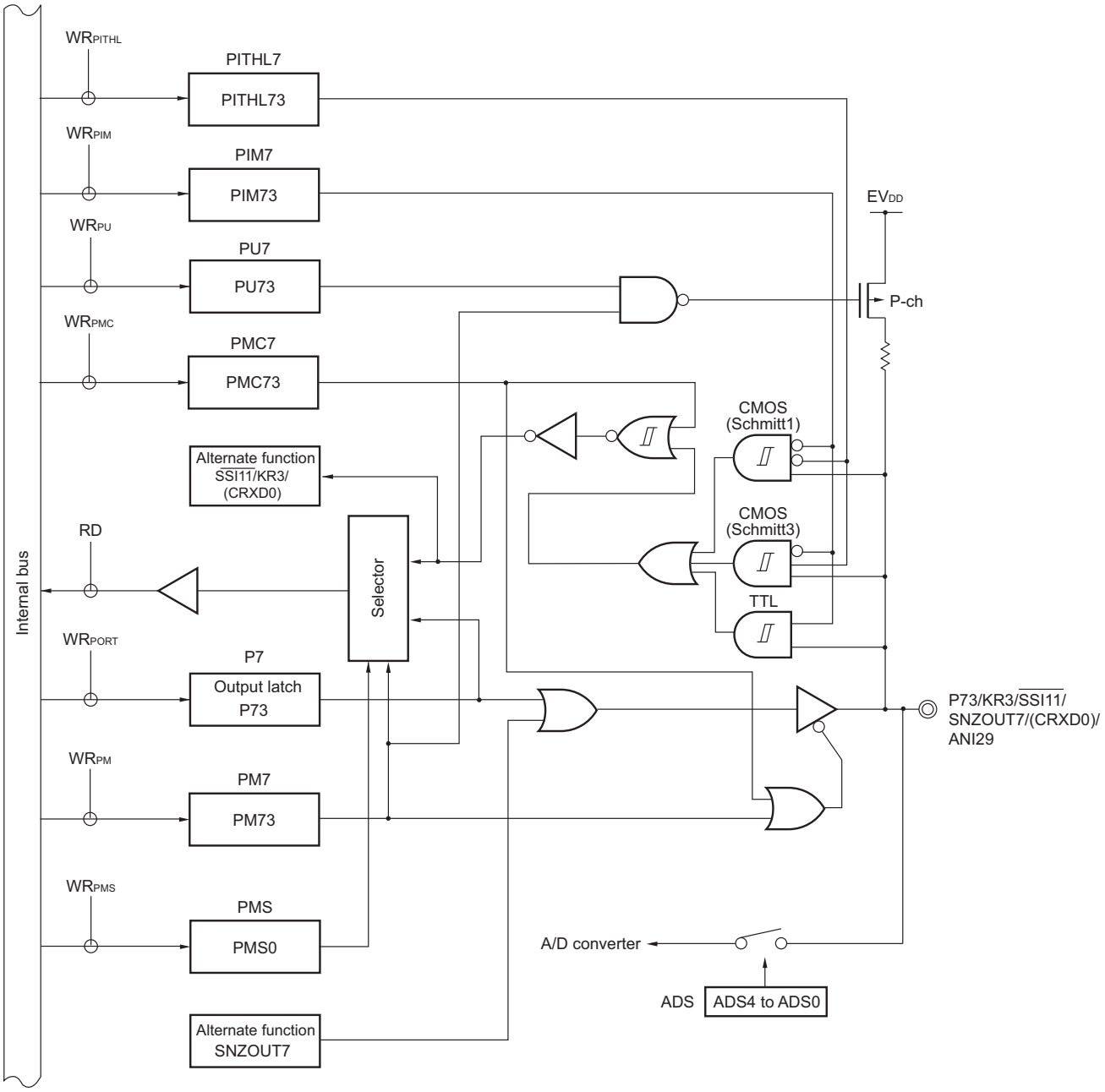
P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
PIM7: Port input mode register 7
POM7: Port output mode register 7
PMC7: Port mode control register 7
PMS: Port mode select register
PITHL7: Port input threshold control register 7
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-43. Block Diagram of P72**



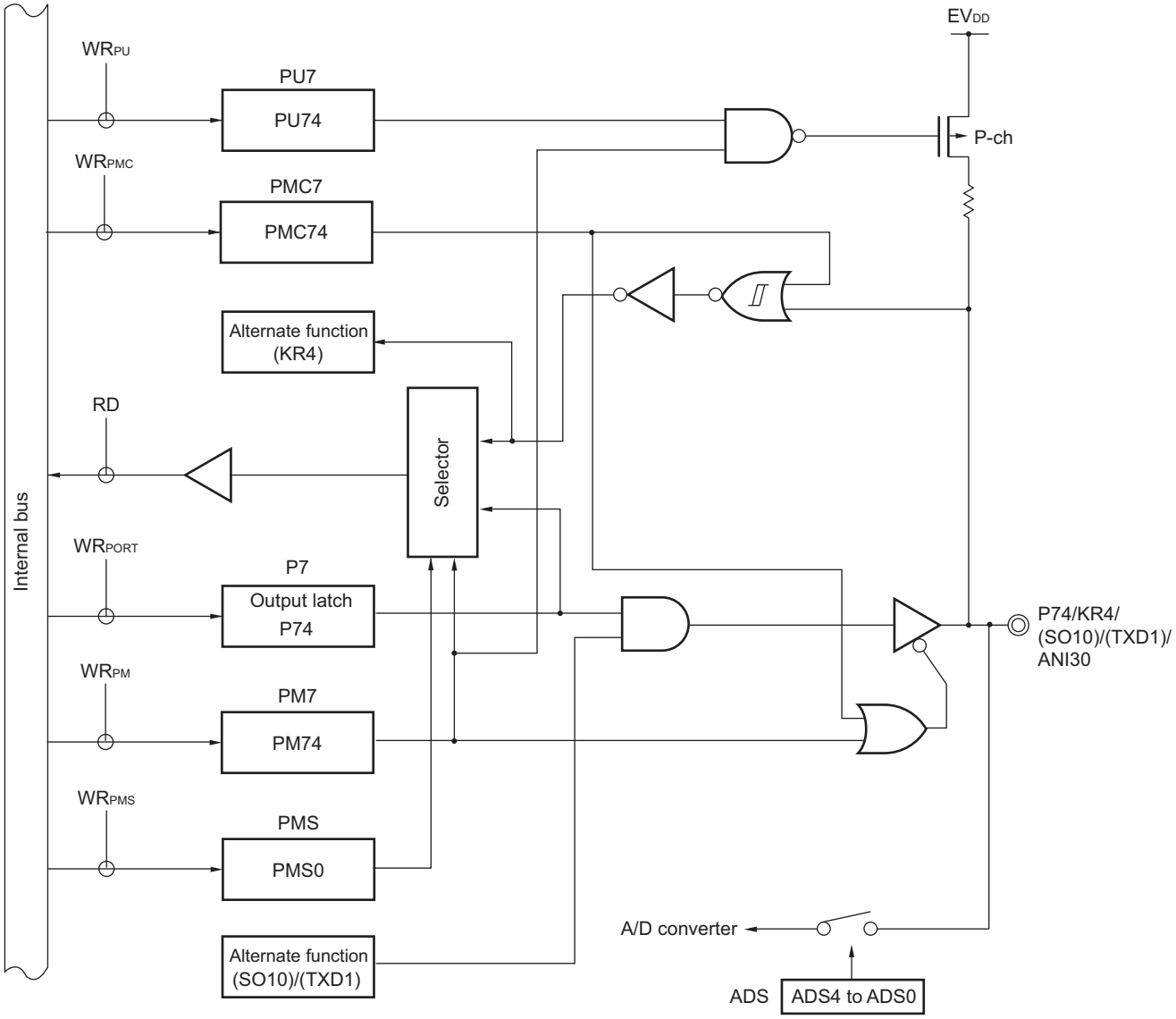

P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
POM7: Port output mode register 7
PMC7: Port mode control register 7
PMS: Port mode select register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-44. Block Diagram of P73**



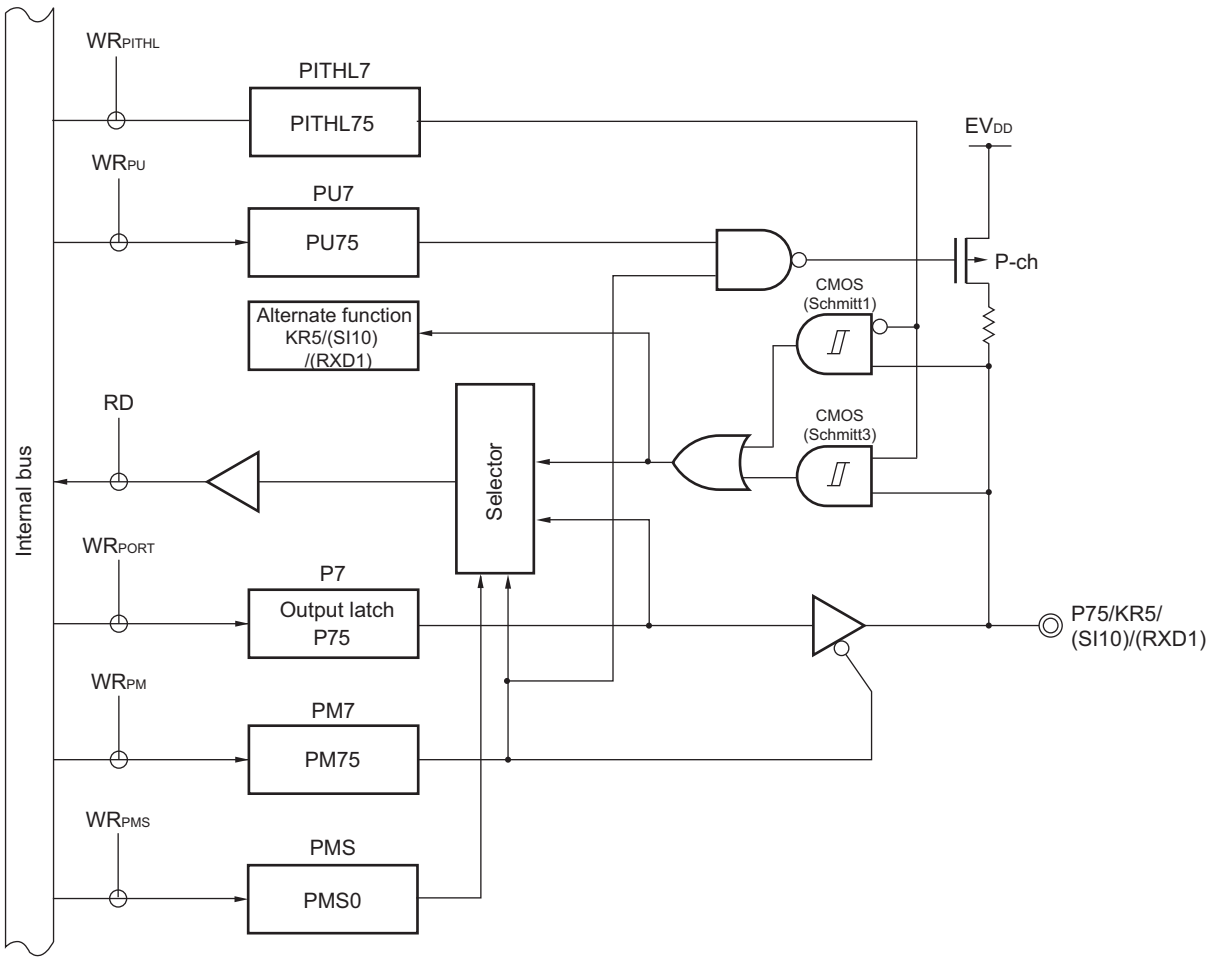

P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
PIM7: Port input mode register 7
PMC7: Port mode control register 7
PMS: Port mode select register
PITHL7: Port input threshold control register 7
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-45. Block Diagram of P74**



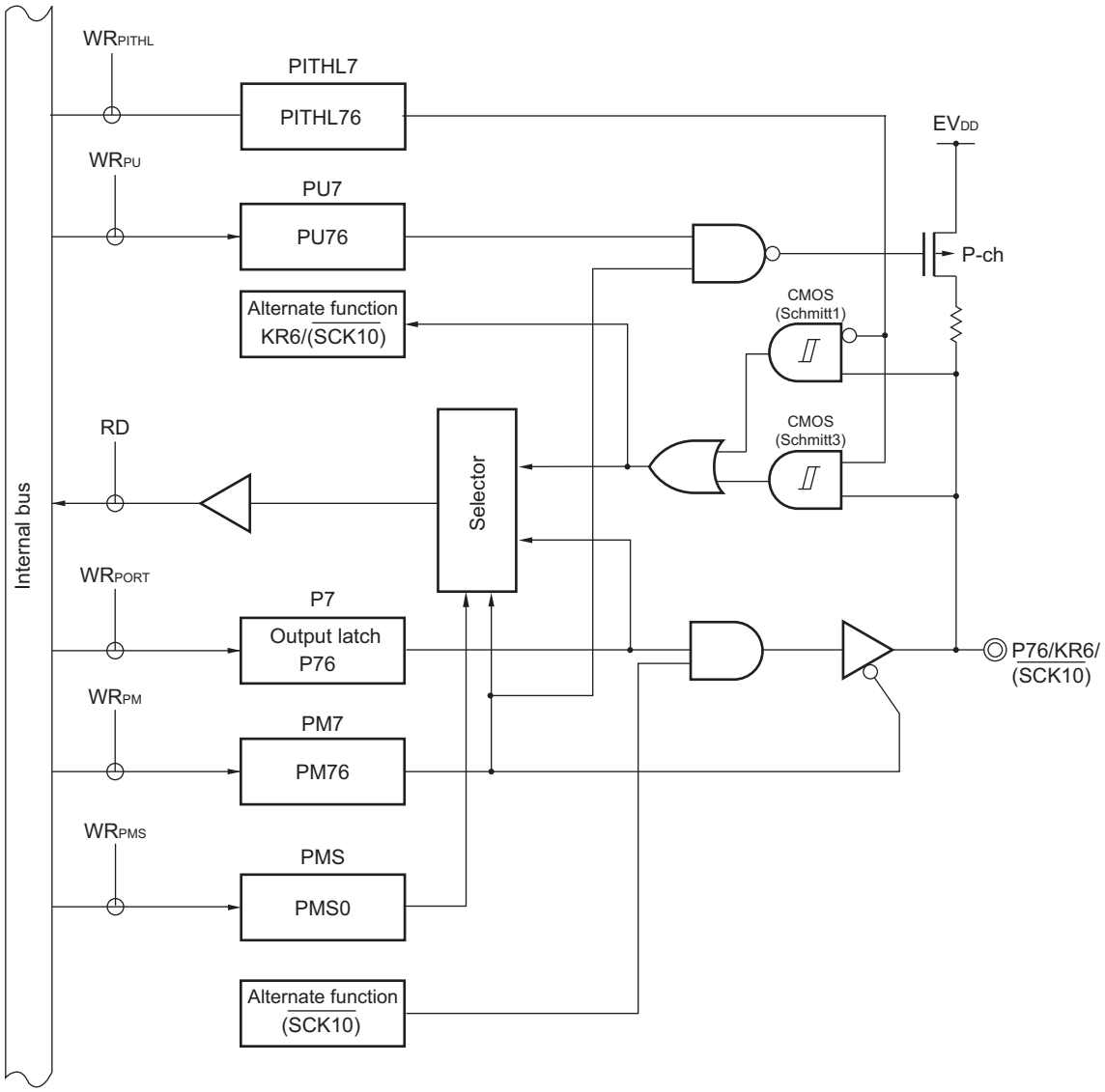

P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
PMC7: Port mode control register 7
PMS: Port mode select register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-46. Block Diagram of P75**



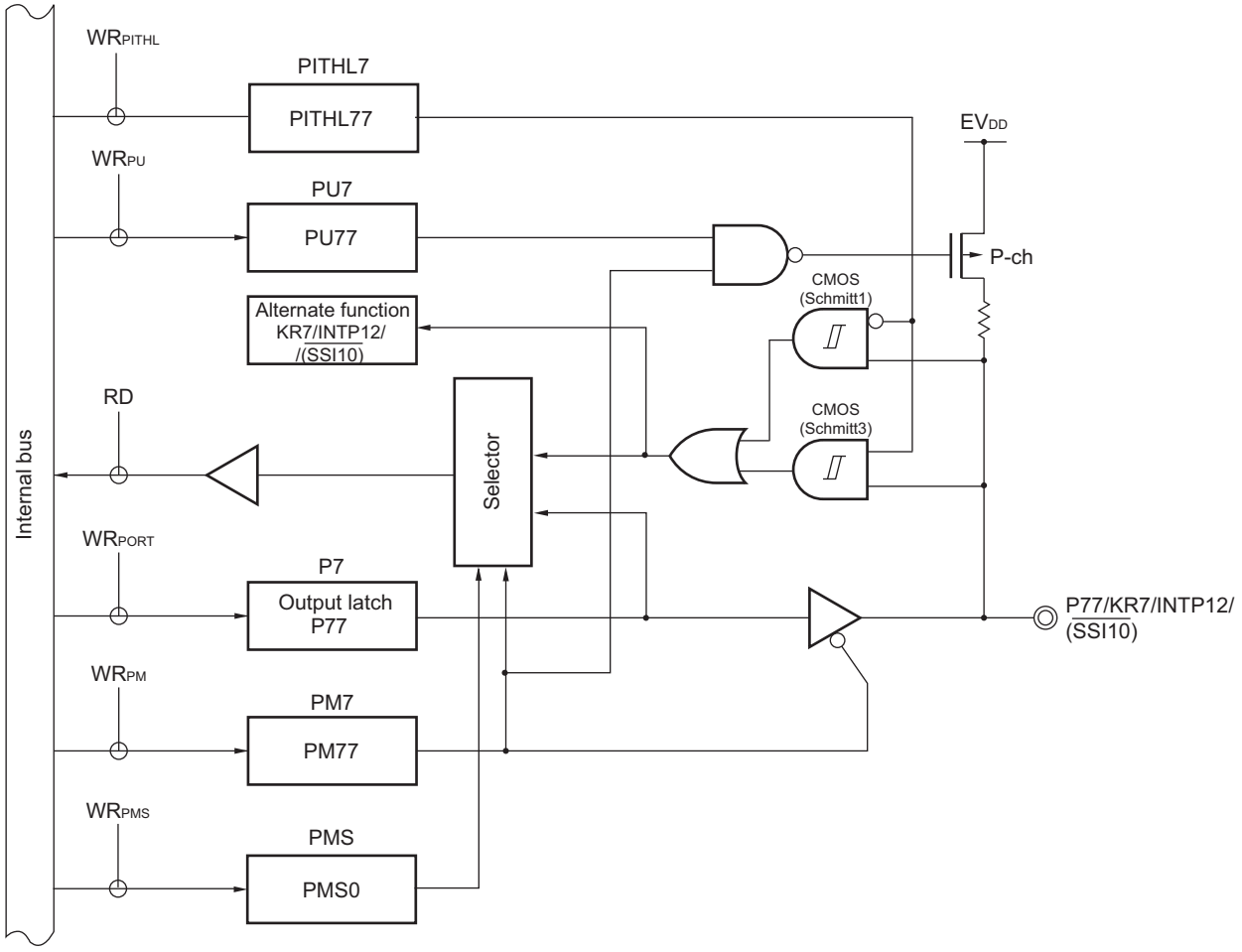

P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
PMS: Port mode select register
PITHL7: Port input threshold control register 7
RD: Read signal
WRxx: Write signal

**Figure 4-47. Block Diagram of P76**




P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
PMS: Port mode select register
PITHL7: Port input threshold control register 7
RD: Read signal
WRxx: Write signal

**Figure 4-48. Block Diagram of P77**




P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
PMS: Port mode select register
PITHL7: Port input threshold control register 7
RD: Read signal
WRxx: Write signal

### 4.2.8 Port 8

Port 8 is an I/O port with an output latch. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8).

This port can also be used for analog input for A/D converter.

P80 to P85 can also be used for D/A converter output, and analog voltage input and reference voltage input for comparator.

To use P80/ANI2 to P87/ANI9 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM8 register. Use these pins starting from the upper bit.

To use P80/ANI2 to P87/ANI9 as digital output pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the output mode by using the PM8 register. Use these pins starting from the upper bit.

To use P80/ANI2 to P87/ANI9 as analog input pins, set them in the analog I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM8 register. Use these pins starting from the lower bit.

**Table 4-13. Settings of Registers When Using Port 8**

| Pin Name | | PM8x | ADPC | Alternate Function Setting | Remark |
|---|---|---|---|---|---|
| Name | I/O | | | | |
| P8n | Input | 1 | 03 to n+3H | – | To use P8n as a port, use these pins from the upper bit. |
| | Output | 0 | 03 to n+3H | | |

**Remarks 1.** PM8x: Port mode register 8
ADPC: A/D port configuration register

**2.** n = 0 to 7

<R>

**Table 4-14. Setting Functions of P80/ANI2/ANO0 Pin**

| ADPC Register | PM8 Register | DAM Register | DAM2 Register | ADS Register | Functions of ANO0/ANI2/P80 Pin |
|---|---|---|---|---|---|
| Digital I/O | Input mode | — | Enables analog output | — | Setting prohibited |
| | | | Disables analog output | | Digital input |
| | Output mode | — | Enables analog output | — | Setting prohibited |
| | | | Disables analog output | | Digital input |
| Analog I/O | Input mode | Enables D/A conversion operation | Enables analog output | Selects ANI | Setting prohibited |
| | | | | Does not selects ANI | Analog output (D/A conversion output) |
| | | | Disables analog output | Selects ANI | Analog input (to be converted) |
| | | | | Does not selects ANI | Analog input (not to be converted) Note |
| | | Stops D/A conversion operation | Enables analog output | Selects ANI | Setting prohibited |
| | | | | Does not selects ANI | Setting prohibited |
| | | | Disables analog output | Selects ANI | Analog input (to be converted) |
| | | | | Does not selects ANI | Analog input (not to be converted) |
| | Output mode | — | — | — | Setting prohibited |

**Note** This is a setting that the D/A converter is used for internal reference voltage of comparator. In this case, set CVRS1, CVRS0 bits of CMPSEL register to 10b (internal reference voltage (DAC output) is selected).

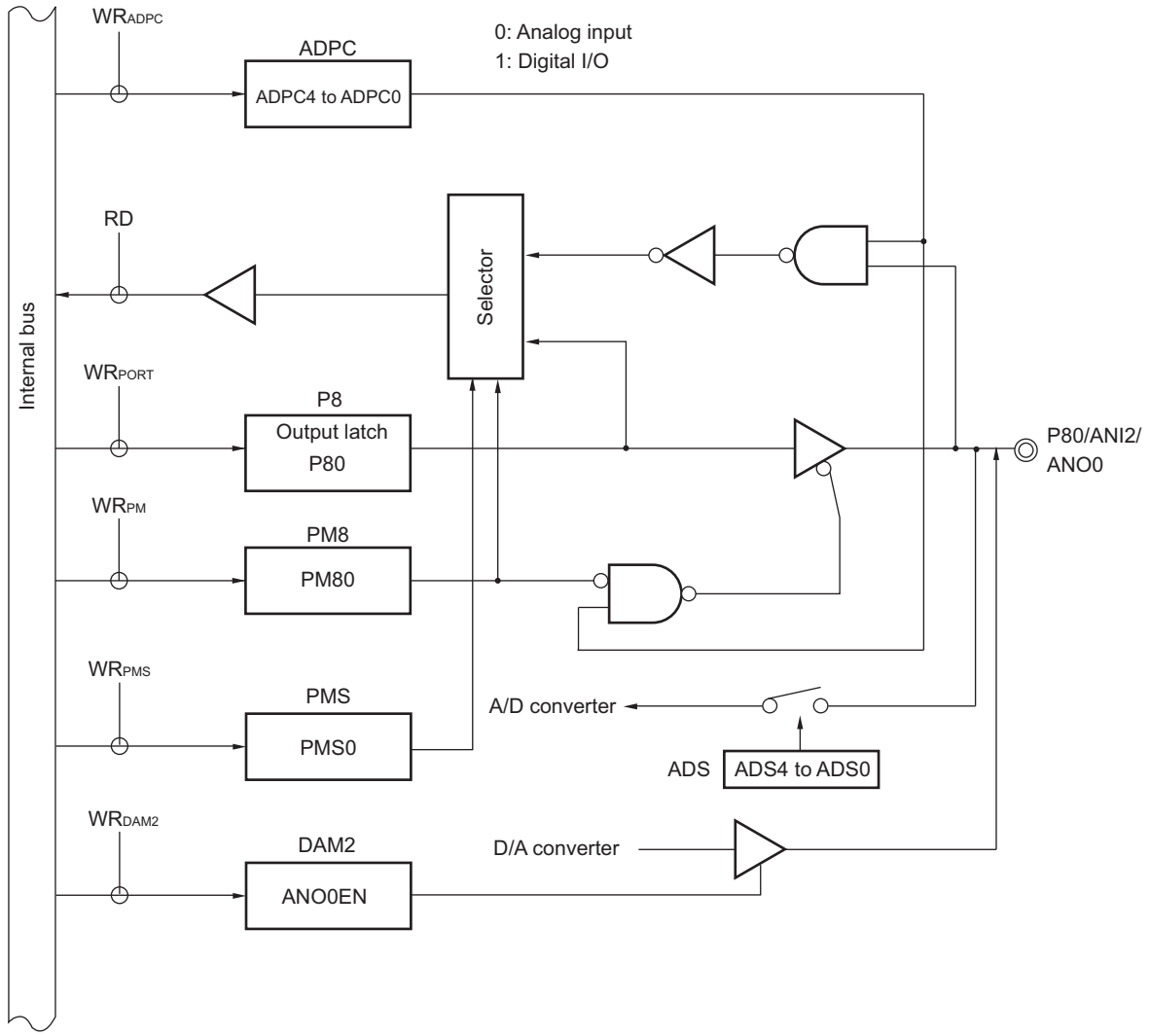
**Table 4-15. Setting Functions of P81/ANI3 to P87/ANI9 Pins**

| ADPC Register | PM8 Register | ADS Register | P81/ANI3 to P87/ANI9 Pins |
|---|---|---|---|
| Digital I/O selection | Input mode | — | Digital input |
| | Output mode | — | Digital output |
| Analog input selection | Input mode | Selects ANI. | Analog input (to be converted) |
| | | Does not select ANI. | Analog input (not to be converted) |
| | Output mode | Selects ANI. | Setting prohibited |
| | | Does not select ANI. | |

Reset signal generation sets P80/ANI3 to P87/ANI9 to analog input mode.

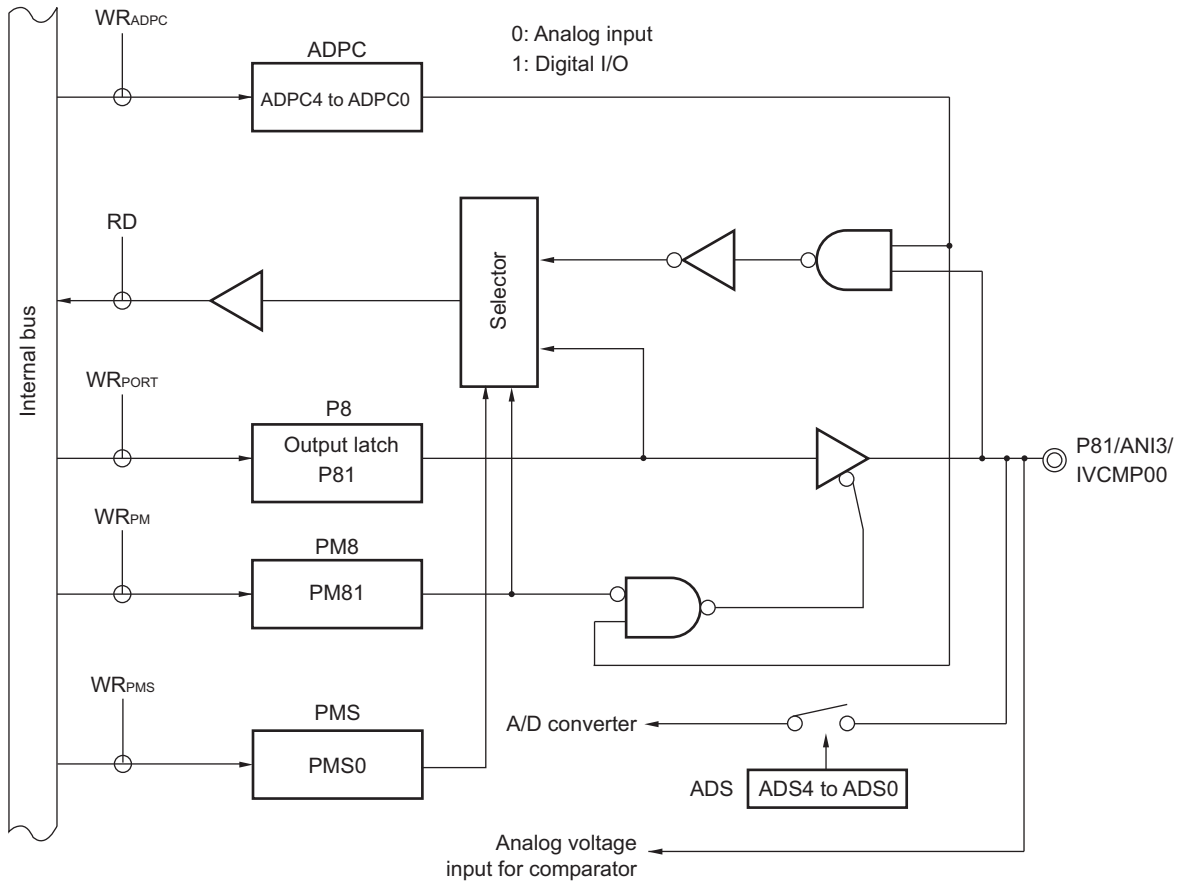
Figures 4-49 to 4-56 show block diagrams of port 8 for 100-pin products.

**Figure 4-49. Block Diagram of P80**



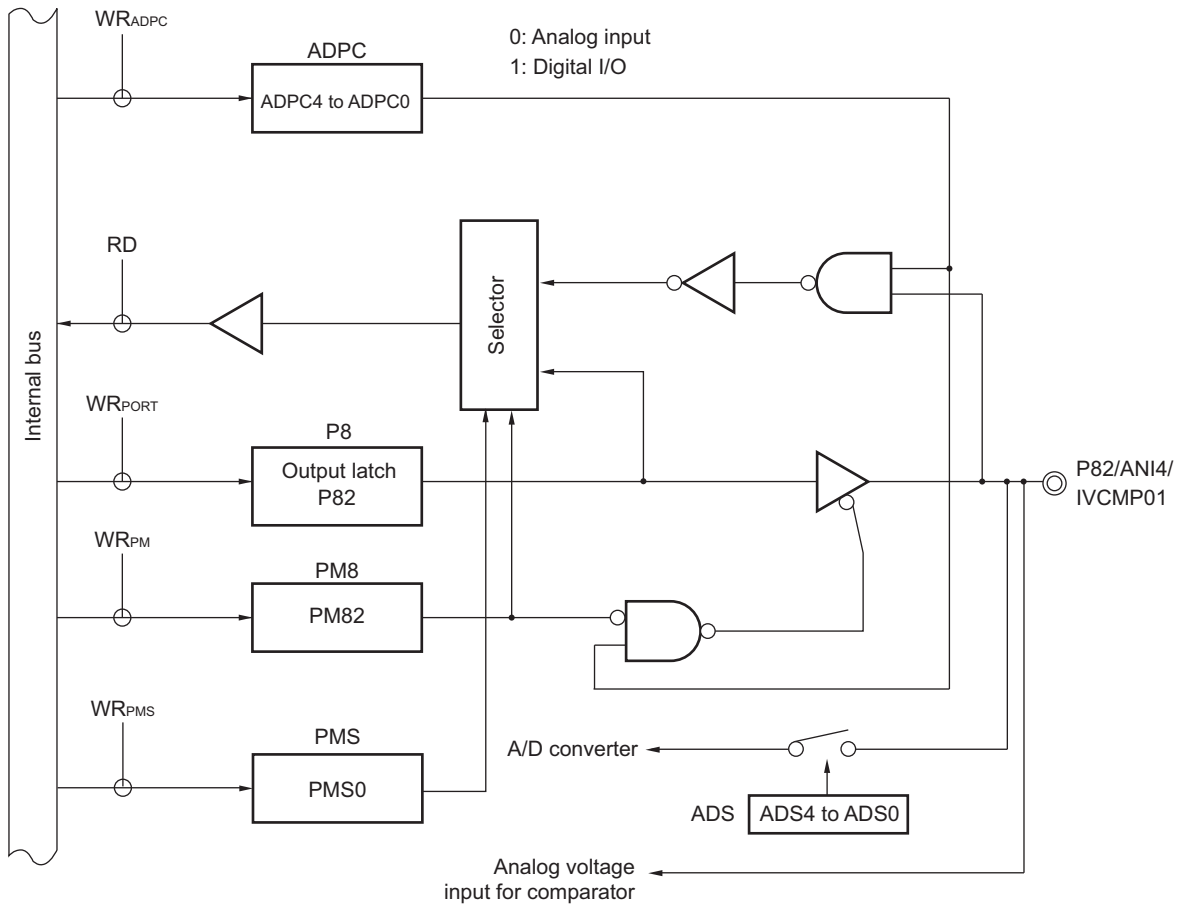

P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
ADPC: A/D port configuration register
DAM2: D/A converter mode register 2
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-50. Block Diagram of P81**



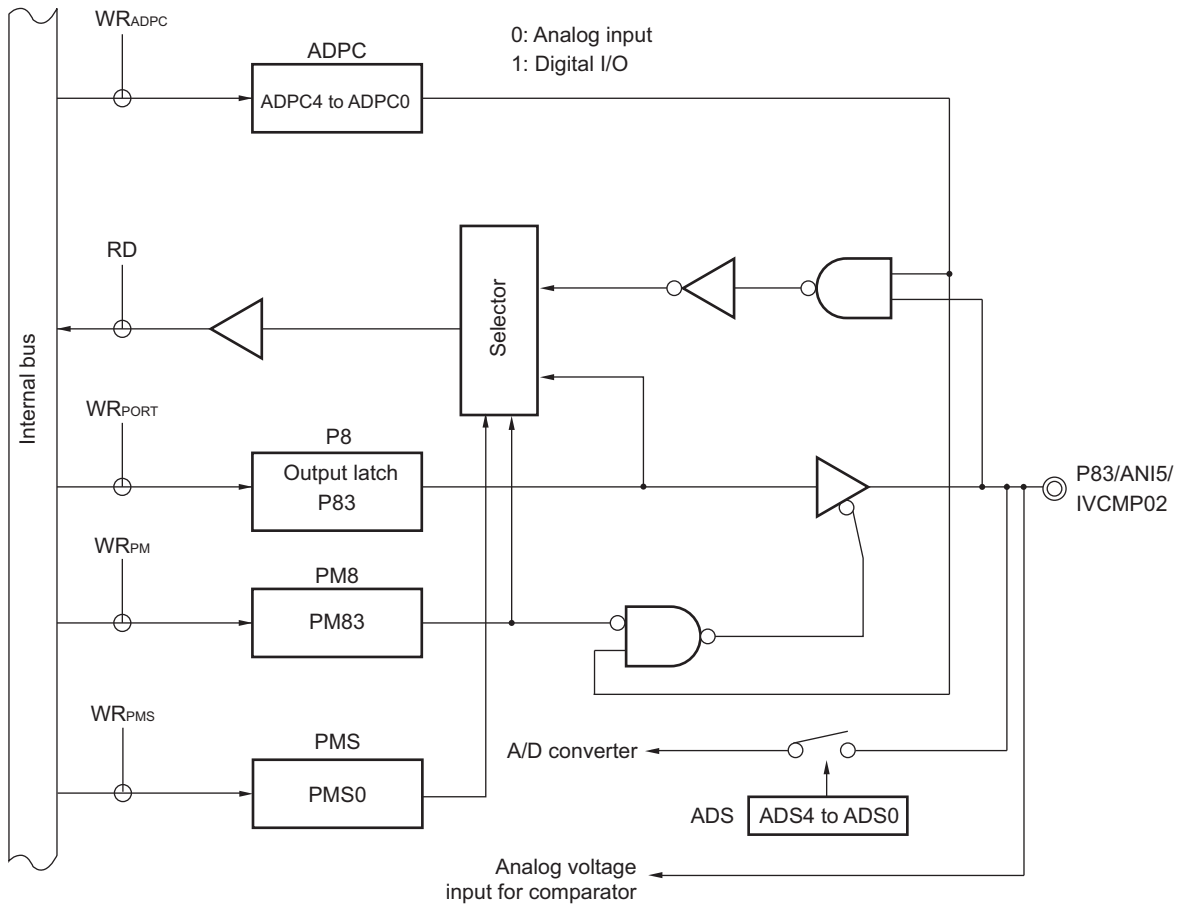

P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
ADPC: A/D port configuration register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-51. Block Diagram of P82**



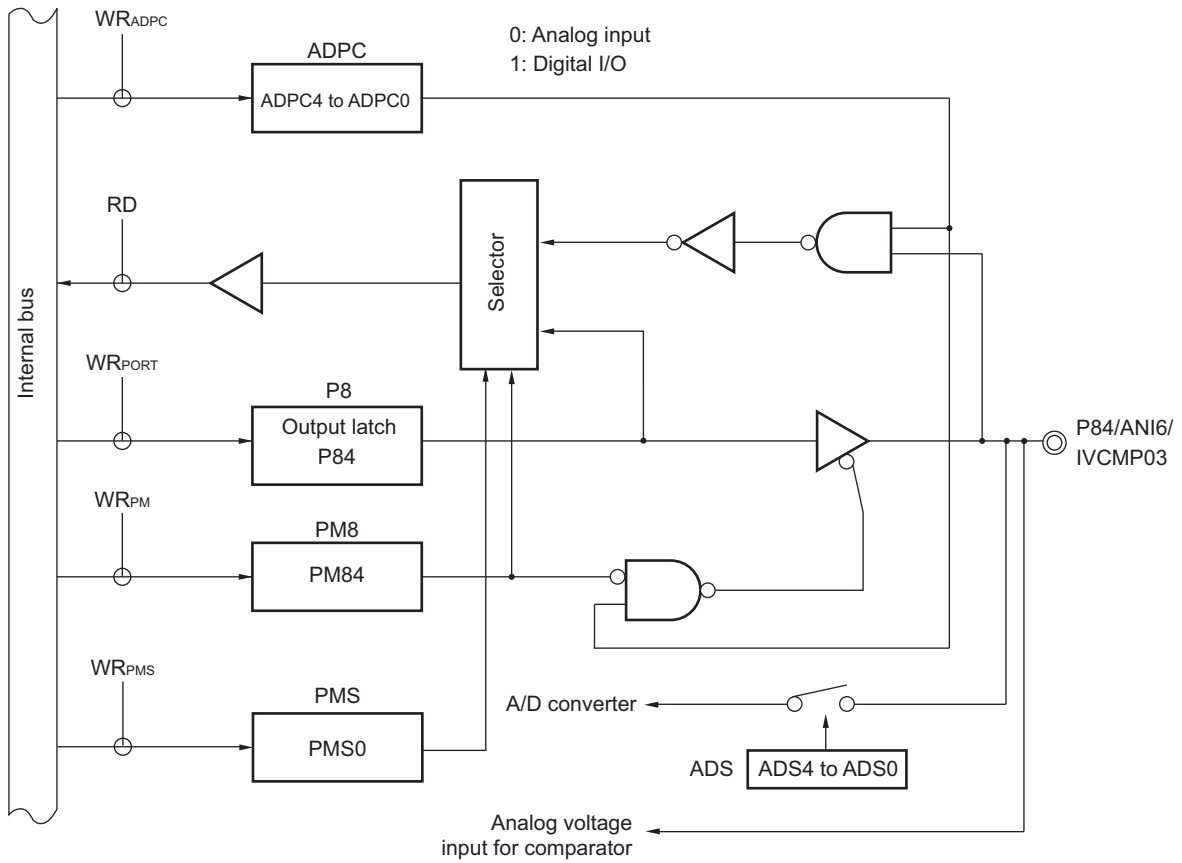

P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
ADPC: A/D port configuration register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-52. Block Diagram of P83**



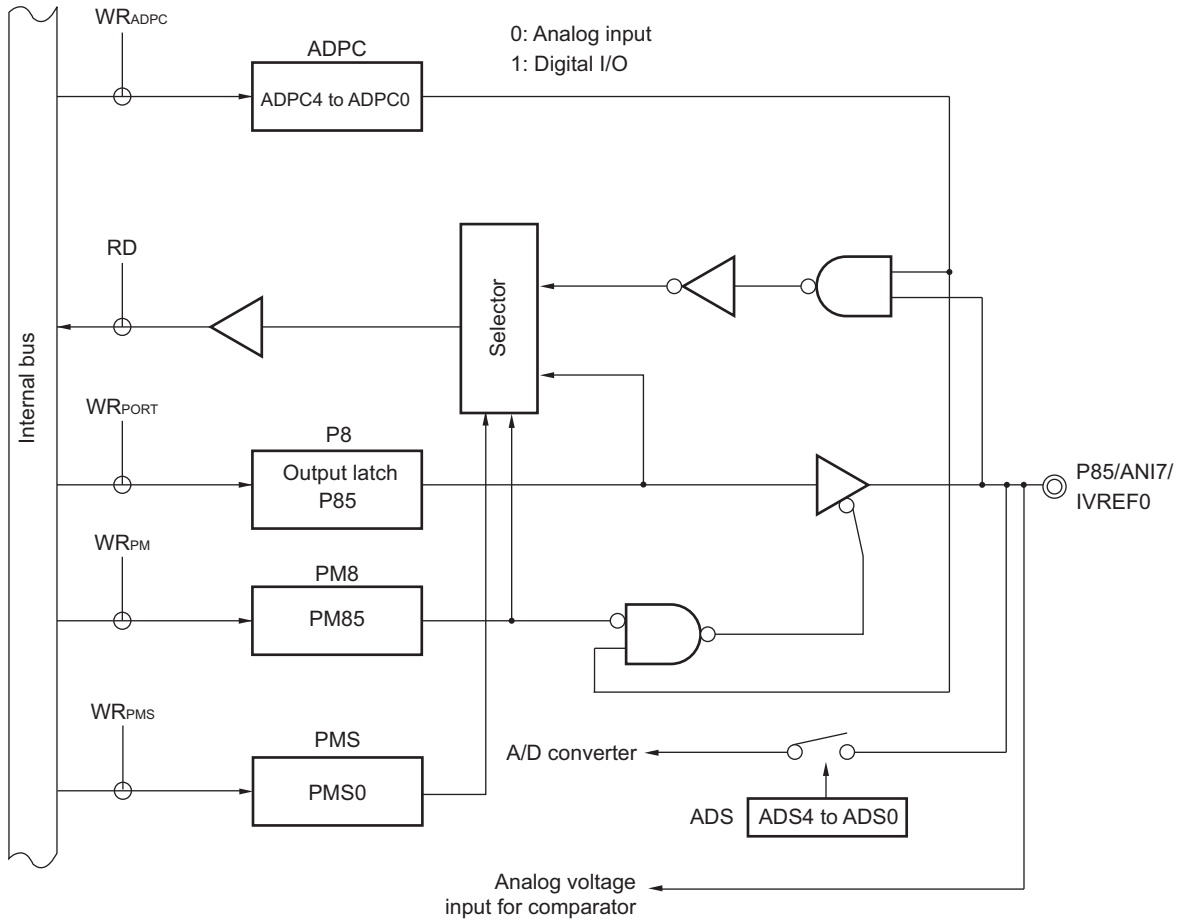

P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
ADPC: A/D port configuration register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-53. Block Diagram of P84**



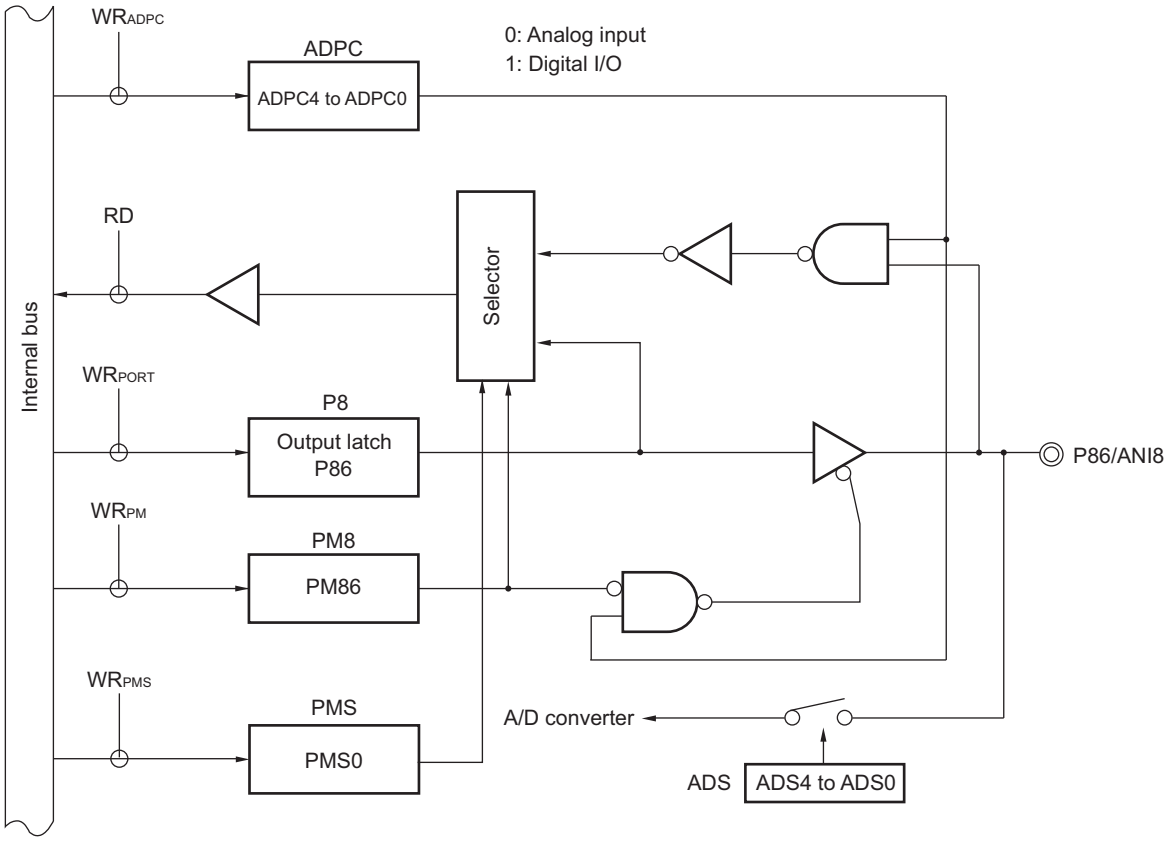

P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
ADPC: A/D port configuration register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-54. Block Diagram of P85**



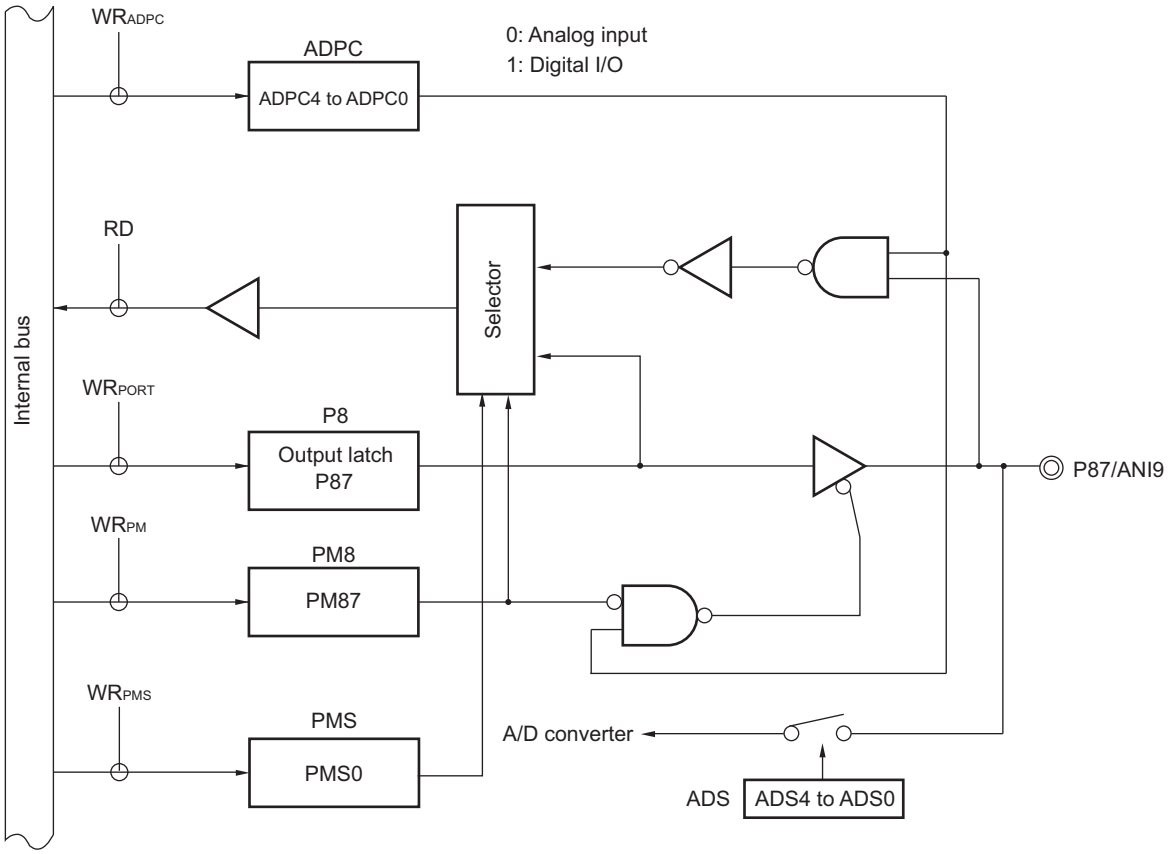

P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
ADPC: A/D port configuration register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-55. Block Diagram of P86**




P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
ADPC: A/D port configuration register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-56. Block Diagram of P87**




P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
ADPC: A/D port configuration register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

### 4.2.9 Port 9

Port 9 is an I/O port with an output latch. Port 9 can be set to the input mode or output mode in 1-bit units using port mode register 9 (PM9).

This port can also be used for A/D converter analog input.

To use P90/ANI10 to P97/ANI17 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM9 register. Use these pins starting from the upper bit.

To use P90/ANI10 to P97/ANI17 as digital output pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the output mode by using the PM9 register. Use these pins starting from the upper bit.

To use P90/ANI10 to P97/ANI17 as analog input pins, set them in the analog input mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM9 register. Use these pins starting from the lower bit.

Reset signal generation sets this port to analog input mode.

**Table 4-16. Settings of Registers When Using Port 9**

| Pin Name | | PM9x | Alternate Function Setting | Remark |
|---|---|---|---|---|
| Name | I/O | | | |
| P90 | Input | 1 | × | |
| | Output | 0 | × | |
| P91 | Input | 1 | × | |
| | Output | 0 | × | |
| P92 | Input | 1 | × | |
| | Output | 0 | × | |
| P93 | Input | 1 | × | |
| | Output | 0 | × | |
| P94 | Input | 1 | × | |
| | Output | 0 | × | |
| P95 | Input | 1 | × | |
| | Output | 0 | × | |
| P96 | Input | 1 | × | |
| | Output | 0 | × | |
| P97 | Input | 1 | × | |
| | Output | 0 | × | |

**Remark** ×: Don't care
PM9x: Port mode register 9

**Table 4-17. Settings of Registers When Using Port 9**

| Pin Name | | PM9x | ADPC | Alternate Function Setting | Remark |
|---|---|---|---|---|---|
| Name | I/O | | | | |
| P9n | Input | 1 | 0C to n+0CH | — | To use P9n as a port, use these pins from the upper bit. |
| | Output | 0 | 0C to n+0CH | | |

**Remarks 1.** PM9x: Port mode register 9
ADPC: A/D port configuration register

**2.** n = 0 to 7

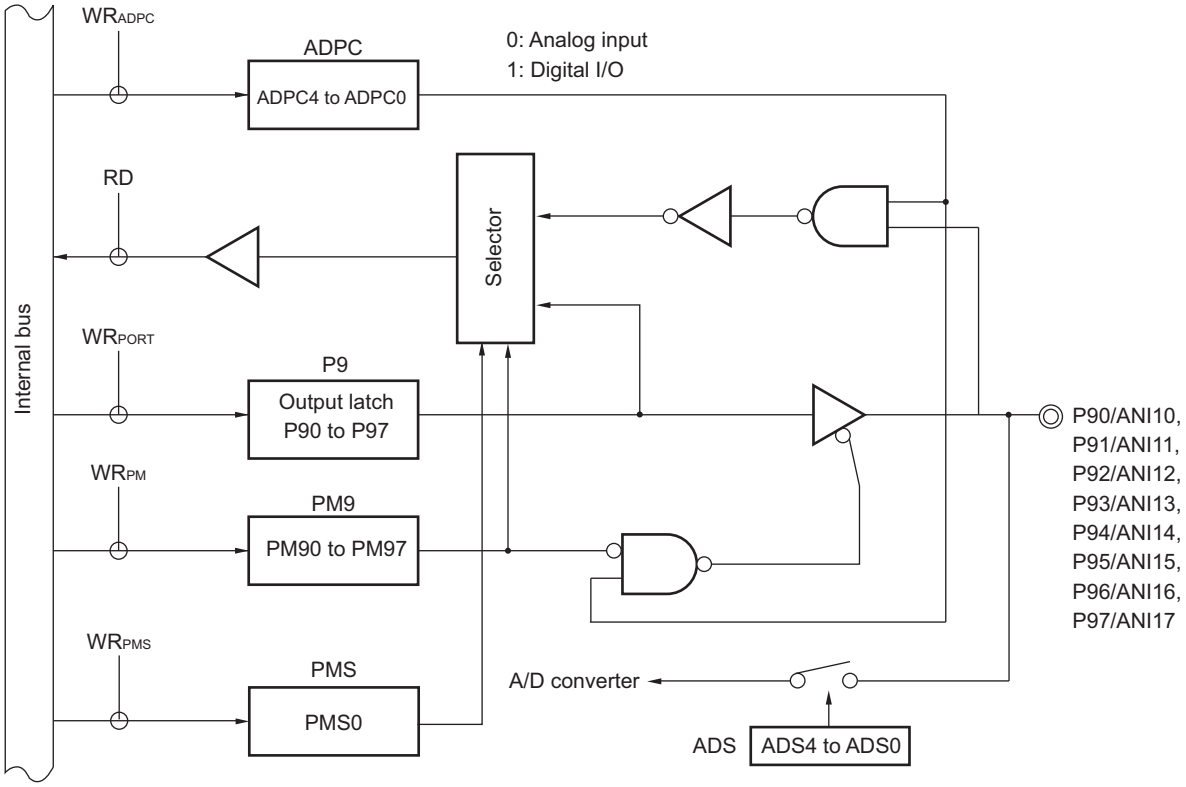
**Table 4-18. Setting Functions of P90/ANI10 to P97/ANI17 Pins**

| ADPC Register | PM9 Register | ADS Register | P90/ANI10 to P97/ANI17 Pins |
|---|---|---|---|
| Digital I/O selection | Input mode | — | Digital input |
| | Output mode | — | Digital output |
| Analog input selection | Input mode | Selects ANI. | Analog input (to be converted) |
| | | Does not select ANI. | Analog input (not to be converted) |
| | Output mode | Selects ANI. | Setting prohibited |
| | | Does not select ANI. | |

Reset signal generation sets P90/ANI10 to P97/ANI17 to analog input mode.

Figure 4-57 shows a block diagram of port 9 for 100-pin products.

**Figure 4-57. Block Diagram of P90 to P97**




P9: Port register 9
PM9: Port mode register 9
PMS: Port mode select register
ADPC: A/D port configuration register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

### 4.2.10 Port 10

Port 10 is an I/O port with an output latch. Port 10 can be set to the input mode or output mode in 1-bit units using port mode register 10 (PM10).

When the P106 and P107 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 10 (PU10). For the P107 pin input, the threshold of the input buffer can be specified in 1-bit units using the port input threshold control register 10 (PITHL10).

This port can also be used for A/D converter analog input and LIN serial data I/O.

To use P100/ANI18 to P105/ANI23 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM10 register. Use these pins starting from the upper bit.

Reset signal generation sets this port to input mode.

To use P100/ANI18 to P105/ANI23 as digital output pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the output mode by using the PM10 register. Use these pins starting from the upper bit.

To use P100/ANI18 to P105/ANI23 as analog input pins, set them in the analog input mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM10 register. Use these pins starting from the lower bit.

Reset signal generation sets this port to analog input mode.

**Table 4-19. Settings of Registers When Using Port 10**

| Pin Name | | PM10x | PITHL10x | Alternate Function Setting Note 2 | Remark |
|---|---|---|---|---|---|
| Name | I/O | | | | |
| P100 Note 3 | Input | 1 | — | × | |
| | Output | 0 | — | × | |
| P101 Note 3 | Input | 1 | — | × | |
| | Output | 0 | — | × | |
| P102 Note 3 | Input | 1 | — | × | |
| | Output | 0 | — | × | |
| P103 Note 3 | Input | 1 | — | × | |
| | Output | 0 | — | × | |
| P104 Note 3 | Input | 1 | — | × | |
| | Output | 0 | — | × | |
| P105 Note 3 | Input | 1 | — | × | |
| | Output | 0 | — | × | |
| P106 | Input | 1 | — | × | |
| | Output | 0 | — | (LTXD1 output = 1) Note 1 | |
| P107 | Input | 1 | 0 | × | CMOS input (Schmitt1 input) |
| | | | 1 | × | CMOS input (Schmitt3 input) |
| | Output | 0 | × | × | |

**Notes 1.** When a pin sharing the serial data output function of the LIN is to be used as a general-purpose port pin, operation of the corresponding LIN must be stopped.

**2.** Functions in parentheses can be assigned via settings in the peripheral I/O redirection register 4 (PIOR4).

**3.** These are the settings of registers in the case where setting of the A/D port configuration register (ADPC) is to select “D” (digital I/O) for the target pin. See Table 4-20 when using this pin as an analog input.

**Remark** ×: Don't care
PM10x: Port mode register 10
PITHL10x: Port input threshold control register 10

**Table 4-20. Settings of Registers When Using Pins of Port 10 as Analog Inputs**

| Pin Name | | PM10x | ADPC | Alternate Function Setting | Remark |
|---|---|---|---|---|---|
| Name | I/O | | | | |
| P10n | Input | 1 | 14 to n + 14H (n = 0 to 4), 00H (n = 5) | — | To use P10n as an analog input, use these pins from the lower bit. |

**Remarks 1.** PM10x: Port mode register 9

ADPC: A/D port configuration register

**2.** n = 0 to 5

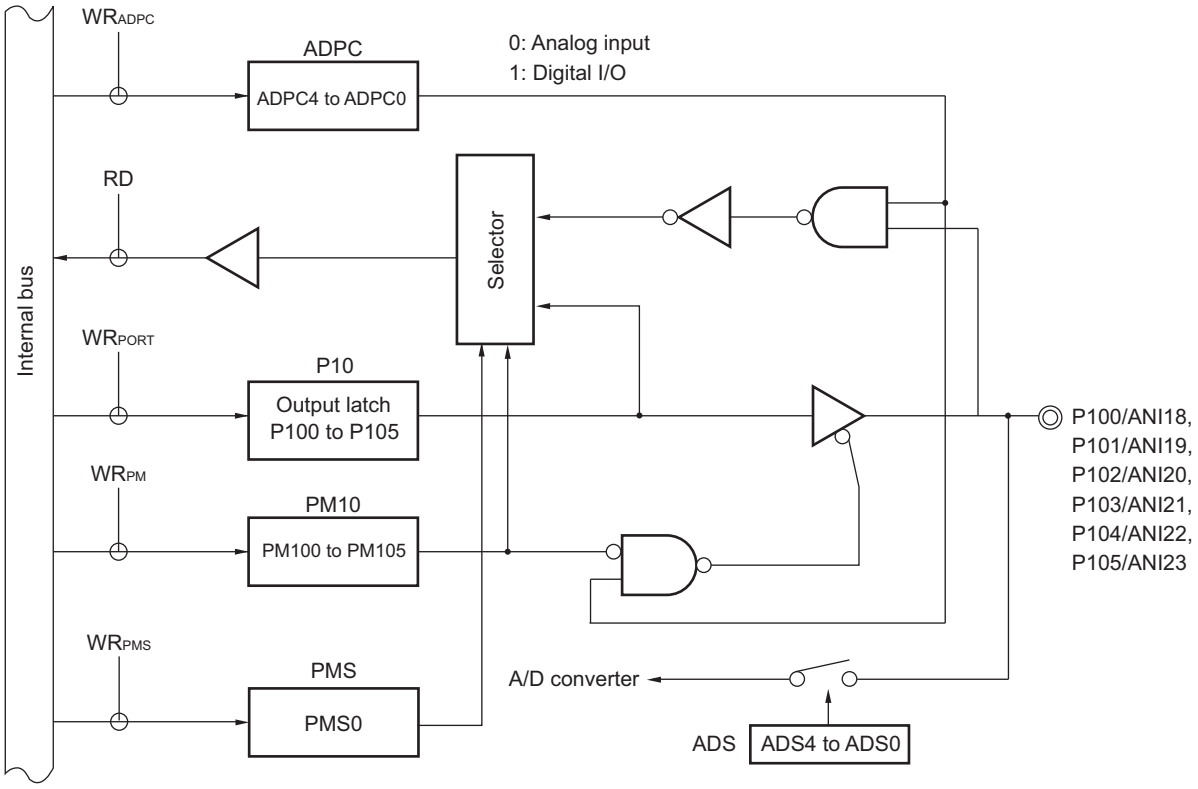
**Table 4-21. Setting Functions of P100/ANI18 to P105/ANI23 Pins**

| ADPC Register | PM10 Register | ADS Register | P100/ANI18 to P105/ANI23 Pins |
|---|---|---|---|
| Digital I/O selection | Input mode | — | Digital input |
| | Output mode | — | Digital output |
| Analog input selection | Input mode | Selects ANI. | Analog input (to be converted) |
| | | Does not select ANI. | Analog input (not to be converted) |
| | Output mode | Selects ANI. | Setting prohibited |
| | | Does not select ANI. | |

Reset signal generation sets P100/ANI18 to P105/ANI23 to analog input mode.

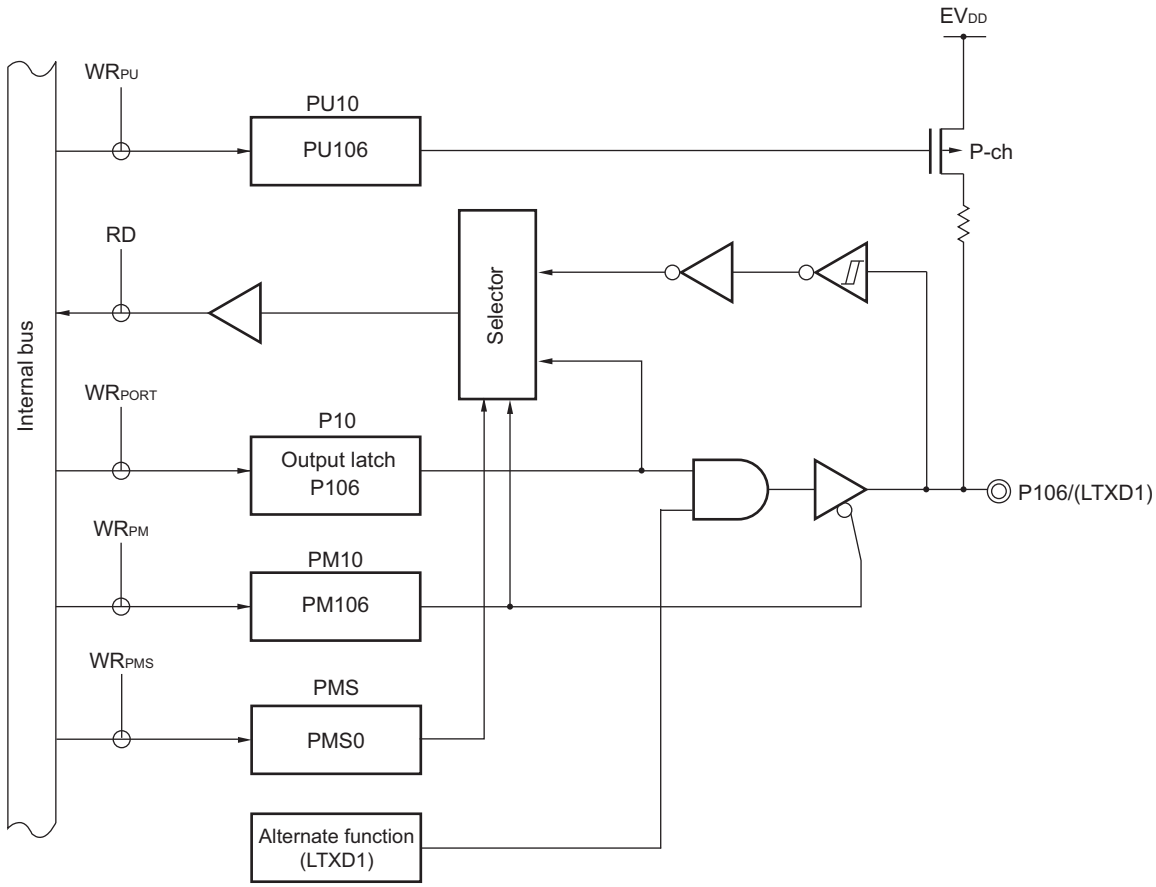
Figures 4-58 to 4-60 show block diagrams of port 10 for 100-pin products.

**Figure 4-58. Block Diagram of P100 to P105**



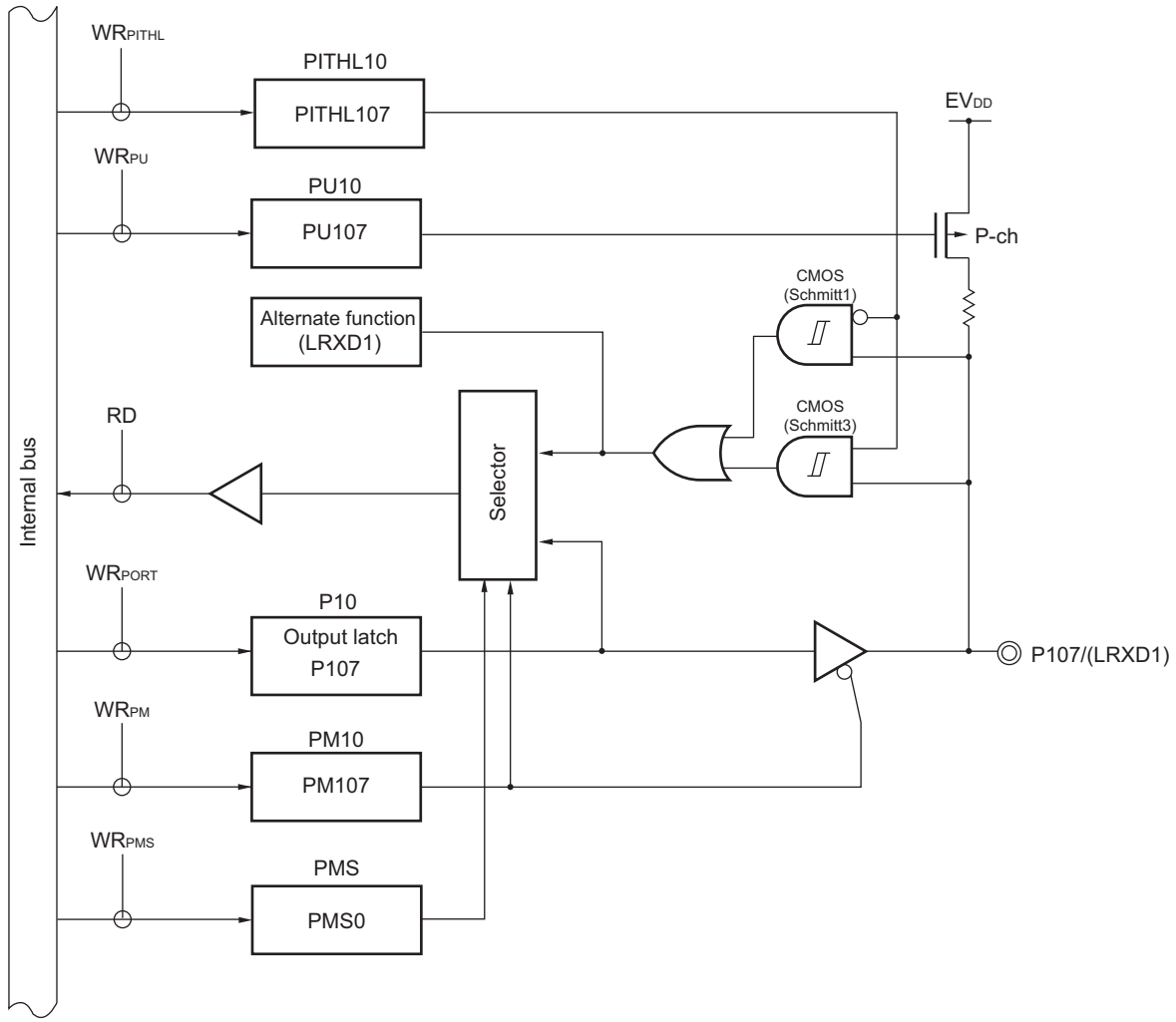

P10: Port register 10
PM10: Port mode register 10
PMS: Port mode select register
ADPC: A/D port configuration register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-59. Block Diagram of P106**




P10: Port register 10
PM10: Port mode register 10
PMS: Port mode select register
PU10: Pull-up resistor option register 10
RD: Read signal
WRxx: Write signal

**Figure 4-60. Block Diagram of P107**




P10: Port register 10
PM10: Port mode register 10
PMS: Port mode select register
PITHL10: Port input threshold control register 10
PU10: Pull-up resistor option register 10
RD: Read signal
WRxx: Write signal

### 4.2.11 Port 12

P120 and P125 to P127 are I/O ports with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When the P120 and P125 to P127 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

Input to the P125 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 12 (PIM12). For the P125 pin input, the threshold of the input buffer can be specified in 1-bit units using the port input threshold control register 12 (PITHL12).

Output from the P120 pin can be specified as N-ch open-drain output ($EV_{DD}$ tolerance) in 1-bit units using port output mode register 12 (POM12).

When P120 and P125 are used as input pins, specify analog input or digital input in 1-bit units using port mode control register 12 (PMC12).

This port can also be used for A/D converter analog input, resonator connection for main system clock, resonator connection for subsystem clock, external clock input for main system clock, external clock input for subsystem clock, timer I/O, serial interface data output, slave select input, external interrupt request input, and SNOOZE status output.

Reset signal generation sets P120 and P125 to analog input mode and P121 to P124 to input mode.

**Table 4-22. Settings of Registers When Using Port 12 (1/2)**

| Pin Name | | PM12x | PIM12x | POM12x | PMC12x | PITHL12x | Alternate Function Setting Note 5 | Remark |
|---|---|---|---|---|---|---|---|---|
| Name | I/O | | | | | | | |
| P120 | Input | 1 | – | × | 0 | – | × | |
| | Output | 0 | – | 0 | 0 | – | TRDIOD0 output = 0 Note 1 | CMOS output |
| | | 0 | – | 1 | 0 | – | TO07 output = 0 Note 2<br>SO01 output = 1 Note 3 | N-ch O.D output |
| P121 | Input | – | – | – | – | – | OSCSEL bit of CMC register = 0<br>or EXCLK bit = 1 | |
| P122 | Input | – | – | – | – | – | OSCSEL bit of CMC register = 0 | |
| P123 | Input | – | – | – | – | – | OSCSELS bit of CMC register = 0<br>or EXCLKS bit = 1<br>or SELLOSC bit of CKSEL register = 1 | |
| P124 | Input | – | – | – | – | – | OSCSELS bit of CMC register = 0<br>or SELLOSC bit of CKSEL register = 1 | |
| P125 | Input | 1 | 0 | – | 0 | 0 | × | CMOS input (Schmitt1 input) |
| | | | | | | 1 | | CMOS input (Schmitt3 input) |
| | | 1 | 1 | – | 0 | × | × | TTL input |
| | Output | 0 | × | – | 0 | × | TRDIOB0 output = 0 Note 1<br>TO03 output = 0 Note 2<br>SNZOUT1 output = 0 Note 4 | |

(**Notes**, **Caution**, and **Remark** are listed on the next page.)

**Table 4-22. Settings of Registers When Using Port 12 (2/2)**

| Pin Name | | PM12x | PIM12x | POM12x | PMC12x | PITHL12x | Alternate Function Setting Note 5 | Remark |
|---|---|---|---|---|---|---|---|---|
| Name | I/O | | | | | | | |
| P126 | Input | 1 | – | – | – | – | × | |
| | Output | 0 | – | – | – | – | (TO01 output = 0) Note2 | |
| P127 | Input | 1 | – | – | – | – | × | |
| | Output | 0 | – | – | – | – | (TO03 output = 0) Note2 | |

**Notes 1.** When a pin sharing a timer RD function is to be used as a general-purpose port pin, the target bit for TRDIOij pin output control in the timer RD output master enable register 1 (TRDOER1) must have the same setting as its initial value (i = A, B, C, D, j = 0, 1)

**2.** When a pin sharing a timer output function of the timer array unit is to be used as a general-purpose port pin, the TOmn bit of the timer output register m (TOm) and the TOEmn bit of the timer output enable register m (TOEm) corresponding to the target unit and channel must have the same setting as in the initial state (m = 0, 1, n = 0 to 7).

**3.** When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn bit of the serial output register m (SOm), the SOEmn bit of the serial output enable register m (SOEm), and the SEmn bit of the serial channel enable status register m (SEm) corresponding to the target unit and channel must have the same setting as its initial value (m = 0, 1, n = 0, 1).

**4.** When a pin sharing the SNOOZE status output function is to be used as a general-purpose port pin, the OUTEN0 to OUTEN7 bits of the SNOOZE status output control registers 0, 1, 2, 3 (PSNZCNT0, 1, 2, 3) must have the same setting as its initial value.

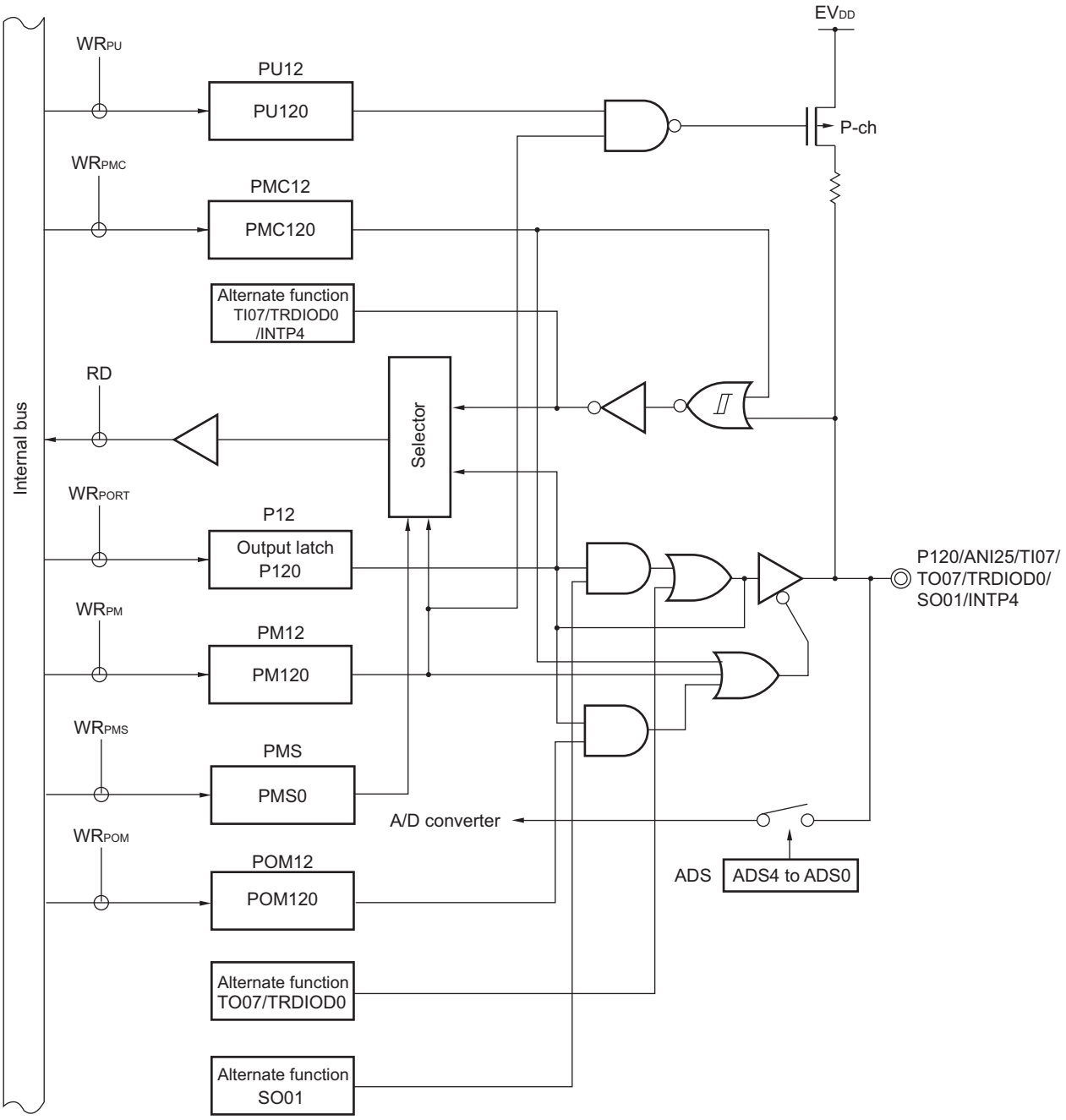
**5.** Functions in parentheses can be assigned via settings in the peripheral I/O redirection register 1 (PIOR1).

**Caution The function setting on P121 to P124 is available only once after the reset release. The port once set for connection to a resonator or an oscillator cannot be used as an input port unless the reset is performed.**

**Remark** ×: Don't care
PM12x: Port mode register 12
PIM12x: Port input mode register 12
POM12x: Port output mode register 12
PMC12x: Port mode control register 12
PITHL12x: Port input threshold control register 12

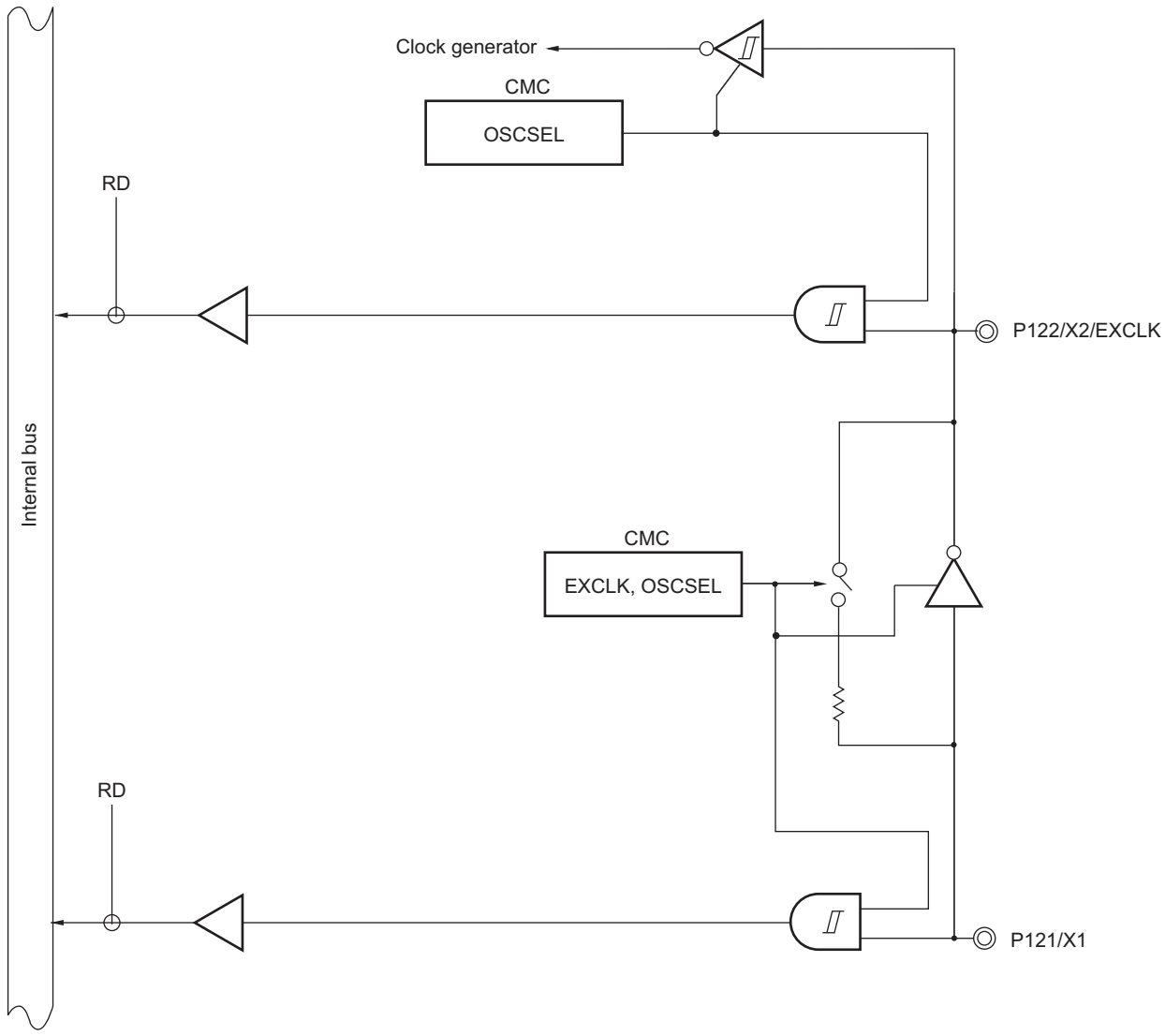
Figures 4-61 to 4-65 show block diagrams of port 12 for 100-pin products.

**Figure 4-61. Block Diagram of P120**



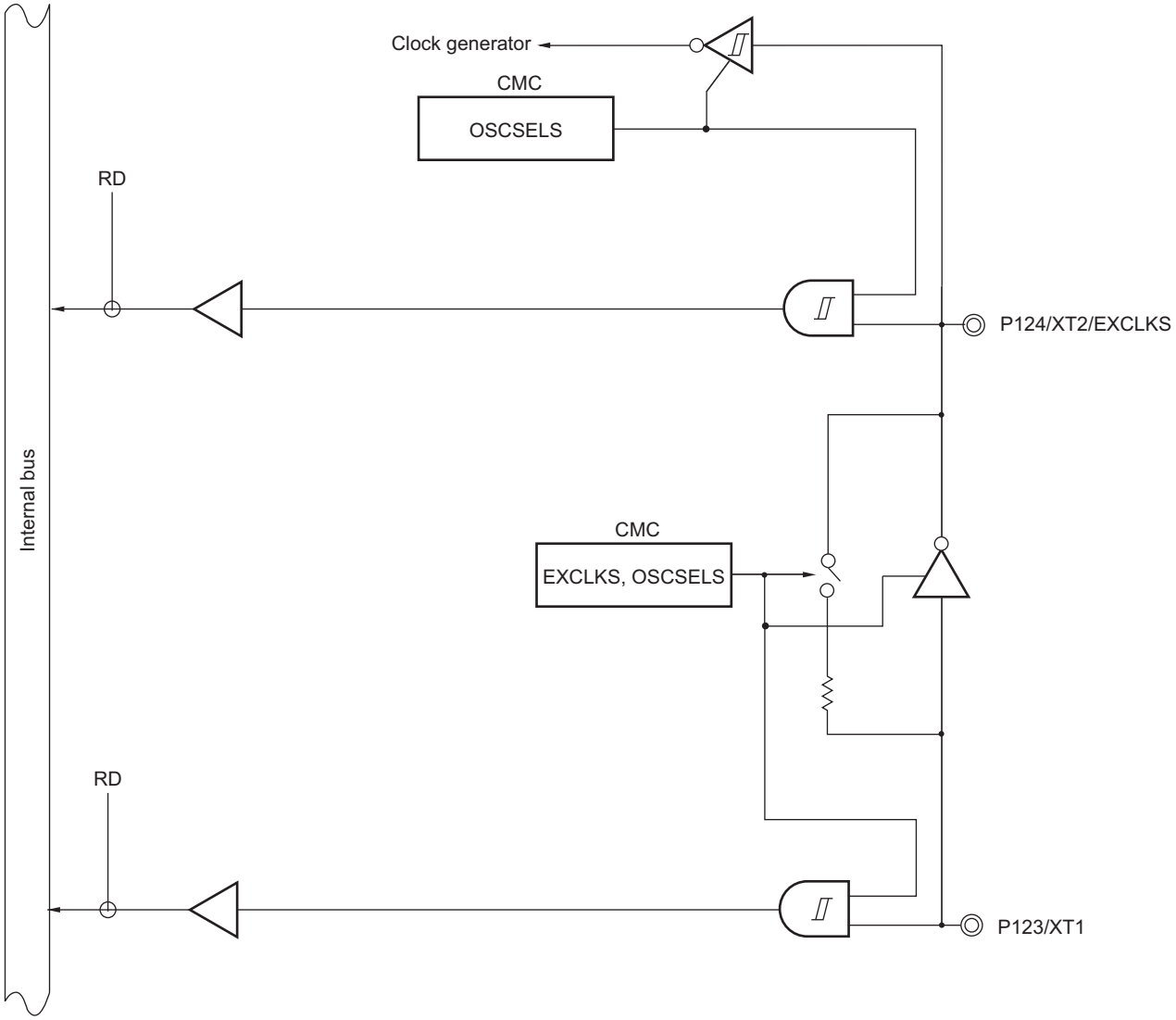

P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
POM12: Port output mode register 12
PMC12: Port mode control register 12
PMS: Port mode select register
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-62. Block Diagram of P121 and P122**



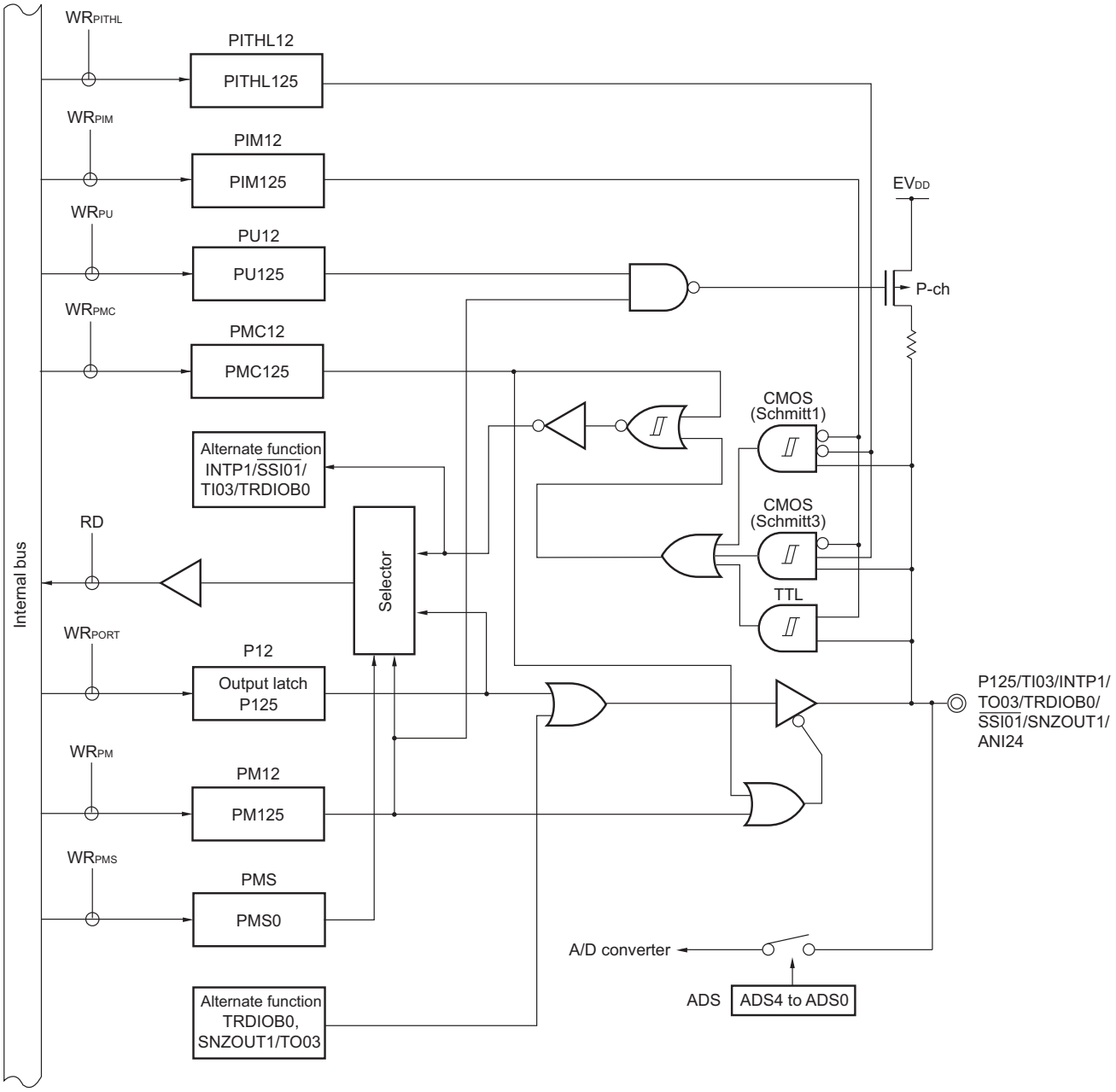

CMC: Clock operation mode control register
RD: Read signal

**Figure 4-63. Block Diagram of P123 and P124**



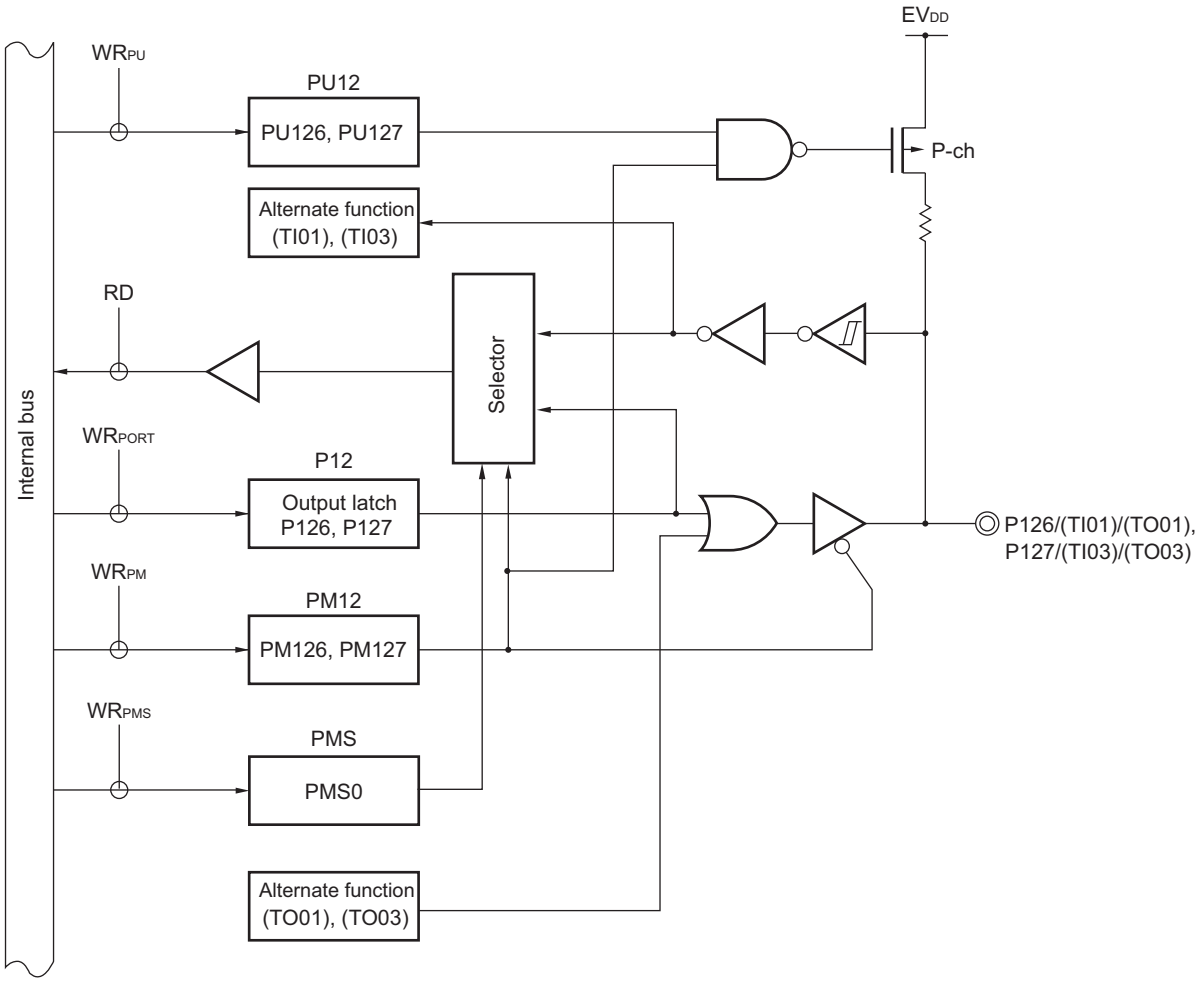

CMC: Clock operation mode control register
RD: Read signal

**Figure 4-64. Block Diagram of P125**




P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
PIM12: Port input mode register 12
PMC12: Port mode control register 12
PMS: Port mode select register
PITHL12: Port input threshold control register 12
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

**Figure 4-65. Block Diagram of P126 and P127**




P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

### 4.2.12 Port 13

P130 is a 1-bit output-only port with an output latch.

P137 is a 1-bit input-only port.

P130 is fixed to output mode, and P137 is fixed to input mode.

This port can also be used for external interrupt request input and reset output.

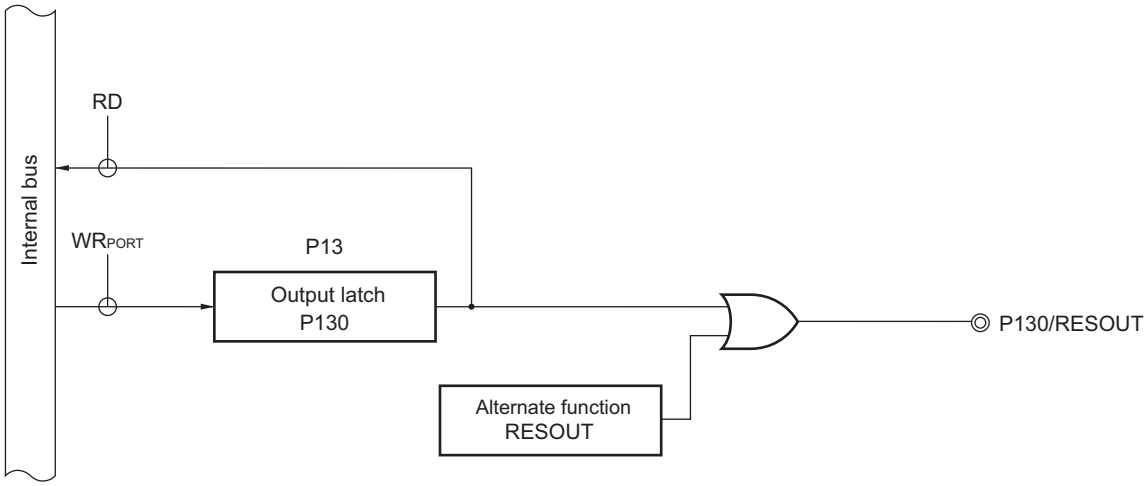
The RESOUT output can be set by an option byte.

**Table 4-23. Settings of Registers When Using Port 13**

| Pin Name | | Alternate Function Setting | Remark |
|---|---|---|---|
| Name | I/O | | |
| P130 | Output | RESOUT | |
| P137 | Input | × | |

Figures 4-66 and 4-67 show block diagrams of port 13 for 100-pin products.

**Figure 4-66. Block Diagram of P130**




P13: Port register 13
RD: Read signal
WRxx: Write signal

**Remark** When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.

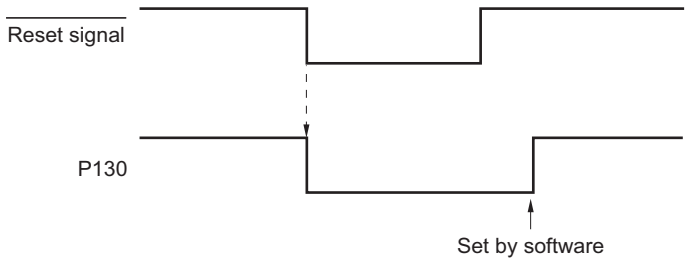
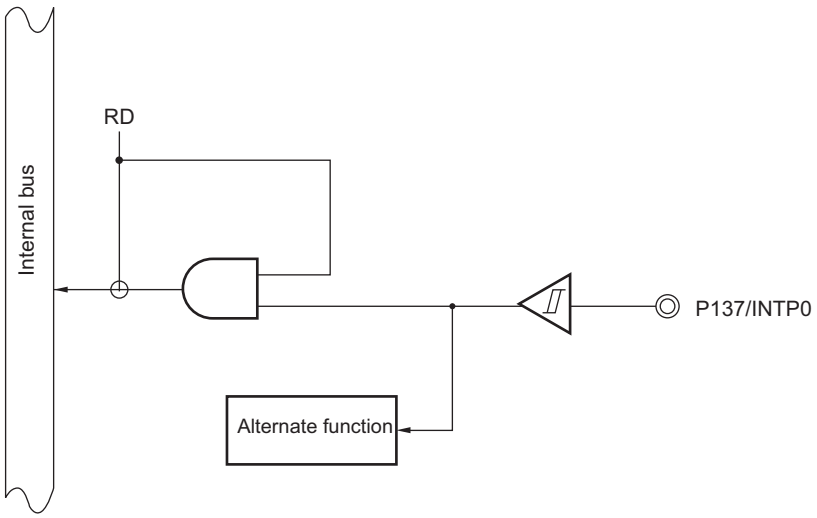


**Figure 4-67. Block Diagram of P137**




RD: Read signal

### 4.2.13 Port 14

Port 14 is an I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140 pin is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 14 (PU14).

This port can also be used for clock/buzzer output.

Reset signal generation sets P140 to input mode.

**Table 4-24. Settings of Registers When Using Port 14**

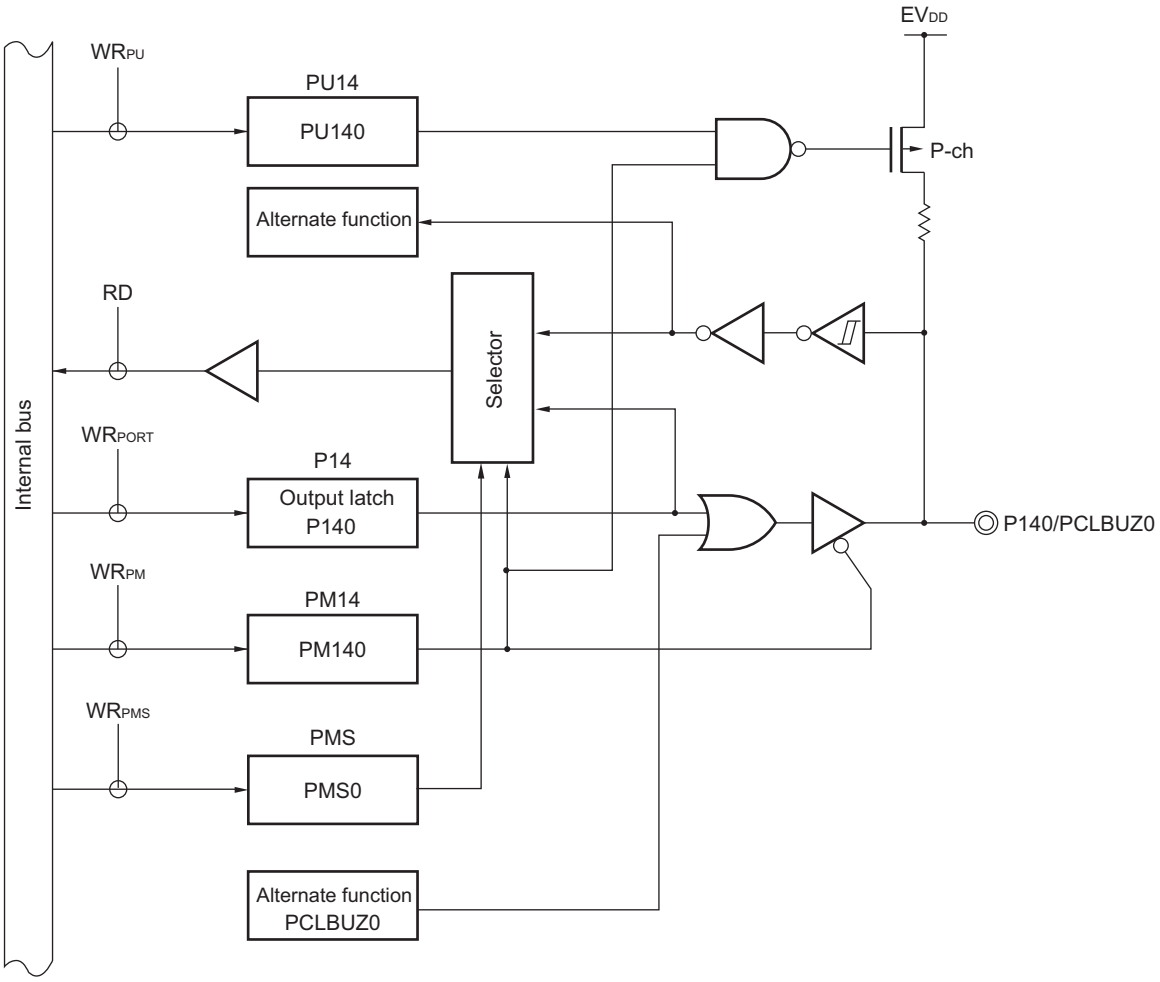
| Pin Name | | PM14x | Alternate Function Setting | Remark |
|---|---|---|---|---|
| Name | I/O | | | |
| P140 | Input | 1 | × | |
| | Output | 0 | PCLBUZ0 output = 0 **Note** | |

**Note** To use a pin multiplexed with the clock/buzzer output function as a general-purpose port, set the PCLOE0 bit in clock output select register 0 (CKS0) to the default value.

**Remark** ×: Don't care
PM14x: Port mode register 14

Figure 4-68 shows a block diagram of port 14 for 100-pin products.

**Figure 4-68. Block Diagram of P140**




P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

### 4.2.14 Port 15

Port 15 is an I/O port with an output latch. Port 15 can be set to the input mode or output mode in 1-bit units using port mode register 15 (PM15). When the P150 to P157 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 15 (PU15).

For the P150, P152, and P153 pin input, the threshold of the input buffer can be specified in 1-bit units using the port input threshold control register 15 (PITHL15).

This port can also be used for data I/O and clock I/O for serial interface (CSI), slave select input, and SNOOZE status output.

Reset signal generation sets P150 to P157 to input mode.

**Table 4-25. Settings of Registers When Using Port 15**

| Pin Name | | PM15x | PITHL15x | Alternate Function Setting Note 4 | Remark |
|---|---|---|---|---|---|
| Name | I/O | | | | |
| P150 | Input | 1 | 0 | × | CMOS input (Schmitt1 input) |
| | | | 1 | × | CMOS input (Schmitt3 input) |
| | Output | 0 | × | × | |
| P151 | Input | 1 | – | × | |
| | Output | 0 | – | (SO11 output = 1) Note 1 | |
| P152 | Input | 1 | 0 | × | CMOS input (Schmitt1 input) |
| | | | 1 | × | CMOS input (Schmitt3 input) |
| | Output | 0 | × | × | |
| P153 | Input | 1 | 0 | × | CMOS input (Schmitt1 input) |
| | | | 1 | × | CMOS input (Schmitt3 input) |
| | Output | 0 | × | ($\overline{\text{SCK11}}$ output = 1) Note 3 | |
| P154 | Input | 1 | – | × | |
| | Output | 0 | – | (SNZOUT7 output = 0) Note 2 | |
| P155 | Input | 1 | – | × | |
| | Output | 0 | – | (SNZOUT6 output = 0) Note 2 | |
| P156 | Input | 1 | – | × | |
| | Output | 0 | – | (SNZOUT5 output = 0) Note 2 | |
| P157 | Input | 1 | – | × | |
| | Output | 0 | – | (SNZOUT4 output = 0) Note 2 | |

**Notes 1.** When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn bit of the serial output register m (SOm), the SOEmn bit of the serial output enable register m (SOEm), and the SEmn bit of the serial channel enable status register m (SEm) corresponding to the target unit and channel must have the same setting as its initial value (m = 0, 1, n = 0, 1).

**2.** When a pin sharing the SNOOZE status output function is to be used as a general-purpose port pin, the OUTEN0 to OUTEN7 bits of the SNOOZE status output control registers 0, 1, 2, 3 (PSNZCNT0, 1, 2, 3) must have the same setting as its initial value.

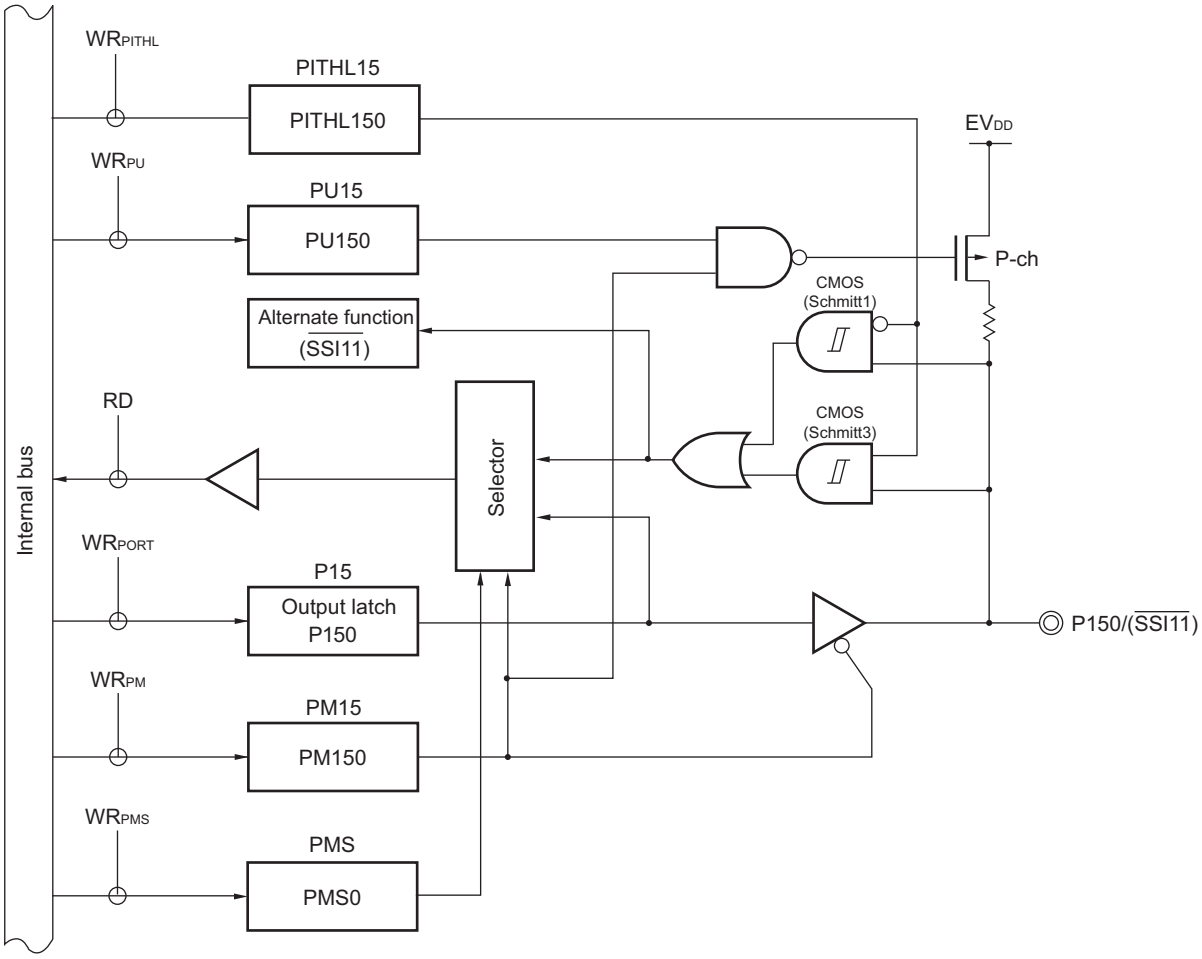
**3.** When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the CKOmn bit of the serial output register m (SOm), the SOEmn bit of the serial output enable register m (SOEm), and the SEmn bit of the serial channel enable status register m (SEm) corresponding to the target unit and channel must have the same setting as its initial value (m = 0, 1, n = 0, 1).

**4.** Functions in parentheses can be assigned via settings in the peripheral I/O redirection registers 4, 6 (PIOR4, PIOR6).

**Remark** ×: Don't care
PM15x: Port mode register 15
PITHL15x: Port input threshold control register 15

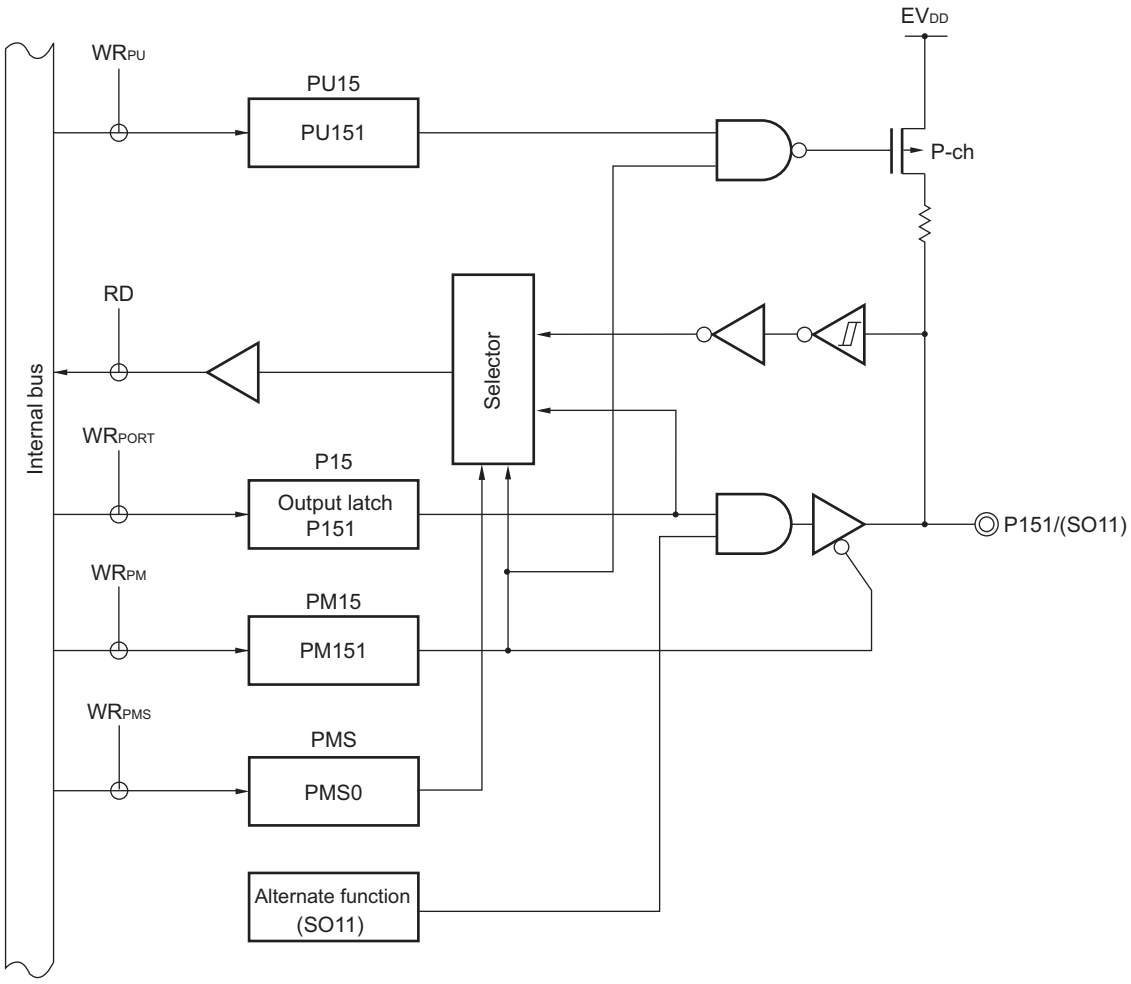
Figure 4-69 to 4-73 show block diagrams of port 15 for 100-pin products.

**Figure 4-69. Block Diagram of P150**



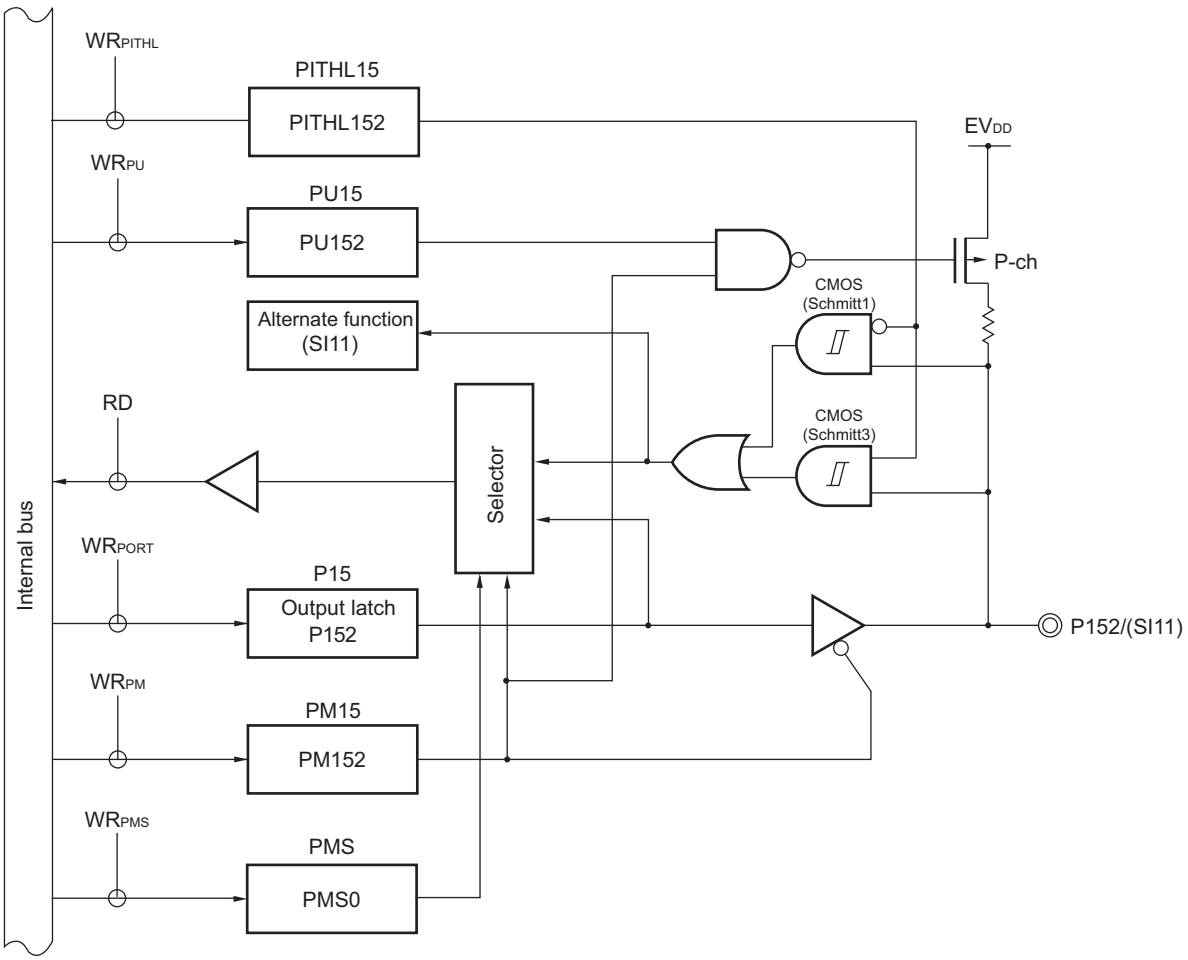

P15: Port register 15
PU15: Pull-up resistor option register 15
PM15: Port mode register 15
PMS: Port mode select register
PITHL15: Port input threshold control register 15
RD: Read signal
WRxx: Write signal

**Figure 4-70. Block Diagram of P151**



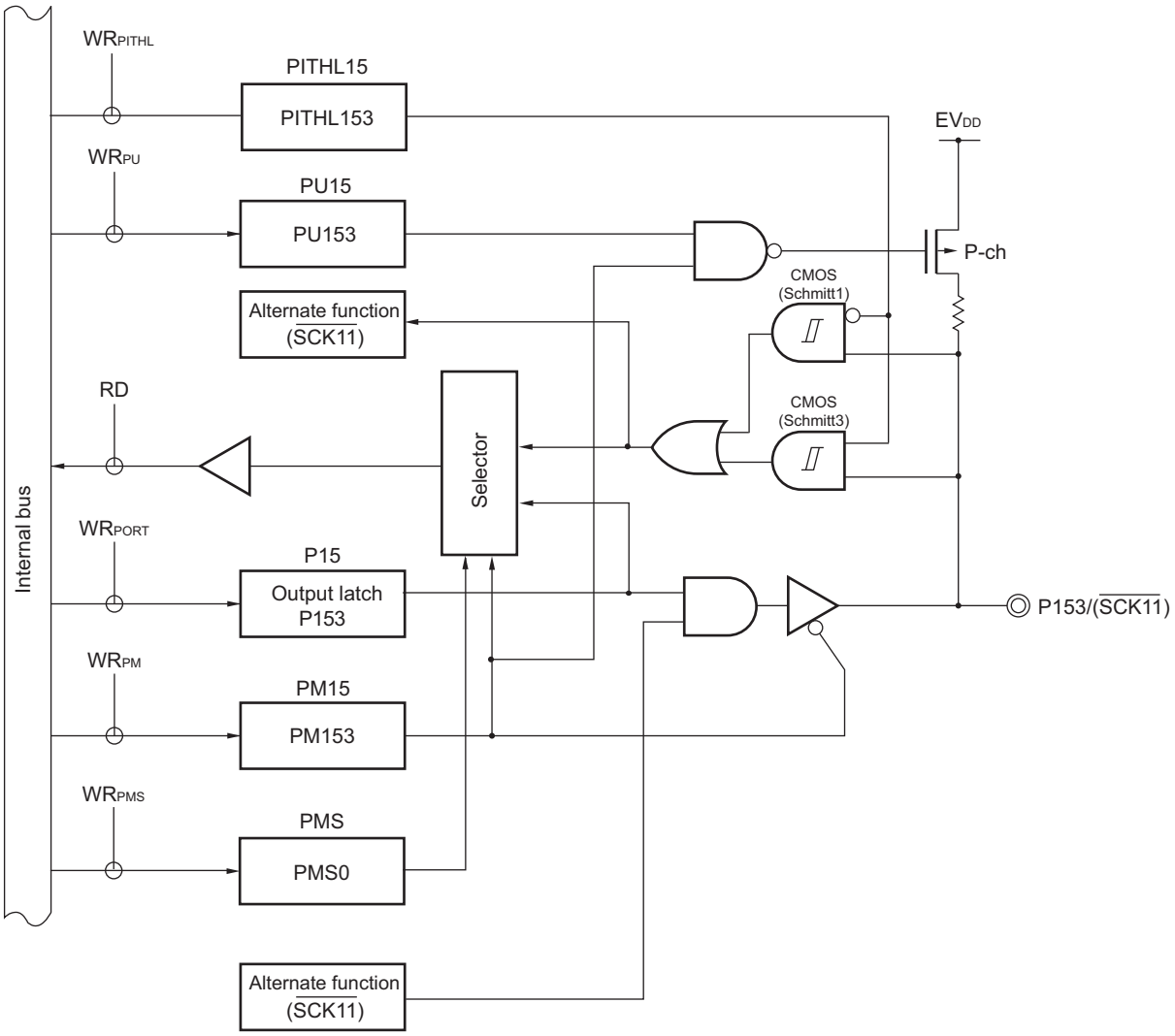

P15: Port register 15
PU15: Pull-up resistor option register 15
PM15: Port mode register 15
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

**Figure 4-71. Block Diagram of P152**



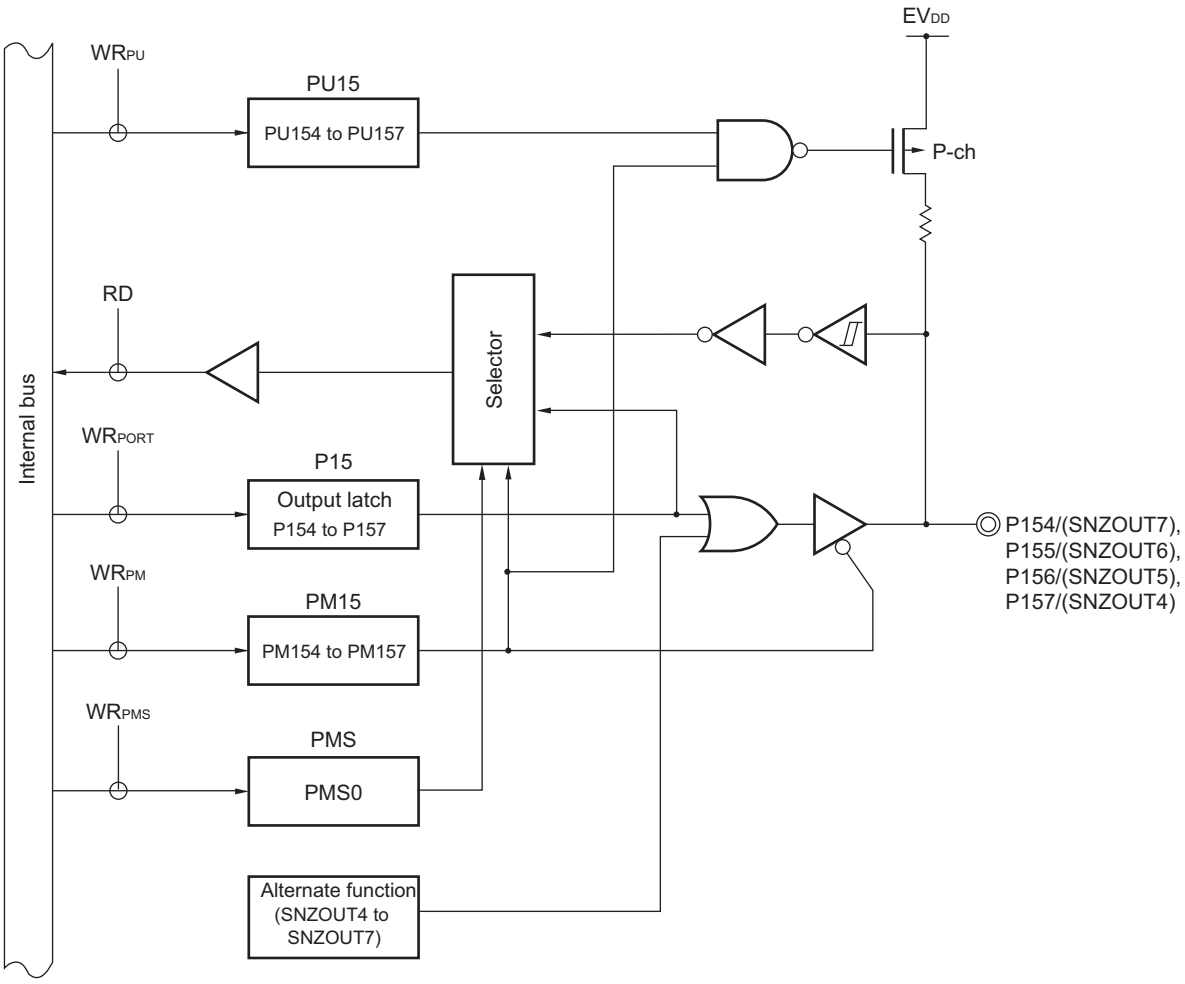

P15: Port register 15
PU15: Pull-up resistor option register 15
PM15: Port mode register 15
PMS: Port mode select register
PITHL15: Port input threshold control register 15
RD: Read signal
WRxx: Write signal

**Figure 4-72. Block Diagram of P153**




P15: Port register 15
PU15: Pull-up resistor option register 15
PM15: Port mode register 15
PMS: Port mode select register
PITHL15: Port input threshold control register 15
RD: Read signal
WRxx: Write signal

**Figure 4-73. Block Diagram of P154 to P157**




P15: Port register 15
PU15: Pull-up resistor option register 15
PM15: Port mode register 15
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

## 4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control registers (PMCxx)
- A/D port configuration register (ADPC)
- Peripheral I/O redirection registers (PIORx)
- Port input threshold control register (PITHLxx)
- Port output slew rate select registers (PSRSEL)
- SNOOZE status output control registers 0 to 3 (PSNZCNT0 to PSNZCNT3)
- Port mode select register (PMS)

**Table 4-26. Port Configuration of Group A Products (64-pin products) (1/4)**

| Port Name | Port Bit | Output Latch | I/O Mode Control | Pull-up Control | Input Type Control | Output Type Control | Operating Mode Control | Input Threshold Control |
|---|---|---|---|---|---|---|---|---|
| PORT0 | 0 | P0.0 | PM0.0 | PU0.0 | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT1 | 0 | P1.0 | PM1.0 | PU1.0 | — | — | — | — |
| | 1 | P1.1 | PM1.1 | PU1.1 | — | — | — | — |
| | 2 | P1.2 | PM1.2 | PU1.2 | — | — | — | — |
| | 3 | P1.3 | PM1.3 | PU1.3 | PIM1.3 | POM1.3 | — | PITHL1.3 |
| | 4 | P1.4 | PM1.4 | PU1.4 | PIM1.4 | POM1.4 | — | PITHL1.4 |
| | 5 | P1.5 | PM1.5 | PU1.5 | — | POM1.5 | — | — |
| | 6 | P1.6 | PM1.6 | PU1.6 | PIM1.6 | POM1.6 | — | PITHL1.6 |
| | 7 | P1.7 | PM1.7 | PU1.7 | PIM1.7 | POM1.7 | — | PITHL1.7 |
| PORT2 | 0 | — | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |

**Remark** —: Not provided.

**Table 4-26. Port Configuration of Group A Products (64-pin products) (2/4)**

| Port Name | Port Bit | Output Latch | I/O Mode Control | Pull-up Control | Input Type Control | Output Type Control | Operating Mode Control | Input Threshold Control |
|---|---|---|---|---|---|---|---|---|
| PORT3 | 0 | P3.0 | PM3.0 | PU3.0 | PIM3.0 | — | — | PITHL3.0 |
| | 1 | P3.1 | PM3.1 | PU3.1 | — | — | — | — |
| | 2 | P3.2 | PM3.2 | PU3.2 | — | — | — | — |
| | 3 | P3.3 | PM3.3 | — | — | — | — | — |
| | 4 | P3.4 | PM3.4 | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT4 | 0 | P4.0 | PM4.0 | PU4.0 | — | — | — | — |
| | 1 | P4.1 | PM4.1 | PU4.1 | — | — | — | — |
| | 2 | P4.2 | PM4.2 | PU4.2 | — | — | — | — |
| | 3 | P4.3 | PM4.3 | PU4.3 | — | — | — | PITHL4.3 |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT5 | 0 | P5.0 | PM5.0 | PU5.0 | — | — | — | PITHL5.0 |
| | 1 | P5.1 | PM5.1 | PU5.1 | — | — | — | — |
| | 2 | P5.2 | PM5.2 | PU5.2 | — | — | — | PITHL5.2 |
| | 3 | P5.3 | PM5.3 | PU5.3 | — | — | — | PITHL5.3 |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT6 | 0 | P6.0 | PM6.0 | PU6.0 | — | POM6.0 | — | PITHL6.0 |
| | 1 | P6.1 | PM6.1 | PU6.1 | — | POM6.1 | — | PITHL6.1 |
| | 2 | P6.2 | PM6.2 | PU6.2 | — | POM6.2 | — | — |
| | 3 | P6.3 | PM6.3 | PU6.3 | — | POM6.3 | — | PITHL6.3 |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT7 | 0 | P7.0 | PM7.0 | PU7.0 | — | — | — | — |
| | 1 | P7.1 | PM7.1 | PU7.1 | — | — | — | PITHL7.1 |
| | 2 | P7.2 | PM7.2 | PU7.2 | — | — | — | — |
| | 3 | P7.3 | PM7.3 | PU7.3 | — | — | — | — |
| | 4 | P7.4 | PM7.4 | PU7.4 | — | — | — | — |
| | 5 | P7.5 | PM7.5 | PU7.5 | — | — | — | — |
| | 6 | P7.6 | PM7.6 | PU7.6 | — | — | — | — |
| | 7 | P7.7 | PM7.7 | PU7.7 | — | — | — | — |

**Remark** —: Not provided.

**Table 4-26. Port Configuration of Group A Products (64-pin products) (3/4)**

| Port Name | Port Bit | Output Latch | I/O Mode Control | Pull-up Control | Input Type Control | Output Type Control | Operating Mode Control | Input Threshold Control |
|---|---|---|---|---|---|---|---|---|
| PORT8 | 0 | P8.0 | PM8.0 | — | — | — | — | — |
| | 1 | P8.1 | PM8.1 | — | — | — | — | — |
| | 2 | P8.2 | PM8.2 | — | — | — | — | — |
| | 3 | P8.3 | PM8.3 | — | — | — | — | — |
| | 4 | P8.4 | PM8.4 | — | — | — | — | — |
| | 5 | P8.5 | PM8.5 | — | — | — | — | — |
| | 6 | P8.6 | PM8.6 | — | — | — | — | — |
| | 7 | P8.7 | PM8.7 | — | — | — | — | — |
| PORT9 | 0 | P9.0 | PM9.0 | — | — | — | — | — |
| | 1 | P9.1 | PM9.1 | — | — | — | — | — |
| | 2 | P9.2 | PM9.2 | PU9.2 | — | — | — | — |
| | 3 | P9.3 | PM9.3 | PU9.3 | — | — | — | — |
| | 4 | P9.4 | PM9.4 | PU9.4 | — | — | — | — |
| | 5 | P9.5 | PM9.5 | PU9.5 | — | — | — | — |
| | 6 | P9.6 | PM9.6 | PU9.6 | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT10 | 0 | — | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT11 | 0 | — | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT12 | 0 | P12.0 | PM12.0 | PU12.0 | — | POM12.0 | — | — |
| | 1 | P12.1 | — | — | — | — | — | — |
| | 2 | P12.2 | — | — | — | — | — | — |
| | 3 | P12.3 | — | — | — | — | — | — |
| | 4 | P12.4 | — | — | — | — | — | — |
| | 5 | P12.5 | PM12.5 | PU12.5 | PIM12.5 | — | — | PITHL12.5 |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |

**Remark** —: Not provided.

**Table 4-26. Port Configuration of Group A Products (64-pin products) (4/4)**

| Port Name | Port Bit | Output Latch | I/O Mode Control | Pull-up Control | Input Type Control | Output Type Control | Operating Mode Control | Input Threshold Control |
|---|---|---|---|---|---|---|---|---|
| PORT13 | 0 | P13.0 | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | P13.7 | — | — | — | — | — | — |
| PORT14 | 0 | P14.0 | PM14.0 | PU14.0 | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT15 | 0 | — | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |

**Remark** —: Not provided.

**Table 4-27. Port Configuration of Products of Groups B to D (80-pin products) (1/4)**

| Port Name | Port Bit | Output Latch | I/O Mode Control | Pull-up Control | Input Type Control | Output Type Control | Operating Mode Control | Input Threshold Control |
|---|---|---|---|---|---|---|---|---|
| PORT0 | 0 | P0.0 | PM0.0 | PU0.0 | — | — | — | — |
| | 1 | P0.1 | PM0.1 | PU0.1 | — | — | — | — |
| | 2 | P0.2 | PM0.2 | PU0.2 | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT1 | 0 | P1.0 | PM1.0 | PU1.0 | PIM1.0 | POM1.0 | — | PITHL1.0 |
| | 1 | P1.1 | PM1.1 | PU1.1 | PIM1.1 | POM1.1 | — | PITHL1.1 |
| | 2 | P1.2 | PM1.2 | PU1.2 | — | POM1.2 | — | — |
| | 3 | P1.3 | PM1.3 | PU1.3 | PIM1.3 | POM1.3 | — | PITHL1.3 |
| | 4 | P1.4 | PM1.4 | PU1.4 | PIM1.4 | POM1.4 | — | PITHL1.4 |
| | 5 | P1.5 | PM1.5 | PU1.5 | — | POM1.5 | — | — |
| | 6 | P1.6 | PM1.6 | PU1.6 | PIM1.6 | POM1.6 | — | PITHL1.6 |
| | 7 | P1.7 | PM1.7 | PU1.7 | PIM1.7 | POM1.7 | — | PITHL1.7 |
| PORT2 | 0 | — | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT3 | 0 | P3.0 | PM3.0 | PU3.0 | PIM3.0 | — | — | PITHL3.0 |
| | 1 | P3.1 | PM3.1 | PU3.1 | — | — | — | — |
| | 2 | P3.2 | PM3.2 | PU3.2 | — | — | — | — |
| | 3 | P3.3 | PM3.3 | — | — | — | — | — |
| | 4 | P3.4 | PM3.4 | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT4 | 0 | P4.0 | PM4.0 | PU4.0 | — | — | — | — |
| | 1 | P4.1 | PM4.1 | PU4.1 | — | — | — | — |
| | 2 | P4.2 | PM4.2 | PU4.2 | — | — | — | — |
| | 3 | P4.3 | PM4.3 | PU4.3 | — | — | — | PITHL4.3 |
| | 4 | P4.4 | PM4.4 | PU4.4 | — | — | — | — |
| | 5 | P4.5 | PM4.5 | PU4.5 | — | — | — | — |
| | 6 | P4.6 | PM4.6 | PU4.6 | — | — | — | — |
| | 7 | P4.7 | PM4.7 | PU4.7 | — | — | — | — |

**Remark** —: Not provided.

**Table 4-27. Port Configuration of Products of Groups B to D (80-pin products) (2/4)**

| Port Name | Port Bit | Output Latch | I/O Mode Control | Pull-up Control | Input Type Control | Output Type Control | Operating Mode Control | Input Threshold Control |
|---|---|---|---|---|---|---|---|---|
| PORT5 | 0 | P5.0 | PM5.0 | PU5.0 | — | — | — | PITHL5.0 |
| | 1 | P5.1 | PM5.1 | PU5.1 | — | — | — | — |
| | 2 | P5.2 | PM5.2 | PU5.2 | — | — | — | PITHL5.2 |
| | 3 | P5.3 | PM5.3 | PU5.3 | — | — | — | PITHL5.3 |
| | 4 | P5.4 | PM5.4 | PU5.4 | PIM5.4 | — | — | PITHL5.4 |
| | 5 | P5.5 | PM5.5 | PU5.5 | — | — | — | — |
| | 6 | P5.6 | PM5.6 | PU5.6 | — | — | — | — |
| | 7 | P5.7 | PM5.7 | PU5.7 | — | — | — | — |
| PORT6 | 0 | P6.0 | PM6.0 | PU6.0 | — | POM6.0 | — | PITHL6.0 |
| | 1 | P6.1 | PM6.1 | PU6.1 | — | POM6.1 | — | PITHL6.1 |
| | 2 | P6.2 | PM6.2 | PU6.2 | PIM6.2 | POM6.2 | — | PITHL6.2 |
| | 3 | P6.3 | PM6.3 | PU6.3 | PIM6.3 | POM6.3 | — | PITHL6.3 |
| | 4 | P6.4 | PM6.4 | PU6.4 | — | — | — | — |
| | 5 | P6.5 | PM6.5 | PU6.5 | — | — | — | — |
| | 6 | P6.6 | PM6.6 | PU6.6 | — | — | — | — |
| | 7 | P6.7 | PM6.7 | PU6.7 | — | — | — | — |
| PORT7 | 0 | P7.0 | PM7.0 | PU7.0 | PIM7.0 | POM7.0 | — | PITHL7.0 |
| | 1 | P7.1 | PM7.1 | PU7.1 | PIM7.1 | POM7.1 | — | PITHL7.1 |
| | 2 | P7.2 | PM7.2 | PU7.2 | — | POM7.2 | — | — |
| | 3 | P7.3 | PM7.3 | PU7.3 | PIM7.3 | — | — | PITHL7.3 |
| | 4 | P7.4 | PM7.4 | PU7.4 | — | — | — | — |
| | 5 | P7.5 | PM7.5 | PU7.5 | — | — | — | PITHL7.5 |
| | 6 | P7.6 | PM7.6 | PU7.6 | — | — | — | PITHL7.6 |
| | 7 | P7.7 | PM7.7 | PU7.7 | — | — | — | PITHL7.7 |
| PORT8 | 0 | P8.0 | PM8.0 | — | — | — | — | — |
| | 1 | P8.1 | PM8.1 | — | — | — | — | — |
| | 2 | P8.2 | PM8.2 | — | — | — | — | — |
| | 3 | P8.3 | PM8.3 | — | — | — | — | — |
| | 4 | P8.4 | PM8.4 | — | — | — | — | — |
| | 5 | P8.5 | PM8.5 | — | — | — | — | — |
| | 6 | P8.6 | PM8.6 | — | — | — | — | — |
| | 7 | P8.7 | PM8.7 | — | — | — | — | — |
| PORT9 | 0 | P9.0 | PM9.0 | — | — | — | — | — |
| | 1 | P9.1 | PM9.1 | — | — | — | — | — |
| | 2 | P9.2 | PM9.2 | — | — | — | — | — |
| | 3 | P9.3 | PM9.3 | — | — | — | — | — |
| | 4 | P9.4 | PM9.4 | — | — | — | — | — |
| | 5 | P9.5 | PM9.5 | — | — | — | — | — |
| | 6 | P9.6 | PM9.6 | PU9.6 | — | — | PMC9.6 | — |
| | 7 | P9.7 | PM9.7 | PU9.7 | — | — | PMC9.7 | — |

**Remark** —: Not provided.

**Table 4-27. Port Configuration of Products of Groups B to D (80-pin products) (3/4)**

| Port Name | Port Bit | Output Latch | I/O Mode Control | Pull-up Control | Input Type Control | Output Type Control | Operating Mode Control | Input Threshold Control |
|---|---|---|---|---|---|---|---|---|
| PORT10 | 0 | — | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT11 | 0 | — | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT12 | 0 | P12.0 | PM12.0 | PU12.0 | — | POM12.0 | PMC12.0 | — |
| | 1 | P12.1 | — | — | — | — | — | — |
| | 2 | P12.2 | — | — | — | — | — | — |
| | 3 | P12.3 | — | — | — | — | — | — |
| | 4 | P12.4 | — | — | — | — | — | — |
| | 5 | P12.5 | PM12.5 | PU12.5 | PIM12.5 | — | PMC12.5 | PITHL12.5 |
| | 6 | P12.6 | PM12.6 | PU12.6 | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT13 | 0 | P13.0 | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | P13.7 | — | — | — | — | — | — |
| PORT14 | 0 | P14.0 | PM14.0 | PU14.0 | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |

**Remark** —: Not provided.

**Table 4-27. Port Configuration of Products of Groups B to D (80-pin products) (4/4)**

| Port Name | Port Bit | Output Latch | I/O Mode Control | Pull-up Control | Input Type Control | Output Type Control | Operating Mode Control | Input Threshold Control |
|---|---|---|---|---|---|---|---|---|
| PORT15 | 0 | — | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |

**Remark** —: Not provided.

**Table 4-28. Port Configuration of Group E Products (100-pin products) (1/4)**

| Port Name | Port Bit | Output Latch | I/O Mode Control | Pull-up Control | Input Type Control | Output Type Control | Operating Mode Control | Input Threshold Control |
|---|---|---|---|---|---|---|---|---|
| PORT0 | 0 | P0.0 | PM0.0 | PU0.0 | — | — | — | — |
| | 1 | P0.1 | PM0.1 | PU0.1 | — | — | — | — |
| | 2 | P0.2 | PM0.2 | PU0.2 | — | — | — | — |
| | 3 | P0.3 | PM0.3 | PU0.3 | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT1 | 0 | P1.0 | PM1.0 | PU1.0 | PIM1.0 | POM1.0 | — | PITHL1.0 |
| | 1 | P1.1 | PM1.1 | PU1.1 | PIM1.1 | POM1.1 | — | PITHL1.1 |
| | 2 | P1.2 | PM1.2 | PU1.2 | — | POM1.2 | — | — |
| | 3 | P1.3 | PM1.3 | PU1.3 | PIM1.3 | POM1.3 | — | PITHL1.3 |
| | 4 | P1.4 | PM1.4 | PU1.4 | PIM1.4 | POM1.4 | — | PITHL1.4 |
| | 5 | P1.5 | PM1.5 | PU1.5 | — | POM1.5 | — | — |
| | 6 | P1.6 | PM1.6 | PU1.6 | PIM1.6 | POM1.6 | — | PITHL1.6 |
| | 7 | P1.7 | PM1.7 | PU1.7 | PIM1.7 | POM1.7 | — | PITHL1.7 |
| PORT2 | 0 | — | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT3 | 0 | P3.0 | PM3.0 | PU3.0 | PIM3.0 | — | — | PITHL3.0 |
| | 1 | P3.1 | PM3.1 | PU3.1 | — | — | — | — |
| | 2 | P3.2 | PM3.2 | PU3.2 | — | — | — | — |
| | 3 | P3.3 | PM3.3 | — | — | — | — | — |
| | 4 | P3.4 | PM3.4 | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT4 | 0 | P4.0 | PM4.0 | PU4.0 | — | — | — | — |
| | 1 | P4.1 | PM4.1 | PU4.1 | — | — | — | — |
| | 2 | P4.2 | PM4.2 | PU4.2 | — | — | — | — |
| | 3 | P4.3 | PM4.3 | PU4.3 | — | — | — | PITHL4.3 |
| | 4 | P4.4 | PM4.4 | PU4.4 | — | — | — | — |
| | 5 | P4.5 | PM4.5 | PU4.5 | — | — | — | — |
| | 6 | P4.6 | PM4.6 | PU4.6 | — | — | — | — |
| | 7 | P4.7 | PM4.7 | PU4.7 | — | — | — | |

**Remark** —: Not provided.

**Table 4-28. Port Configuration of Group E Products (100-pin products) (2/4)**

| Port Name | Port Bit | Output Latch | I/O Mode Control | Pull-up Control | Input Type Control | Output Type Control | Operating Mode Control | Input Threshold Control |
|---|---|---|---|---|---|---|---|---|
| PORT5 | 0 | P5.0 | PM5.0 | PU5.0 | — | — | — | PITHL5.0 |
| | 1 | P5.1 | PM5.1 | PU5.1 | — | — | — | — |
| | 2 | P5.2 | PM5.2 | PU5.2 | — | — | — | PITHL5.2 |
| | 3 | P5.3 | PM5.3 | PU5.3 | — | — | — | PITHL5.3 |
| | 4 | P5.4 | PM5.4 | PU5.4 | PIM5.4 | — | — | PITHL5.4 |
| | 5 | P5.5 | PM5.5 | PU5.5 | — | — | — | — |
| | 6 | P5.6 | PM5.6 | PU5.6 | — | — | — | — |
| | 7 | P5.7 | PM5.7 | PU5.7 | — | — | — | — |
| PORT6 | 0 | P6.0 | PM6.0 | PU6.0 | — | POM6.0 | — | PITHL6.0 |
| | 1 | P6.1 | PM6.1 | PU6.1 | — | POM6.1 | — | PITHL6.1 |
| | 2 | P6.2 | PM6.2 | PU6.2 | PIM6.2 | POM6.2 | — | PITHL6.2 |
| | 3 | P6.3 | PM6.3 | PU6.3 | PIM6.3 | POM6.3 | — | PITHL6.3 |
| | 4 | P6.4 | PM6.4 | PU6.4 | — | — | — | — |
| | 5 | P6.5 | PM6.5 | PU6.5 | — | — | — | — |
| | 6 | P6.6 | PM6.6 | PU6.6 | — | — | — | — |
| | 7 | P6.7 | PM6.7 | PU6.7 | — | — | — | — |
| PORT7 | 0 | P7.0 | PM7.0 | PU7.0 | PIM7.0 | POM7.0 | PMC7.0 | PITHL7.0 |
| | 1 | P7.1 | PM7.1 | PU7.1 | PIM7.1 | POM7.1 | PMC7.1 | PITHL7.1 |
| | 2 | P7.2 | PM7.2 | PU7.2 | — | POM7.2 | PMC7.2 | — |
| | 3 | P7.3 | PM7.3 | PU7.3 | PIM7.3 | — | PMC7.3 | PITHL7.3 |
| | 4 | P7.4 | PM7.4 | PU7.4 | — | — | PMC7.4 | — |
| | 5 | P7.5 | PM7.5 | PU7.5 | — | — | — | PITHL7.5 |
| | 6 | P7.6 | PM7.6 | PU7.6 | — | — | — | PITHL7.6 |
| | 7 | P7.7 | PM7.7 | PU7.7 | — | — | — | PITHL7.7 |
| PORT8 | 0 | P8.0 | PM8.0 | — | — | — | — | — |
| | 1 | P8.1 | PM8.1 | — | — | — | — | — |
| | 2 | P8.2 | PM8.2 | — | — | — | — | — |
| | 3 | P8.3 | PM8.3 | — | — | — | — | — |
| | 4 | P8.4 | PM8.4 | — | — | — | — | — |
| | 5 | P8.5 | PM8.5 | — | — | — | — | — |
| | 6 | P8.6 | PM8.6 | — | — | — | — | — |
| | 7 | P8.7 | PM8.7 | — | — | — | — | — |
| PORT9 | 0 | P9.0 | PM9.0 | — | — | — | — | — |
| | 1 | P9.1 | PM9.1 | — | — | — | — | — |
| | 2 | P9.2 | PM9.2 | — | — | — | — | — |
| | 3 | P9.3 | PM9.3 | — | — | — | — | — |
| | 4 | P9.4 | PM9.4 | — | — | — | — | — |
| | 5 | P9.5 | PM9.5 | — | — | — | — | — |
| | 6 | P9.6 | PM9.6 | — | — | — | — | — |
| | 7 | P9.7 | PM9.7 | — | — | — | — | — |

**Remark** —: Not provided.

**Table 4-28. Port Configuration of Group E Products (100-pin products) (3/4)**

| Port Name | Port Bit | Output Latch | I/O Mode Control | Pull-up Control | Input Type Control | Output Type Control | Operating Mode Control | Input Threshold Control |
|---|---|---|---|---|---|---|---|---|
| PORT10 | 0 | P10.0 | PM10.0 | — | — | — | — | — |
| | 1 | P10.1 | PM10.1 | — | — | — | — | — |
| | 2 | P10.2 | PM10.2 | — | — | — | — | — |
| | 3 | P10.3 | PM10.3 | — | — | — | — | — |
| | 4 | P10.4 | PM10.4 | — | — | — | — | — |
| | 5 | P10.5 | PM10.5 | — | — | — | — | — |
| | 6 | P10.6 | PM10.6 | PU10.6 | — | — | — | — |
| | 7 | P10.7 | PM10.7 | PU10.7 | — | — | — | PITHL10.7 |
| PORT11 | 0 | — | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |
| PORT12 | 0 | P12.0 | PM12.0 | PU12.0 | — | POM12.0 | PMC12.0 | — |
| | 1 | P12.1 | — | — | — | — | — | — |
| | 2 | P12.2 | — | — | — | — | — | — |
| | 3 | P12.3 | — | — | — | — | — | — |
| | 4 | P12.4 | — | — | — | — | — | — |
| | 5 | P12.5 | PM12.5 | PU12.5 | PIM12.5 | — | PMC12.5 | PITHL12.5 |
| | 6 | P12.6 | PM12.6 | PU12.6 | — | — | — | — |
| | 7 | P12.7 | PM12.7 | PU12.7 | — | — | — | — |
| PORT13 | 0 | P13.0 | — | — | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | P13.7 | — | — | — | — | — | — |
| PORT14 | 0 | P14.0 | PM14.0 | PU14.0 | — | — | — | — |
| | 1 | — | — | — | — | — | — | — |
| | 2 | — | — | — | — | — | — | — |
| | 3 | — | — | — | — | — | — | — |
| | 4 | — | — | — | — | — | — | — |
| | 5 | — | — | — | — | — | — | — |
| | 6 | — | — | — | — | — | — | — |
| | 7 | — | — | — | — | — | — | — |

**Remark** —: Not provided.

**Table 4-28. Port Configuration of Group E Products (100-pin products) (4/4)**

| Port Name | Port Bit | Output Latch | I/O Mode Control | Pull-up Control | Input Type Control | Output Type Control | Operating Mode Control | Input Threshold Control |
|---|---|---|---|---|---|---|---|---|
| PORT15 | 0 | P15.0 | PM15.0 | PU15.0 | — | — | — | PITHL15.0 |
| | 1 | P15.1 | PM15.1 | PU15.1 | — | — | — | — |
| | 2 | P15.2 | PM15.2 | PU15.2 | — | — | — | PITHL15.2 |
| | 3 | P15.3 | PM15.3 | PU15.3 | — | — | — | PITHL15.3 |
| | 4 | P15.4 | PM15.4 | PU15.4 | — | — | — | — |
| | 5 | P15.5 | PM15.5 | PU15.5 | — | — | — | — |
| | 6 | P15.6 | PM15.6 | PU15.6 | — | — | — | — |
| | 7 | P15.7 | PM15.7 | PU15.7 | — | — | — | — |

**Remark** —: Not provided.

### 4.3.1 Port mode registers (PMxx)

These registers specify input or output mode for the port in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

When port pins are used as alternate-function pins, set the port mode register by referencing **4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function**.

**Figure 4-74. Format of Port Mode Register (100-pin products)**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | After reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PM0 | 1 | 1 | 1 | 1 | PM03 | PM02 | PM01 | PM00 | FFF20H | FFH | R/W |
| PM1 | PM17 | PM16 | PM15 | PM14 | PM13 | PM12 | PM11 | PM10 | FFF21H | FFH | R/W |
| PM3 | 1 | 1 | 1 | PM34 | PM33 | PM32 | PM31 | PM30 | FFF23H | FFH | R/W |
| PM4 | PM47 | PM46 | PM45 | PM44 | PM43 | PM42 | PM41 | PM40 | FFF24H | FFH | R/W |
| PM5 | PM57 | PM56 | PM55 | PM54 | PM53 | PM52 | PM51 | PM50 | FFF25H | FFH | R/W |
| PM6 | PM67 | PM66 | PM65 | PM64 | PM63 | PM62 | PM61 | PM60 | FFF26H | FFH | R/W |
| PM7 | PM77 | PM76 | PM75 | PM74 | PM73 | PM72 | PM71 | PM70 | FFF27H | FFH | R/W |
| PM8 | PM87 | PM86 | PM85 | PM84 | PM83 | PM82 | PM81 | PM80 | FFF28H | FFH | R/W |
| PM9 | PM97 | PM96 | PM95 | PM94 | PM93 | PM92 | PM91 | PM90 | FFF29H | FFH | R/W |
| PM10 | PM107 | PM106 | PM105 | PM104 | PM103 | PM102 | PM101 | PM100 | FFF2AH | FFH | R/W |
| PM12 | PM127 | PM126 | PM125 | 1 | 1 | 1 | 1 | PM120 | FFF2CH | FFH | R/W |
| PM14 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | PM140 | FFF2EH | FFH | R/W |
| PM15 | PM157 | PM156 | PM155 | PM154 | PM153 | PM152 | PM151 | PM150 | FFF2FH | FFH | R/W |

| PMmn | Pmn pin I/O mode selection<br>(m = 0, 1, 3 to 10, 12, 14, 15; n = 0 to 7) |
|---|---|
| 0 | Output mode (output buffer on) |
| 1 | Input mode (output buffer off) |

**Caution Be sure to set bits for pins that are not present to their initial values.**

### 4.3.2 Port registers (Pxx)

These registers set the output latch value of a port.

If the data is read in the input mode, the pin level is read. If it is read in the output mode, the output latch value is read **Note**.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Even when PMxx is set to 0 (output mode), the pin level can be read from Pxx by setting PMS.0 (port mode select) to 1.

Reset signal generation clears these registers to 00H.

**Note** When P33, P34, P70 to P74, P80 to P87, P90 to P97, P100 to P105, P120, and P125 are set up as analog inputs of the A/D converter, P80 is set up as D/A converter output, or P81 to P85 are set up as analog inputs of the comparator, if a port is read while in the input mode, 0 is always returned, not the pin level.

**Figure 4-75. Format of Port Register (100-pin products)**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | After reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| P0 | 0 | 0 | 0 | 0 | P03 | P02 | P01 | P00 | FFF00H | 00H (output latch) | R/W |
| P1 | P17 | P16 | P15 | P14 | P13 | P12 | P11 | P10 | FFF01H | 00H (output latch) | R/W |
| P3 | 0 | 0 | 0 | P34 | P33 | P32 | P31 | P30 | FFF03H | 00H (output latch) | R/W |
| P4 | P47 | P46 | P45 | P44 | P43 | P42 | P41 | P40 | FFF04H | 00H (output latch) | R/W |
| P5 | P57 | P56 | P55 | P54 | P53 | P52 | P51 | P50 | FFF05H | 00H (output latch) | R/W |
| P6 | P67 | P66 | P65 | P64 | P63 | P62 | P61 | P60 | FFF06H | 00H (output latch) | R/W |
| P7 | P77 | P76 | P75 | P74 | P73 | P72 | P71 | P70 | FFF07H | 00H (output latch) | R/W |
| P8 | P87 | P86 | P85 | P84 | P83 | P82 | P81 | P80 | FFF08H | 00H (output latch) | R/W |
| P9 | P97 | P96 | P95 | P94 | P93 | P92 | P91 | P90 | FFF09H | 00H (output latch) | R/W |
| P10 | P107 | P106 | P105 | P104 | P103 | P102 | P101 | P100 | FFF0AH | 00H (output latch) | R/W |
| P12 | P127 | P126 | P125 | P124 | P123 | P122 | P121 | P120 | FFF0CH | Undefined | R/W Note 1 |
| P13 | P137 | 0 | 0 | 0 | 0 | 0 | 0 | P130 | FFF0DH | Undefined Note 2 | R/W Note 1 |
| P14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | P140 | FFF0EH | 00H (output latch) | R/W |
| P15 | P157 | P156 | P155 | P154 | P153 | P152 | P151 | P150 | FFF0FH | 00H (output latch) | R/W |

| Pmn | m = 0, 1, 3 to 10, 12 to 15; n = 0 to 7 | |
|---|---|---|
| | Output data control (in output mode) | Input data read (in input mode) |
| 0 | Output 0 | Input low level |
| 1 | Output 1 | Input high level |

<R>

**Notes 1.** P121 to P124 and P137 are read-only.

**2.** P130 bit depends on the setting of User Option Byte (000C2H/020C2H).
RESOUTB = 0 (Selects P130 as the RESOUT pin): P130 = 1
RESOUTB = 1 (Selects P130 as a general port pin): P130 = 0

**Caution Be sure to set bits for pins that are not present to their initial values.**

### 4.3.3 Pull-up resistor option registers (PUxx)

These registers specify whether the on-chip pull-up resistors are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode (PMmn = 1 and POMmn = 0) for the pins to which the use of an on-chip pull-up resistor has been specified in these registers. On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins or analog pins, regardless of the settings of these registers.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H (only PU4 is set to 01H).

**Figure 4-76. Format of Pull-up Resistor Option Register (100-pin products) Note**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | After reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PU0 | 0 | 0 | 0 | 0 | PU03 | PU02 | PU01 | PU00 | F0030H | 00H | R/W |
| PU1 | PU17 | PU16 | PU15 | PU14 | PU13 | PU12 | PU11 | PU10 | F0031H | 00H | R/W |
| PU3 | 0 | 0 | 0 | 0 | 0 | PU32 | PU31 | PU30 | F0033H | 00H | R/W |
| PU4 | PU47 | PU46 | PU45 | PU44 | PU43 | PU42 | PU41 | PU40 | F0034H | 01H | R/W |
| PU5 | PU57 | PU56 | PU55 | PU54 | PU53 | PU52 | PU51 | PU50 | F0035H | 00H | R/W |
| PU6 | PU67 | PU66 | PU65 | PU64 | PU63 | PU62 | PU61 | PU60 | F0036H | 00H | R/W |
| PU7 | PU77 | PU76 | PU75 | PU74 | PU73 | PU72 | PU71 | PU70 | F0037H | 00H | R/W |
| PU9 | PU97 | PU96 | PU95 | PU94 | PU93 | PU92 | 0 | 0 | F0039H | 00H | R/W |
| PU10 | PU107 | PU106 | 0 | 0 | 0 | 0 | 0 | 0 | F003AH | 00H | R/W |
| PU12 | PU127 | PU126 | PU125 | 0 | 0 | 0 | 0 | PU120 | F003CH | 00H | R/W |
| PU14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PU140 | F003EH | 00H | R/W |
| PU15 | PU157 | PU156 | PU155 | PU154 | PU153 | PU152 | PU151 | PU150 | F003FH | 00H | R/W |

| PUmn | Pmn pin on-chip pull-up resistor selection<br>(m = 0, 1, 3 to 7, 9, 10, 12, 14, 15; n = 0 to 7) |
|---|---|
| 0 | On-chip pull-up resistor not connected |
| 1 | On-chip pull-up resistor connected |

**Note** Group E products do not have the PU9 register (products of Groups A to D have the PU9 register).

**Caution Be sure to set bits for pins that are not present to their initial values.**

### 4.3.4 Port input mode registers (PIM1, PIM3, PIM5 to PIM7, PIM12)

These registers set the input buffer of P10, P11, P13, P14, P16, P17, P30, P54, P62, P63, P70, P71, P73, and P125 in 1-bit units.

TTL input buffer can be selected during serial communication with an external device of the different potential.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Figure 4-77. Format of Port Input Mode Register (100-pin products)**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | After reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PIM1 | PIM17 | PIM16 | 0 | PIM14 | PIM13 | 0 | PIM11 | PIM10 | F0041H | 00H | R/W |
| PIM3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PIM30 | F0043H | 00H | R/W |
| PIM5 | 0 | 0 | 0 | PIM54 | 0 | 0 | 0 | 0 | F0045H | 00H | R/W |
| PIM6 | 0 | 0 | 0 | 0 | PIM63 | PIM62 | 0 | 0 | F0046H | 00H | R/W |
| PIM7 | 0 | 0 | 0 | 0 | PIM73 | 0 | PIM71 | PIM70 | F0047H | 00H | R/W |
| PIM12 | 0 | 0 | PIM125 | 0 | 0 | 0 | 0 | 0 | F004CH | 00H | R/W |

| PIMmn | Pmn pin input buffer selection<br>(m = 1, 3, 5 to 7, 12; n = 0 to 7) |
|---|---|
| 0 | Normal input buffer |
| 1 | TTL input buffer |

**Caution Be sure to set bits for pins that are not present to their initial values.**

### 4.3.5 Port output mode registers (POM1, POM6, POM7, POM12)

These registers set the output mode of P10 to P17, P60 to P63, P70 to P72, and P120 in 1-bit units.

N-ch open-drain output ($EV_{DD}$ tolerance) mode can be selected for the SDA00, SDA01, SDA10, and SDA11 pins during serial communication with an external device of the different potential or during simplified IIC communication with an external device of the same potential, and it can be also selected for the SDAA0 and SCLA0 pins during IIC communication.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Figure 4-78. Format of Port Output Mode Register (100-pin products)**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | After reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| POM1 | POM17 | POM16 | POM15 | POM14 | POM13 | POM12 | POM11 | POM10 | F0051H | 00H | R/W |
| POM6 | 0 | 0 | 0 | 0 | POM63 | POM62 | POM61 | POM60 | F0056H | 00H | R/W |
| POM7 | 0 | 0 | 0 | 0 | 0 | POM72 | POM71 | POM70 | F0057H | 00H | R/W |
| POM12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | POM120 | F005CH | 00H | R/W |

| POMmn | Pmn pin output mode selection<br>(m = 1, 6, 7, 12; n = 0 to 7) |
|---|---|
| 0 | Normal output mode |
| 1 | N-ch open-drain output ($EV_{DD}$ tolerance) mode |

**Cautions 1. The on-chip pull-up resistor cannot be used when POMmn is set to 1.**

**2. Be sure to set bits for pins that are not present to their initial values.**

### 4.3.6 Port mode control registers 7, 9, 12 (PMC7, PMC9, PMC12)

These registers set the P70 to P74, P96, P97, P120, and P125 digital I/O or analog input in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to FFH.

**Figure 4-79. Format of Port Mode Control Register (100-pin products)**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | After reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PMC7 | 1 | 1 | 1 | PMC74 Note 1 | PMC73 Note 1 | PMC72 Note 1 | PMC71 Note 1 | PMC70 | F0067H | FFH | R/W |
| PMC9 | PMC97 Note 2 | PMC96 Note 2 | 1 | 1 | 1 | 1 | 1 | 1 | F0069H | FFH | R/W |
| PMC12 | 1 | 1 | PMC125 | 1 | 1 | 1 | 1 | PMC120 | F006CH | FFH | R/W |

| PMCmn | Pmn pin digital I/O or analog input selection (m = 7, 9, 12; n = 0 to 7) |
|---|---|
| 0 | Digital I/O (alternate function other than analog input) |
| 1 | Analog input |

<R> **Notes 1.** Be sure to clear the following bits to 0.

PMC71 to PMC74 bits in the RL78/F14 products with 64 pins and 128 Kbytes to 256 Kbytes of code flash memory.

PMC73 bit in the RL78/F14 products with 48 pins and 128 Kbytes to 256 Kbytes of code flash memory.

**2.** The ADPC and PMC9 registers are used to select the digital I/O or analog input functions for the P96/ANI16 and P97/ANI17 pins and for the P96/ANI26 and P97/ANI27 pins, respectively. For details on pin functions allocated to each product, see **1.5 Pin Configurations**.

<R> **Cautions 1. Be sure to set bits for pins that are not present to their initial values, and see Note 1 for PMC71 to PMC74.**

**2. Set port pins specified as analog input pins to input mode by using port mode register x (PMx).**

### 4.3.7 A/D port configuration register (ADPC)

This register is used to switch the P33/ANI0/AVREFP, P34/ANI1/AVREFM, P80/ANI2/ANO0, P81/ANI3/IVCMP00 to P84/ANI6/IVCMP03, P85/ANI7/IVREF0, and P86/ANI8 to P105/ANI23 pins to digital I/O of port or analog input of A/D converter.

This register is also used to switch the P80/ANI2/ANO0 pins to digital I/O of port or analog output of D/A converter.

This register is also used to switch the P81/ANI3/IVCMP00 to P84/ANI6/IVCMP03 and P85/ANI7/IVREF0 pins to analog input of comparator or digital I/O of port.

The ADPC register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 4-80. Format of A/D Port Configuration Register (ADPC) (100-pin products)**

Address: F0076H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADPC | 0 | 0 | 0 | ADPC4 | ADPC3 | ADPC2 | ADPC1 | ADPC0 |

| ADPC4 to ADPC0 | | | | | ANI23/P105 | ANI22/P104 | ANI21/P103 | ANI20/P102 | ANI19/P101 | ANI18/P100 | ANI17/P97 | ANI16/P96 | ANI15/P95 | ANI14/P94 | ANI13/P93 | ANI12/P92 | ANI11/P91 | ANI10/P90 | ANI9/P87 | ANI8/P86 | ANI7/P85 | ANI6/P84 | ANI5/P83 | ANI4/P82 | ANI3/P81 | ANI2/P80 | ANI1/P34 | ANI0/P33 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 0 | 0 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D |
| 0 | 0 | 0 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A |
| 0 | 0 | 0 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A |
| 0 | 0 | 1 | 0 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A |
| 0 | 0 | 1 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A |
| 0 | 0 | 1 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A |
| 0 | 0 | 1 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A |
| 0 | 1 | 0 | 0 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A |
| 0 | 1 | 0 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A |
| 0 | 1 | 0 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 0 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 0 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 0 | 0 | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 0 | 1 | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 1 | 0 | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 1 | 1 | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 0 | 0 | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 0 | 1 | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 1 | 0 | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 1 | 1 | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 1 | 0 | 0 | 0 | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| Other than above | | | | | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | |

**Cautions**
1. **Set the channel used for A/D conversion to the input mode by using port mode registers 3, 8, 9, 10 (PM3, PM8, PM9, PM10).**
2. **Set the channel used for D/A conversion or comparator to the input mode by using port mode registers 3, 8 (PM3, PM8).**
3. **Do not set the pin set by the ADPC register as digital I/O by the analog input channel specification register (ADS).**
4. **Do not set the pin set by the ADPC register as digital I/O by D/A converter mode register (DAM) as D/A conversion operation enable.**
5. **Do not set the pin set by the ADPC register as digital I/O by the comparator I/O select register (CMPSEL).**

### 4.3.8 Port input threshold control register (PITHL1, PITHL3 to PITHL7, PITHL10, PITHL12, PITHL15)

These registers are used to specify the threshold value of the input buffers for P10, P11, P13, P14, P16, P17, P30, P43, P50, P52 to P54, P60 to P63, P70, P71, P73, P75 to P77, P107, P125, P150, P152, and P153 in 1-bit units.

These registers can set $V_{IL}$ to 0.5 $EV_{DD}$ for the serial communications interface and some external interrupts.

The PITHL1, PITHL3 to PITHL7, PITHL10, PITHL12, and PITHL15 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Figure 4-81. Format of Port Input Threshold Control Register (100-pin products)**

Address: F0021H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PITHL1 | PITHL17 | PITHL16 | 0 | PITHL14 | PITHL13 | 0 | PITHL11 | PITHL10 |

Address: F0023H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PITHL3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PITHL30 |

Address: F0024H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PITHL4 | 0 | 0 | 0 | 0 | PITHL43 | 0 | 0 | 0 |

Address: F0025H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PITHL5 | 0 | 0 | 0 | PITHL54 | PITHL53 | PITHL52 | 0 | PITHL50 |

Address: F0026H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PITHL6 | 0 | 0 | 0 | 0 | PITHL63 | PITHL62 | PITHL61 | PITHL60 |

Address: F0027H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PITHL7 | PITHL77 | PITHL76 | PITHL75 | 0 | PITHL73 | 0 | PITHL71 | PITHL70 |

Address: F002AH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PITHL10 | PITHL107 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Address: F002CH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PITHL12 | 0 | 0 | PITHL125 | 0 | 0 | 0 | 0 | 0 |

Address: F002FH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PITHL15 | 0 | 0 | 0 | 0 | PITHL153 | PITHL152 | 0 | PITHL150 |

| PITHLmn | Selection of the input buffer threshold for Pmn pins (m = 1, 3 to 7, 10, 12, 15; n = 0 to 7) |
|---|---|
| 0 | Schmitt1 input |
| 1 | Schmitt3 input |

| PIMmn | PITHLmn | Selection of the input buffer threshold for Pmn pins<br>(m = 1, 3 to 7, 10, 12, 15; n = 0 to 7) |
|---|---|---|
| 0 | 0 | Schmitt1 input |
| 0 | 1 | Schmitt3 input |
| 1 | 0 | TTL input |
| 1 | 1 | Setting prohibited |

**Caution Be sure to set bits for pins that are not present to their initial values.**

### 4.3.9 Peripheral I/O redirection register 0 (PIOR0)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

PIOR0 enables or disables redirection of the timer array unit functions; that is, it specifies which I/O port is assigned to each input pin of timer array unit 0.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 4-82. Format of Peripheral I/O Redirection Register 0 (PIOR0)**

Address: F0016H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIOR0 | PIOR07 | PIOR06 | PIOR05 | PIOR04 | PIOR03 | PIOR02 | PIOR01 | PIOR00 |

| Bit | Function | 100-pin | | 80-pin | | 64-pin | | 48-pin | |
|---|---|---|---|---|---|---|---|---|---|
| | | Setting value | | Setting value | | Setting value | | Setting value | |
| | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| PIOR07 | TI07 | P120 | P44 | P120 | P44 | P120 | — | P120 | — |
| PIOR06 | TI06 | P14 | P02 | P14 | P02 | P14 | — | P14 | — |
| PIOR05 | TI05 | P15 | P00 | P15 | P00 | P15 | P00 | P15 | P00 |
| PIOR04 | TI04 | P13 | P01 | P13 | P01 | P13 | — | P13 | — |
| PIOR03 | TI03 | P125 | P127 | P125 | — | P125 | — | P125 | — |
| PIOR02 | TI02 | P16 | P67 | P16 | P67 | P16 | — | P16 | — |
| PIOR01 | TI01 | P30 | P126 | P30 | P126 | P30 | — | P30 | — |
| PIOR00 | TI00 | P17 | P66 | P17 | P66 | P17 | — | P17 | — |

**Caution The 32-, 30-, and 20-pin products do not have the PIOR0 register.**

### 4.3.10 Peripheral I/O redirection register 1 (PIOR1)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

PIOR1 enables or disables redirection of the timer array unit functions; that is, it specifies which I/O port is assigned to each output pin of timer array unit 0.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 4-83. Format of Peripheral I/O Redirection Register 1 (PIOR1)**

Address: F0017H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIOR1 | PIOR17 | PIOR16 | PIOR15 | PIOR14 | PIOR13 | PIOR12 | PIOR11 | PIOR10 |

| Bit | Function | 100-pin | | 80-pin | | 64-pin | | 48-pin | |
|---|---|---|---|---|---|---|---|---|---|
| | | Setting value | | Setting value | | Setting value | | Setting value | |
| | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| PIOR17 | TO07 | P120 | P44 | P120 | P44 | P120 | — | P120 | — |
| PIOR16 | TO06 | P14 | P02 | P14 | P02 | P14 | — | P14 | — |
| PIOR15 | TO05 | P15 | P00 | P15 | P00 | P15 | P00 | P15 | P00 |
| PIOR14 | TO04 | P13 | P01 | P13 | P01 | P13 | — | P13 | — |
| PIOR13 | TO03 | P125 | P127 | P125 | — | P125 | — | P125 | — |
| PIOR12 | TO02 | P16 | P67 | P16 | P67 | P16 | — | P16 | — |
| PIOR11 | TO01 | P30 | P126 | P30 | P126 | P30 | — | P30 | — |
| PIOR10 | TO00 | P17 | P66 | P17 | P66 | P17 | — | P17 | — |

**Caution The 32-, 30-, and 20-pin products do not have the PIOR1 register.**

### 4.3.11 Peripheral I/O redirection register 2 (PIOR2)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

PIOR2 enables or disables redirection of the timer array unit functions; that is, it specifies which I/O port is assigned to each input pin of timer array unit 1.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 4-84. Format of Peripheral I/O Redirection Register 2 (PIOR2)**

Address: F0018H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIOR2 | PIOR27 | PIOR26 | PIOR25 | PIOR24 | PIOR23 | PIOR22 | PIOR21 | PIOR20 |

| Bit | Function | 100-pin | | 80-pin | |
|---|---|---|---|---|---|
| | | Setting value | | Setting value | |
| | | 0 | 1 | 0 | 1 |
| PIOR27 | TI17 | P71 | P57 | P71 Note 1 | P57 Note 1 |
| PIOR26 | TI16 | P32 | P65 | P32 Note 1 | P65 Note 1 |
| PIOR25 | TI15 | P70 | P56 | P70 Note 1 | P56 Note 1 |
| PIOR24 | TI14 | P31 | P64 | P31 Note 1 | P64 Note 1 |
| PIOR23 | TI13 | P10 | P55 | P10 Note 2 | P55 Note 2 |
| PIOR22 | TI12 | P11 | P46 | P11 Note 2 | P46 Note 2 |
| PIOR21 | TI11 | P12 | P54 | P12 Note 2 | P54 Note 2 |
| PIOR20 | TI10 | P41 | P45 | P41 Note 2 | P45 Note 2 |

**Notes 1.** Group E products only

**2.** Products of Groups B to E only

**Caution The 64-, 48-, 32-, 30-, and 20-pin products do not have the PIOR2 register.**

### 4.3.12 Peripheral I/O redirection register 3 (PIOR3)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

PIOR3 enables or disables redirection of the timer array unit functions; that is, it specifies which I/O port is assigned to each output pin of timer array unit 1.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 4-85. Format of Peripheral I/O Redirection Register 3 (PIOR3)**

Address: F0019H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIOR3 | PIOR37 | PIOR36 | PIOR35 | PIOR34 | PIOR33 | PIOR32 | PIOR31 | PIOR30 |

| Bit | Function | 100-pin | | 80-pin | |
|---|---|---|---|---|---|
| | | Setting value | | Setting value | |
| | | 0 | 1 | 0 | 1 |
| PIOR37 | TO17 | P71 | P57 | P71 Note 1 | P57 Note 1 |
| PIOR36 | TO16 | P32 | P65 | P32 Note 1 | P65 Note 1 |
| PIOR35 | TO15 | P70 | P56 | P70 Note 1 | P56 Note 1 |
| PIOR34 | TO14 | P31 | P64 | P31 Note 1 | P64 Note 1 |
| PIOR33 | TO13 | P10 | P55 | P10 Note 2 | P55 Note 2 |
| PIOR32 | TO12 | P11 | P46 | P11 Note 2 | P46 Note 2 |
| PIOR31 | TO11 | P12 | P54 | P12 Note 2 | P54 Note 2 |
| PIOR30 | TO10 | P41 | P45 | P41 Note 2 | P45 Note 2 |

**Notes 1.** Group E products only

**2.** Products of Groups B to E only

**Caution The 64-, 48-, 32-, 30-, and 20-pin products do not have the PIOR3 register.**

### 4.3.13 Peripheral I/O redirection register 4 (PIOR4)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

PIOR4 enables or disables redirection of the serial communication functions; that is, it specifies which I/O port is assigned to each serial data I/O pin of CAN, serial data I/O pin of LIN, serial data I/O pin of the serial array unit, clock I/O pin, and slave select input pin.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 4-86. Format of Peripheral I/O Redirection Register 4 (PIOR4)**

Address: F001AH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIOR4 | 0 | PIOR46 | PIOR45 | PIOR44 | PIOR43 | PIOR42 | PIOR41 | PIOR40 |

| Bit | Function | 100-pin | | 80-pin | | 64-pin | | 48-pin | | 32-pin | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Setting value | | Setting value | | Setting value | | Setting value | | Setting value | |
| | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| PIOR46 | CRxD0 | P11 | P73 | P11 Note 1 | P73 Note 1 | P11 Note 1 | P73 Note 1 | P11 Note 1 | P73 Note 1 | P11 Note 2 | — |
| | CTxD0 | P10 | P72 | P10 Note 1 | P72 Note 1 | P10 Note 1 | P72 Note 1 | P10 Note 1 | P72 Note 1 | P10 Note 2 | — |
| PIOR45 | LRxD1 | P11 | P107 | P11 Note 3 | — | P11 Note 3 | — | P11 Note 3 | — | Use prohibited | |
| | LTxD1 | P10 | P106 | P10 Note 3 | — | P10 Note 3 | — | P10 Note 3 | — | | |
| PIOR44 | LRxD0 | P14 | P43 | P14 | P43 | P14 | P43 | P14 | — | P14 | — |
| | LTxD0 | P13 | P42 | P13 | P42 | P13 | P42 | P13 | — | P13 | — |
| PIOR43 | SI11/ SDA11 | P70 | P152 Note 6 | P70 Note 4 | — | P70 Note 4 | — | P70 Note 4 | — | Use prohibited | |
| | SO11 | P72 | P151 | P72 Note 4 | — | P72 Note 4 | — | P72 Note 4 | — | | |
| | SCL11/ $\overline{\text{SCK11}}$ | P71 | P153 Note 6 | P71 Note 4 | — | P71 Note 4 | — | P71 Note 4 | — | | |
| | $\overline{\text{SSI11}}$ | P73 | P150 | P73 Note 4 | — | P73 Note 4 | — | P73 Note 4 | — | | |
| PIOR42 | SI10/ SDA10/ RxD1 | P11 | P75 Note 6 | P11 Note 4 | P75 Note 4, 6 | P11 Note 4 | P75 Note 4, 6 | P11 Note 4 | — | P11 Note 5 | — |
| | SO10/ TxD1 | P12 | P74 | P12 Note 4 | P74 Note 4 | P12 Note 4 | P74 Note 4 | P12 Note 4 | — | P12 Note 5 | — |
| | SCL10/ $\overline{\text{SCK10}}$ | P10 | P76 Note 6 | P10 Note 4 | P76 Note 4, 6 | P10 Note 4 | P76 Note 4, 6 | P10 Note 4 | — | P10 Note 5 | — |
| | $\overline{\text{SSI10}}$ | P54 | P77 | P54 Note 4 | P77 Note 4 | — | P77 Note 4 | — | — | — | — |
| PIOR41 | SI01/ SDA01 | P13 | P53 Note 6 | P13 | P53 Note 6 | P13 | P53 Note 6 | P13 | — | P13 | — |
| | SO01 | P120 | P51 | P120 | P51 | P120 | P51 | P120 | — | P120 | — |
| | SCL01/ $\overline{\text{SCK01}}$ | P14 | P52 Note 6 | P14 | P52 Note 6 | P14 | P52 Note 6 | P14 | — | P14 | — |
| | $\overline{\text{SSI01}}$ | P125 | P50 | P125 | P50 | P125 | P50 | P125 | — | P125 | — |
| PIOR40 | SI00/ SDA00/ RxD0 | P16 | P61 | P16 | P61 | P16 | P61 | P16 | P61 | P16 | P61 |
| | SO00/ TxD0 | P15 | P62 | P15 | P62 | P15 | P62 | P15 | P62 | P15 | P62 |
| | SCL00/ $\overline{\text{SCK00}}$ | P17 | P60 | P17 | P60 | P17 | P60 | P17 | P60 | P17 | P60 |
| | $\overline{\text{SSI00}}$ | P30 | P63 | P30 | P63 | P30 | P63 | P30 | P63 | P30 | P63 |

**Notes 1.** Products of Groups C to E only
**2.** Products of Groups C and D only
**3.** Group E products only
**4.** Products of Groups B to E only
**5.** Products of Groups B to D only
**6.** The simplified IIC function (SDA and SCL) cannot be used when PIOR is 1.

**Cautions 1. The 30- and 20-pin products do not have the PIOR4 register.**
**2. Set the bit the use of which is prohibited to 0.**

### 4.3.14 Peripheral I/O redirection register 5 (PIOR5)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

PIOR5 enables or disables redirection of the external interrupt input and key interrupt input; that is, it specifies which I/O port is assigned to each external interrupt input pin or key interrupt input pin.

This register can be set by an 8-bit memory manipulation instruction.

Bits 7 to 4 and 1 are read-only because no functions are assigned to them. The other bits can be read or written to.

After reset cancellation, this register can always be read or written to.

Any reset source clears this register to 00H.

**Figure 4-87. Format of Peripheral I/O Redirection Register 5 (PIOR5)**

Address: F001BH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIOR5 | 0 | 0 | 0 | 0 | PIOR53 | PIOR52 | 0 | PIOR50 |

| Bit | Function | 100-pin | | 80-pin | | 64-pin | | 48-pin | | 32-pin | | 30-pin | | 20-pin | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Setting value | | Setting value | | Setting value | | Setting value | | Setting value | | Setting value | | Setting value | |
| | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| PIOR53 | INTP3 | P17 | P50 | P17 | P50 | P17 | P50 | P17 | — | P17 | — | P17 | — | P17 | — |
| PIOR52 | INTP2 | P30 | P31 | P30 | P31 | P30 | P31 | P30 | P31 | P30 | — | P30 | — | P30 | — |
| PIOR50 | KR7 | P77 | — | P77 | — | P77 | P96 | — | P92 | — | — | — | P87 | — | — |
| | KR6 | P76 | — | P76 | — | P76 | P95 | — | P91 | — | — | — | P86 | — | — |
| | KR5 | P75 | — | P75 | — | P75 | P94 | — | P90 | — | P85 | — | P85 | — | — |
| | KR4 | P74 | — | P74 | — | P74 | P93 | — | P87 | — | P84 | — | P84 | — | — |
| | KR3 | P73 | — | P73 | — | P73 | P92 | P73 | P86 | — | P83 | — | P83 | — | — |
| | KR2 | P72 | — | P72 | — | P72 | P91 | P72 | P85 | — | P82 | — | P82 | — | — |
| | KR1 | P71 | — | P71 | — | P71 | P90 | P71 | P84 | — | P81 | — | P81 | — | P81 |
| | KR0 | P70 | — | P70 | — | P70 | P87 | P70 | P83 | — | P80 | — | P80 | — | P80 |

### 4.3.15 Peripheral I/O redirection register 6 (PIOR6)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

PIOR6 enables or disables redirection of the SNOOZE status output functions; that is, it specifies which I/O port is assigned to each SNOOZE status output pin.

This register can be set by an 8-bit memory manipulation instruction.

After reset cancellation, this register can always be read or written to.

Any reset source clears this register to 00H.

**Figure 4-88. Format of Peripheral I/O Redirection Register 6 (PIOR6)**

Address: F001CH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIOR6 | PIOR67 | PIOR66 | PIOR65 | PIOR64 | PIOR63 | PIOR62 | PIOR61 | PIOR60 |

| Bit | Function | 100-pin | | 80-pin | |
|---|---|---|---|---|---|
| | | Setting value | | Setting value | |
| | | 0 | 1 | 0 | 1 |
| PIOR67 | SNZOUT7 | P73 | P154 | P73 | — |
| PIOR66 | SNZOUT6 | P72 | P155 | P72 | — |
| PIOR65 | SNZOUT5 | P71 | P156 | P71 | — |
| PIOR64 | SNZOUT4 | P70 | P157 | P70 | — |
| PIOR63 | SNZOUT3 | P12 | P64 | P12 | P64 |
| PIOR62 | SNZOUT2 | P41 | P65 | P41 | P65 |
| PIOR61 | SNZOUT1 | P125 | P56 | P125 | P56 |
| PIOR60 | SNZOUT0 | P30 | P57 | P30 | P57 |

**Caution The 64-, 48-, 32-, 30-, and 20-pin products do not have the PIOR6 register.**

### 4.3.16 Peripheral I/O redirection register 7 (PIOR7)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

PIOR7 enables or disables redirection of the timer RD I/O functions; that is, it specifies which I/O port is assigned to each I/O pin of timer RD0.

This register can be set by an 8-bit memory manipulation instruction.

Bits 7 to 4 and 2 are read-only. The other bits can be read or written to.

After reset cancellation, this register can always be read or written to.

Any reset source clears this register to 00H.

**Figure 4-89. Format of Peripheral I/O Redirection Register 7 (PIOR7)**

Address: F001DH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIOR7 | 0 | 0 | 0 | 0 | PIOR73 | 0 | PIOR71 | PIOR70 |

| Bit | Function | 100-pin | | 80-pin | | 64-pin | | 48-pin | | 32-pin | | 30-pin | | 20-pin | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Setting value | | Setting value | | Setting value | | Setting value | | Setting value | | Setting value | | Setting value | |
| | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| PIOR73 | TRDIOD0 | P120 | P12 | P120 | P12 | P120 | P12 | P120 | P12 | P120 | P12 | P120 | P12 | P120 | — |
| PIOR71 | TRDIOB0 | P125 | P11 | P125 | P11 | P125 | P11 | P125 | P11 | P125 | P11 | P125 | P11 | P125 | — |
| PIOR70 | TRDIOA0/ TRDCLK0 | P13 | P15 | P13 | P15 | P13 | P15 | P13 | P15 | P13 | P15 | P13 | P15 | P13 | P15 |

### 4.3.17 Peripheral I/O redirection register 8 (PIOR8)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

PIOR8 enables or disables redirection of the real-time clock correction clock (1 Hz) output function; that is, it specifies which I/O port is assigned to the real-time clock correction clock (1 Hz) output pin.

This register can be set by an 8-bit memory manipulation instruction.

Bits 7 to 1 are read-only. Bit 0 can be read or written to.

After reset cancellation, this register can always be read or written to.

Any reset source clears this register to 00H.

**Figure 4-90. Format of Peripheral I/O Redirection Register 8 (PIOR8)**

Address: F001EH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIOR8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PIOR80 |

| Bit | Function | 100-pin | |
|---|---|---|---|
| | | Setting value | |
| | | 0 | 1 |
| PIOR87 | — | — | — |
| PIOR86 | — | — | — |
| PIOR85 | — | — | — |
| PIOR84 | — | — | — |
| PIOR83 | — | — | — |
| PIOR82 | — | — | — |
| PIOR81 | — | — | — |
| PIOR80 | RTC1HZ | P15 | P03 |

**Caution The 80-, 64-, 48-, 32-, 30-, and 20-pin products do not have the PIOR8 register.**

### 4.3.18 Port output slew rate register (PSRSEL)

This register is used to select the slew rate for the port output.

It can be set by a 1-bit or 8-bit memory manipulation instruction.

Any reset source clears this register to 00H.

**Caution The slew rate of target pins including the alternate functions is changed.**

**Figure 4-91. Format of Port Output Slew Rate Register (PSRSEL)**

Address: F0220H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PSRSEL | 0 | 0 | PSR140 | PSR14 | PSR120 | PSR30 | PSR12 | PSR10 |

| PSR140 | Control target output port: P140/PCLBUZ0 |
|---|---|
| 0 | Normal slew rate |
| 1 | Special slew rate (slower than normal slew rate) |

**Caution PSR140 is not provided in the 32-, 30-, or 20-pin products.**

| PSR14 | Control target output port: P14/$\overline{\text{SCK01}}$/SCL01/TO06/TRDIOC0 |
|---|---|
| 0 | Normal slew rate |
| 1 | Special slew rate (slower than normal slew rate) |

| PSR120 | Control target output port: P120/SO01/TO07/TRDIOD0 |
|---|---|
| 0 | Normal slew rate |
| 1 | Special slew rate (slower than normal slew rate) |

| PSR30 | Control target output port: P30/TO01/TRDIOD1/SNZOUT0 |
|---|---|
| 0 | Normal slew rate |
| 1 | Special slew rate (slower than normal slew rate) |

| PSR12 | Control target output port: P12/SO10/TO11/(TRDIOD0)/TXD1/SNZOUT3 |
|---|---|
| 0 | Normal slew rate |
| 1 | Special slew rate (slower than normal slew rate) |

**Caution PSR12 is not provided in the 20-pin products.**

| PSR10 | Control target output port: P10/$\overline{\text{SCK10}}$/TO13/TRJO0/SCL10/LTXD1/CTXD0 |
|---|---|
| 0 | Normal slew rate |
| 1 | Special slew rate (slower than normal slew rate) |

**Caution PSR10 is not provided in the 20-pin products.**

### 4.3.19 SNOOZE status output control register 0 (PSNZCNT0)

This register is used to output signals indicating that the SNOOZE mode has been entered through external pins.
It can be set by a 1-bit or 8-bit memory manipulation instruction.
Bits 7, 6, 3, and 2 are read-only because no functions are assigned to them. The other bits can be read or written to.
Any reset source clears this register to 00H.

**Cautions 1. Set the target port pin to the output mode and the output latch to 0 before using the SNOOZE status output function.**

**2. Stop the output from peripheral functions assigned to output-enabled port pins and make the register settings at the time of SNZOUT output.**

**3. Set WUTMMCK0 to 1 at the time of SNZOUT output.**

**Figure 4-92. Format of SNOOZE Status Output Control Register 0 (PSNZCNT0)**

Address: F0222H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PSNZCNT0 | 0 | 0 | SNZACT1 | OUTEN1 | 0 | 0 | SNZACT0 | OUTEN0 |

| SNZACT1 | SNZOUT1 active level |
|---|---|
| 0 | When PIOR61 = 0: Active level of SNOOZE status output to P125 is "H".<br>When PIOR61 = 1: Active level of SNOOZE status output to P56 is "H" (only in 100-pin or 80-pin products). |
| 1 | When PIOR61 = 0: Active level of SNOOZE status output to P125 is "L".<br>When PIOR61 = 1: Active level of SNOOZE status output to P56 is "L" (only in 100-pin or 80-pin products). |

| OUTEN1 | SNZOUT1 enable/disable |
|---|---|
| 0 | When PIOR61 = 0: SNOOZE status output to P125 is disabled.<br>When PIOR61 = 1: SNOOZE status output to P56 is disabled (only in 100-pin or 80-pin products). |
| 1 | When PIOR61 = 0: SNOOZE status output to P125 is enabled.<br>When PIOR61 = 1: SNOOZE status output to P56 is enabled (only in 100-pin or 80-pin products). |

| SNZACT0 | SNZOUT0 active level |
|---|---|
| 0 | When PIOR60 = 0: Active level of SNOOZE status output to P30 is "H".<br>When PIOR60 = 1: Active level of SNOOZE status output to P57 is "H" (only in 100-pin or 80-pin products). |
| 1 | When PIOR60 = 0: Active level of SNOOZE status output to P30 is "L".<br>When PIOR60 = 1: Active level of SNOOZE status output to P57 is "L" (only in 100-pin or 80-pin products). |

| OUTEN0 | SNZOUT0 enable/disable |
|---|---|
| 0 | When PIOR60 = 0: SNOOZE status output to P30 is disabled.<br>When PIOR60 = 1: SNOOZE status output to P57 is disabled (only in 100-pin or 80-pin products). |
| 1 | When PIOR60 = 0: SNOOZE status output to P30 is enabled.<br>When PIOR60 = 1: SNOOZE status output to P57 is enabled (only in 100-pin or 80-pin products). |

### 4.3.20 SNOOZE status output control register 1 (PSNZCNT1)

This register is used to output signals indicating that the SNOOZE mode has been entered through external pins.
It can be set by a 1-bit or 8-bit memory manipulation instruction.
Bits 7, 6, 3, and 2 are read-only because no functions are assigned to them. The other bits can be read or written to.
Any reset source clears this register to 00H.

**Cautions 1. Set the target port pin to the output mode and the output latch to 0 before using the SNOOZE status output control register.**

**2. Stop the output from peripheral functions assigned to output-enabled port pins and make the register settings at the time of SNZOUT output.**

**3. Set WUTMMCK0 to 1 at the time of SNZOUT output.**

**Figure 4-93. Format of SNOOZE Status Output Control Register 1 (PSNZCNT1)**

Address: F0223H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PSNZCNT1 | 0 | 0 | SNZACT3 | OUTEN3 | 0 | 0 | SNZACT2 | OUTEN2 |

| SNZACT3 | SNZOUT3 active level |
|---|---|
| 0 | When PIOR63 = 0: Active level of SNOOZE status output to P12 is "H".<br>When PIOR63 = 1: Active level of SNOOZE status output to P64 is "H" (only in 100-pin or 80-pin products). |
| 1 | When PIOR63 = 0: Active level of SNOOZE status output to P12 is "L".<br>When PIOR63 = 1: Active level of SNOOZE status output to P64 is "L" (only in 100-pin or 80-pin products). |

| OUTEN3 | SNZOUT3 enable/disable |
|---|---|
| 0 | When PIOR63 = 0: SNOOZE status output to P12 is disabled.<br>When PIOR63 = 1: SNOOZE status output to P64 is disabled (only in 100-pin or 80-pin products). |
| 1 | When PIOR63 = 0: SNOOZE status output to P12 is enabled.<br>When PIOR63 = 1: SNOOZE status output to P64 is enabled (only in 100-pin or 80-pin products). |

| SNZACT2 | SNZOUT2 active level |
|---|---|
| 0 | When PIOR62 = 0: Active level of SNOOZE status output to P41 is "H".<br>When PIOR62 = 1: Active level of SNOOZE status output to P65 is "H" (only in 100-pin or 80-pin products). |
| 1 | When PIOR62 = 0: Active level of SNOOZE status output to P41 is "L".<br>When PIOR62 = 1: Active level of SNOOZE status output to P65 is "L" (only in 100-pin or 80-pin products). |

| OUTEN2 | SNZOUT2 enable/disable |
|---|---|
| 0 | When PIOR62 = 0: SNOOZE status output to P41 is disabled.<br>When PIOR62 = 1: SNOOZE status output to P65 is disabled (only in 100-pin or 80-pin products). |
| 1 | When PIOR62 = 0: SNOOZE status output to P41 is enabled.<br>When PIOR62 = 1: SNOOZE status output to P65 is enabled (only in 100-pin or 80-pin products). |

**Caution The 20-pin products do not have the PSNZCNT1 register.**

### 4.3.21 SNOOZE status output control register 2 (PSNZCNT2)

This register is used to output signals indicating that the SNOOZE mode has been entered through external pins.
It can be set by a 1-bit or 8-bit memory manipulation instruction.
Bits 7, 6, 3, and 2 are read-only because no functions are assigned to them. The other bits can be read or written to.
Any reset source clears this register to 00H.

**Cautions 1. Set the target port pin to the output mode and the output latch to 0 before using the SNOOZE status output control register.**

**2. Stop the output from peripheral functions assigned to output-enabled port pins and make the register settings at the time of SNZOUT output.**

**3. Set WUTMMCK0 to 1 at the time of SNZOUT output.**

**Figure 4-94. Format of SNOOZE Status Output Control Register 2 (PSNZCNT2)**

Address: F0224H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PSNZCNT2 | 0 | 0 | SNZACT5 | OUTEN5 | 0 | 0 | SNZACT4 | OUTEN4 |

| SNZACT5 | SNZOUT5 active level |
|---|---|
| 0 | When PIOR65 = 0: Active level of SNOOZE status output to P71 is "H".<br>When PIOR65 = 1: Active level of SNOOZE status output to P156 is "H" (only in 100-pin products). |
| 1 | When PIOR65 = 0: Active level of SNOOZE status output to P71 is "L".<br>When PIOR65 = 1: Active level of SNOOZE status output to P156 is "L" (only in 100-pin products). |

| OUTEN5 | SNZOUT5 enable/disable |
|---|---|
| 0 | When PIOR65 = 0: SNOOZE status output to P71 is disabled.<br>When PIOR65 = 1: SNOOZE status output to P156 is disabled (only in 100-pin products). |
| 1 | When PIOR65 = 0: SNOOZE status output to P71 is enabled.<br>When PIOR65 = 1: SNOOZE status output to P156 is enabled (only in 100-pin products). |

| SNZACT4 | SNZOUT4 active level |
|---|---|
| 0 | When PIOR64 = 0: Active level of SNOOZE status output to P70 is "H".<br>When PIOR64 = 1: Active level of SNOOZE status output to P157 is "H" (only in 100-pin products). |
| 1 | When PIOR64 = 0: Active level of SNOOZE status output to P70 is "L".<br>When PIOR64 = 1: Active level of SNOOZE status output to P157 is "L" (only in 100-pin products). |

| OUTEN4 | SNZOUT4 enable/disable |
|---|---|
| 0 | When PIOR64 = 0: SNOOZE status output to P70 is disabled.<br>When PIOR64 = 1: SNOOZE status output to P157 is disabled (only in 100-pin products). |
| 1 | When PIOR64 = 0: SNOOZE status output to P70 is enabled.<br>When PIOR64 = 1: SNOOZE status output to P157 is enabled (only in 100-pin products). |

**Caution The 20-pin, 30-pin, and 32-pin products do not have the PSNZCNT2 register.**

### 4.3.22 SNOOZE status output control register 3 (PSNZCNT3)

This register is used to output signals indicating that the SNOOZE mode has been entered through external pins.
It can be set by a 1-bit or 8-bit memory manipulation instruction.
Bits 7, 6, 3, and 2 are read-only because no functions are assigned to them. The other bits can be read or written to.
Any reset source clears this register to 00H.

**Cautions 1. Set the target port pin to the output mode and the output latch to 0 before using the SNOOZE status output control register.**

**2. Stop the output from peripheral functions assigned to output-enabled port pins and make the register settings at the time of SNZOUT output.**

**3. Set WUTMMCK0 to 1 at the time of SNZOUT output.**

**Figure 4-95. Format of SNOOZE Status Output Control Register 3 (PSNZCNT3)**

Address: F0225H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PSNZCNT3 | 0 | 0 | SNZACT7 | OUTEN7 | 0 | 0 | SNZACT6 | OUTEN6 |

| SNZACT7 | SNZOUT7 active level |
|---|---|
| 0 | When PIOR67 = 0: Active level of SNOOZE status output to P73 is "H".<br>When PIOR67 = 1: Active level of SNOOZE status output to P154 is "H" (only in 100-pin products). |
| 1 | When PIOR67 = 0: Active level of SNOOZE status output to P73 is "L".<br>When PIOR67 = 1: Active level of SNOOZE status output to P154 is "L" (only in 100-pin products). |

| OUTEN7 | SNZOUT7 enable/disable |
|---|---|
| 0 | When PIOR67 = 0: SNOOZE status output to P73 is disabled.<br>When PIOR67 = 1: SNOOZE status output to P154 is disabled (only in 100-pin products). |
| 1 | When PIOR67 = 0: SNOOZE status output to P73 is enabled.<br>When PIOR67 = 1: SNOOZE status output to P154 is enabled (only in 100-pin products). |

| SNZACT6 | SNZOUT6 active level |
|---|---|
| 0 | When PIOR66 = 0: Active level of SNOOZE status output to P72 is "H".<br>When PIOR66 = 1: Active level of SNOOZE status output to P155 is "H" (only in 100-pin products). |
| 1 | When PIOR66 = 0: Active level of SNOOZE status output to P72 is "L".<br>When PIOR66 = 1: Active level of SNOOZE status output to P155 is "L" (only in 100-pin products). |

| OUTEN6 | SNZOUT6 enable/disable |
|---|---|
| 0 | When PIOR66 = 0: SNOOZE status output to P72 is disabled.<br>When PIOR66 = 1: SNOOZE status output to P155 is disabled (only in 100-pin products). |
| 1 | When PIOR66 = 0: SNOOZE status output to P72 is enabled.<br>When PIOR66 = 1: SNOOZE status output to P155 is enabled (only in 100-pin products). |

**Caution The 20-pin, 30-pin, and 32-pin products do not have the PSNZCNT3 register.**

### 4.3.23 Port mode select register (PMS)

This register is provided to support IEC60730.

It selects whether to read the output latch value or the pin output level when the port is set to output mode.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Any reset source clears this register to 00H.

**Figure 4-96. Port Mode Select Register (PMS)**

Address: F0077H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PMS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PMS0 |

| PMS0 | Data read from Pmn when PMmn = 0 (m = 0 to 15; n = 0 to 7) |
|---|---|
| 0 | Initial setting. When PMmn = 0 (output mode), the value of Pmn (output latch) is read. |
| 1 | When PMmn = 0 (output mode), the pin level is read. |

| PMmn | PMS0 | Data read from Pmn |
|---|---|---|
| 0 | 0 | Value of the Pmn register (output latch) |
| 0 | 1 | Pin output level |
| 1 | 0 | Pin input level |
| 1 | 1 | Pin input level |

## 4.4 Port Function Operations

Port operations differ depending on whether the input or output mode is set, as shown below.

### 4.4.1 Writing to I/O port

**(1) Output mode**

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

**(2) Input mode**

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change. Accordingly, data can be written in byte units to a port having both input and output pins.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

### 4.4.2 Reading from I/O port

**(1) Output mode**

The output latch contents are read by a transfer instruction. The output latch contents do not change.
When the PMS0 bit in the port mode select register is set to 1, the pin level can be read from Pxx.

**(2) Input mode**

The pin status is read by a transfer instruction. The output latch contents do not change.

### 4.4.3 Operations on I/O port

**(1) Output mode**

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

**(2) Input mode**

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change. Accordingly, data can be written in byte units to a port having both input and output pins.
The data of the output latch is cleared when a reset signal is generated.

### 4.4.4 Connecting to external device with different potential (3 V)

It is possible to connect to an external device with a different potential (3 V) by changing $EV_{DD}$ to accord with the power supply of the connected device. In products in which $EV_{DD}$ cannot be specified independently or if $EV_{DD}$ cannot be changed to accord with the power supply of the connected device for some reason, I/O connection with an external device operating on 3 V when the system is operating on $V_{DD}$ = 4.0 V to 5.5 V is still possible via the serial interface by using ports 1, 6, 7, and 12.

Regarding inputs, CMOS/TTL switching is possible on a bit-by-bit basis by the port input mode registers 1 and 7 (PIM1, PIM7).

Moreover, regarding outputs, different potentials can be supported by switching the output buffer to the N-ch open-drain ($EV_{DD}$ tolerance) by the port output mode registers 1, 6, 7, and 12 (POM1, POM6, POM7, and POM12).

#### (1) Setting procedure when using I/O pins of UART0, UART1, CSI00, CSI01, CSI10, and CSI11 functions

**(a) Use as 3 V input port**

<1> If pull-up is needed, externally pull up the pin to be used (on-chip pull-up resistor cannot be used).

| | |
|---|---|
| In case of UART0: | P16 |
| In case of UART1: | P11 |
| In case of CSI00: | P16, P17 |
| In case of CSI01: | P13, P14 |
| In case of CSI10: | P11, P10 |
| In case of CSI11: | P70, P71 |

<2> After reset release, the port mode is the input mode (Hi-Z).

<3> Set the corresponding bit of the PIM1 and PIM7 registers to 1 to switch to the TTL input buffer.

<4> $V_{IH}/V_{IL}$ operates on 3 V operating voltage.

**(b) Use as 3 V output port**

<1> Pull up externally the pin to be used (on-chip pull-up resistor cannot be used).

In case of UART0: P15
In case of UART1: P12
In case of CSI00: P15, P17
In case of CSI01: P14, P120
In case of CSI10: P10, P12
In case of CSI11: P71, P72

<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM1, POM6, POM7, and POM12 registers to 1 to set the N-ch open-drain output ($EV_{DD}$ tolerance) mode.
<5> Set the output mode by manipulating the PM1, PM6, PM7, and PM12 registers.
At this time, the output data is high level, so the pin is in the Hi-Z state.
<6> Communication is started by setting the serial array unit.

**(2) Setting procedure when using I/O pins of simplified IIC00, IIC01, IIC10, and IIC11 functions**

<1> Externally pull up the pin to be used (on-chip pull-up resistor cannot be used).

In case of simplified IIC00: P16, P17
In case of simplified IIC01: P13, P14
In case of simplified IIC10: P10, P11
In case of simplified IIC11: P70, P71

<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM1 and POM7 registers to 1 to set the N-ch open-drain output ($EV_{DD}$ tolerance) mode.
<5> Set the corresponding bit of the PIM1 and PIM7 registers to 1 to switch to the TTL input buffer.
<6> Set the corresponding bit of the PM1 and PM7 registers to the output mode (data I/O is possible in the output mode).
At this time, the output data is high level, so the pin is in the Hi-Z state.
<7> Enable the operation of the serial array unit and set the mode to the simplified IIC mode.

## 4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function

To use the alternate function of a port pin, set the port mode register, and output latch as shown in Table 4-29.

**Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (1/8)**

| Pin Name | Alternate Function | | PIORXX | POMXX | PMCXX | PMXX | PXX | PIMXX | PITHLXX |
|---|---|---|---|---|---|---|---|---|---|
| | Function Name | I/O | | | | | | | |
| P00 | INTP9 | Input | 0 | – | – | 1 | × | – | – |
| | (TI05) | Input | 1 | – | – | 1 | × | – | – |
| | (TO05) | Output | 1 | – | – | 0 | 0 | – | – |
| P01 | (TI04) | Input | 1 | – | – | 1 | × | – | – |
| | (TO04) | Output | 1 | – | – | 0 | 0 | – | – |
| P02 | (TI06) | Input | 1 | – | – | 1 | × | – | – |
| | (TO06) | Output | 1 | – | – | 0 | 0 | – | – |
| P03 | (RTC1HZ) | Output | 1 | – | – | 0 | 0 | – | – |
| P10 | TI13 | Input | 0 | × | – | 1 | × | 0 | 0/1 |
| | TO13 | Output | 0 | 0 | – | 0 | 0 | × | × |
| | TRJO0 | Output | × | 0 | – | 0 | 0 | × | × |
| | $\overline{\text{SCK10}}$ | Input | 0 | × | – | 1 | × | 0/1 | 0/1 |
| | | Output | 0 | 0/1 | – | 0 | 1 | × | × |
| | SCL10 | Output | 0 | 0/1 | – | 0 | 1 | × | × |
| | LTXD1 | Output | 0 | 0 | – | 0 | 1 | × | × |
| | CTXD0 | Output | 0 | 0 | – | 0 | 1 | × | × |
| P11 | TI12 | Input | 0 | × | – | 1 | × | 0 | 0/1 |
| | SI10 | Input | 0 | × | – | 1 | × | 0/1 | 0/1 |
| | TO12 | Output | 0 | 0 | – | 0 | 0 | × | × |
| | SDA10 | I/O | 0 | 1 | – | 0 | 1 | 0/1 | 0/1 |
| | RXD1 | Input | 0 | × | – | 1 | × | 0/1 | 0/1 |
| | LRXD1 | Input | 0 | × | – | 1 | × | 0 | 0/1 |
| | CRXD0 | Input | 0 | × | – | 1 | × | 0 | 0/1 |
| | (TRDIOB0) | Input | 1 | × | – | 1 | × | 0 | 0/1 |
| | | Output | 1 | 0 | – | 0 | 0 | × | × |

**Remarks 1.** ×: Don't care
PIORxx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch

**2.** The relationship between pins and their alternate functions shown in this table indicates the relationship when a 100-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIORxx, POMxx, PMCxx, PMxx, and Pxx set in the same way.

**3.** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

(The notes are described after the last table.)

**Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (2/8)**

| Pin Name | Alternate Function | | PIORXX | POMXX | PMCXX | PMXX | PXX | PIMXX | PITHLXX |
|---|---|---|---|---|---|---|---|---|---|
| | Function Name | I/O | | | | | | | |
| P12 | TI11 | Input | 0 | × | – | 1 | × | – | – |
| | INTP5 | Input | × | × | – | 1 | × | – | – |
| | TO11 | Output | 0 | 0 | – | 0 | 0 | – | – |
| | SO10 | Output | 0 | 0/1 | – | 0 | 1 | – | – |
| | TXD1 | Output | 0 | 0/1 | – | 0 | 1 | – | – |
| | SNZOUT3 | Output | 0 | 0 | – | 0 | 0 | – | – |
| | (TRDIOD0) | Input | 1 | × | – | 1 | × | – | – |
| | | Output | 1 | 0 | – | 0 | 0 | – | – |
| P13 | TI04 | Input | 0 | × | – | 1 | × | 0 | 0/1 |
| | SI01 | Input | 0 | × | – | 1 | × | 0/1 | 0/1 |
| | TRDIOA0 | Input | 0 | × | – | 1 | × | 0 | 0/1 |
| | | Output | 0 | 0 | – | 0 | 0 | × | × |
| | TRDCLK0 | Input | 0 | × | – | 1 | × | 0 | 0/1 |
| | TO04 | Output | 0 | 0 | – | 0 | 0 | × | × |
| | SDA01 | I/O | 0 | 1 | – | 0 | 1 | 0/1 | 0/1 |
| | LTXD0 | Output | 0 | 0 | – | 0 | 1 | × | × |
| P14 | TI06 | Input | 0 | × | – | 1 | × | 0 | 0/1 |
| | TO06 | Output | 0 | 0 | – | 0 | 0 | × | × |
| | TRDIOC0 | Input | × | × | – | 1 | × | 0 | 0/1 |
| | | Output | × | 0 | – | 0 | 0 | × | × |
| | $\overline{\text{SCK01}}$ | Input | 0 | × | – | 1 | × | 0/1 | 0/1 |
| | | Output | 0 | 0/1 | – | 0 | 1 | × | × |
| | SCL01 | Output | 0 | 0/1 | – | 0 | 1 | × | × |
| | LRXD0 | Input | 0 | × | – | 1 | × | 0 | 0/1 |
| P15 | TI05 | Input | 0 | × | – | 1 | × | – | – |
| | TO05 | Output | 0 | 0 | – | 0 | 0 | – | – |
| | TRDIOA1 | Input | × | × | – | 1 | × | – | – |
| | | Output | × | 0 | – | 0 | 0 | – | – |
| | SO00 | Output | 0 | 0/1 | – | 0 | 1 | – | – |
| | TXD0 | Output | 0 | 0/1 | – | 0 | 1 | – | – |
| | RTC1HZ | Output | 0 | 0 | – | 0 | 0 | – | – |
| | (TRDIOA0) | Input | 1 | × | – | 1 | × | – | – |
| | | Output | 1 | 0 | – | 0 | 0 | – | – |
| | (TRDCLK0) | Input | 1 | × | – | 1 | × | – | – |

**Remarks 1.** ×: Don't care
PIORxx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch

**2.** The relationship between pins and their alternate functions shown in this table indicates the relationship when a 100-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIORxx, POMxx, PMCxx, PMxx, and Pxx set in the same way.

**3.** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

(The notes are described after the last table.)

**Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (3/8)**

| Pin Name | Alternate Function | | PIORXX | POMXX | PMCXX | PMXX | PXX | PIMXX | PITHLXX |
|---|---|---|---|---|---|---|---|---|---|
| | Function Name | I/O | | | | | | | |
| P16 | TI02 | Input | 0 | × | – | 1 | × | 0 | 0/1 |
| | SI00 | Input | 0 | × | – | 1 | × | 0/1 | 0/1 |
| | TRDIOC1 | Input | × | × | – | 1 | × | 0 | 0/1 |
| | | Output | × | 0 | – | 0 | 0 | × | × |
| | TO02 | Output | 0 | 0 | – | 0 | 0 | × | × |
| | SDA00 | I/O | 0 | 1 | – | 0 | 1 | 0/1 | 0/1 |
| | RXD0 | Input | 0 | × | – | 1 | × | 0/1 | 0/1 |
| P17 | TI00 | Input | 0 | × | – | 1 | × | 0 | 0/1 |
| | INTP3 | Input | 0 | × | – | 1 | × | 0 | 0/1 |
| | TRDIOB1 | Input | × | × | – | 1 | × | 0 | 0/1 |
| | | Output | × | 0 | – | 0 | 0 | × | × |
| | $\overline{SCK00}$ | Input | 0 | × | – | 1 | × | 0/1 | 0/1 |
| | | Output | 0 | 0/1 | | 0 | 1 | × | × |
| | SCL00 | Output | 0 | 0/1 | – | 0 | 1 | × | × |
| | TO00 | Output | 0 | 0 | – | 0 | 0 | × | × |
| P30 | TI01 | Input | 0 | – | – | 1 | × | 0 | 0/1 |
| | INTP2 | Input | 0 | – | – | 1 | × | 0 | 0/1 |
| | TRDIOD1 | Input | × | – | – | 1 | × | 0 | 0/1 |
| | | Output | × | – | – | 0 | 0 | × | × |
| | $\overline{SSI00}$ | Input | 0 | – | – | 1 | × | 0/1 | 0/1 |
| | TO01 | Output | 0 | – | – | 0 | 0 | × | × |
| | SNZOUT0 | Output | 0 | – | – | 0 | 0 | × | × |
| P31 | TI14 | Input | 0 | – | – | 1 | × | – | – |
| | TO14 | Output | 0 | – | – | 0 | 0 | – | – |
| | STOPST | Output | × | – | – | 0 | 0 | – | – |
| | (INTP2) | Input | 1 | – | – | 1 | × | – | – |
| P32 | TI16 | Input | 0 | – | – | 1 | × | – | – |
| | INTP7 | Input | 0 | – | – | 1 | × | – | – |
| | TO16 | Output | 0 | – | – | 0 | 0 | – | – |
| P33 | ANI00 | Input | × | – | – | 1 | × | – | – |
| | $AV_{REFP}$ | Input | × | – | – | 1 | × | – | – |
| P34 | ANI01 | Input | × | – | – | 1 | × | – | – |
| | $AV_{REFM}$ | Input | × | – | – | 1 | × | – | – |

**Remarks 1.** ×: Don't care
PIORxx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch

**2.** The relationship between pins and their alternate functions shown in this table indicates the relationship when a 100-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIORxx, POMxx, PMCxx, PMxx, and Pxx set in the same way.

**3.** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

(The notes are described after the last table.)

**Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (4/8)**

| Pin Name | Alternate Function | | PIORXX | POMXX | PMCXX | PMXX | PXX | PIMXX | PITHLXX |
|---|---|---|---|---|---|---|---|---|---|
| | Function Name | I/O | | | | | | | |
| P40 | TOOL0 | I/O | × | – | – | × | × | – | – |
| P41 | TI10 | Input | 0 | – | – | 1 | × | – | – |
| | TO10 | Output | 0 | – | – | 0 | 0 | – | – |
| | TRJIO0 | Input | × | – | – | 1 | × | – | – |
| | | Output | × | – | – | 0 | 0 | – | – |
| | VCOUT0 | Output | × | – | – | 0 | 0 | – | – |
| | SNZOUT2 | Output | 0 | – | – | 0 | 0 | – | – |
| P42 | (LTXD0) | Output | 1 | – | – | 0 | 1 | – | – |
| P43 | (LRXD0) | Input | 1 | – | – | 1 | × | – | 0/1 |
| P44 | (TI07) | Input | 1 | – | – | 1 | × | – | – |
| | (TO07) | Output | 1 | – | – | 0 | 0 | – | – |
| P45 | (TI10) | Input | 1 | – | – | 1 | × | – | – |
| | (TO10) | Output | 1 | – | – | 0 | 0 | – | – |
| P46 | (TI12) | Input | 1 | – | – | 1 | × | – | – |
| | (TO12) | Output | 1 | – | – | 0 | 0 | – | – |
| P47 | INTP13 | Input | × | – | – | 1 | × | – | – |
| P50 | ($\overline{\text{SSI01}}$) | Input | 1 | – | – | 1 | × | – | 0/1 |
| | (INTP3) | Input | 1 | – | – | 1 | × | – | 0/1 |
| P51 | INTP11 | Input | × | – | – | 1 | × | – | – |
| | (SO01) | Output | 1 | – | – | 0 | 1 | – | – |
| P52 | ($\overline{\text{SCK01}}$) | Input | 1 | – | – | 1 | × | – | 0/1 |
| | | Output | 1 | – | – | 0 | 1 | – | × |
| | (STOPST)[Note] | Output | × | – | – | 0 | 0 | – | 0/1 |
| P53 | INTP10 | Input | × | – | – | 1 | × | – | 0/1 |
| | (SI01) | Input | 1 | – | – | 1 | × | – | 0/1 |
| P54 | $\overline{\text{SSI10}}$ | Input | × | – | – | 1 | × | 0/1 | 0/1 |
| | (TI11) | Input | 1 | – | – | 1 | × | 0 | 0/1 |
| | (TO11) | Output | 1 | – | – | 0 | 0 | × | × |
| P55 | (TI13) | Input | 1 | – | – | 1 | × | – | – |
| | (TO13) | Output | 1 | – | – | 0 | 0 | – | – |
| P56 | (TI15) | Input | 1 | – | – | 1 | × | – | – |
| | (TO15) | Output | 1 | – | – | 0 | 0 | – | – |
| | (SNZOUT1) | Output | 1 | – | – | 0 | 0 | – | – |
| P57 | (TI17) | Input | 1 | – | – | 1 | × | – | – |
| | (TO17) | Output | 1 | – | – | 0 | 0 | – | – |
| | (SNZOUT0) | Output | 1 | – | – | 0 | 0 | – | – |

**Note** The STOPST function can be assigned via settings in the STOP status output control register (STPSTC).

**Remarks 1.** ×: Don't care
PIORxx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch

**2.** The relationship between pins and their alternate functions shown in this table indicates the relationship when a 100-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIORxx, POMxx, PMCxx, PMxx, and Pxx set in the same way.

**3.** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

(The notes are described after the last table.)

**Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (5/8)**

| Pin Name | Alternate Function | | PIORXX | POMXX | PMCXX | PMXX | PXX | PIMXX | PITHLXX |
|---|---|---|---|---|---|---|---|---|---|
| | Function Name | I/O | | | | | | | |
| P60 | ($\overline{SCK00}$) | Input | 1 | × | – | 1 | × | – | 0/1 |
| | | Output | 1 | 0 | – | 0 | 1 | – | × |
| | (SCL00) | Output | 1 | 0/1 | – | 0 | 1 | – | × |
| P61 | (SI00) | Input | 1 | × | – | 1 | × | – | 0/1 |
| | (SDA00) | I/O | 1 | 1 | – | 0 | 1 | – | 0/1 |
| | (RXD0) | Input | 1 | × | – | 1 | × | – | 0/1 |
| P62 | SCLA0 | I/O | × | 1 | – | 0 | 0 | 0/1 | 0/1 |
| | (SO00) | Output | 1 | 0 | – | 0 | 1 | × | × |
| | (TXD0) | Output | 1 | 0 | – | 0 | 1 | × | × |
| P63 | SDAA0 | I/O | × | 1 | – | 0 | 0 | 0/1 | 0/1 |
| | ($\overline{SSI00}$) | Input | 1 | × | – | 1 | × | 0 | 0/1 |
| P64 | (TI14) | Input | 1 | – | – | 1 | × | – | – |
| | (TO14) | Output | 1 | – | – | 0 | 0 | – | – |
| | (SNZOUT3) | Output | 1 | – | – | 0 | 0 | – | – |
| P65 | (TI16) | Input | 1 | – | – | 1 | × | – | – |
| | (TO16) | Output | 1 | – | – | 0 | 0 | – | – |
| | (SNZOUT2) | Output | 1 | – | – | 0 | 0 | – | – |
| P66 | (TI00) | Input | 1 | – | – | 1 | × | – | – |
| | (TO00) | Output | 1 | – | – | 0 | 0 | – | – |
| P67 | (TI02) | Input | 1 | – | – | 1 | × | – | – |
| | (TO02) | Output | 1 | – | – | 0 | 0 | – | – |
| P70 | ANI26 | Input | × | × | 1 | 1 | × | × | × |
| | TI15 | Input | 0 | × | 0 | 1 | × | 0 | 0/1 |
| | SI11 | Input | 0 | × | 0 | 1 | × | 0/1 | 0/1 |
| | INTP8 | Input | × | × | 0 | 1 | × | 0 | 0/1 |
| | TO15 | Output | 0 | 0 | 0 | 0 | 0 | × | × |
| | KR0 | Input | 0 | × | 0 | 1 | × | 0 | 0/1 |
| | SDA11 | I/O | 0 | 1 | 0 | 0 | 1 | 0/1 | 0/1 |
| | SNZOUT4 | Output | 0 | 0 | 0 | 0 | 0 | × | × |

**Remarks 1.** ×: Don't care
PIORxx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch

**2.** The relationship between pins and their alternate functions shown in this table indicates the relationship when a 100-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIORxx, POMxx, PMCxx, PMxx, and Pxx set in the same way.

**3.** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

(The notes are described after the last table.)

**Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (6/8)**

| Pin Name | Alternate Function | | PIORXX | POMXX | PMCXX | PMXX | PXX | PIMXX | PITHLXX |
|---|---|---|---|---|---|---|---|---|---|
| | Function Name | I/O | | | | | | | |
| P71 | ANI27 | Input | × | × | 1 | 1 | × | × | × |
| | TI17 | Input | 0 | × | 0 | 1 | × | 0 | 0/1 |
| | INTP6 | Input | × | × | 0 | 1 | × | 0 | 0/1 |
| | TO17 | Output | 0 | 0 | 0 | 0 | 0 | × | × |
| | KR1 | Input | 0 | × | 0 | 1 | × | 0 | 0/1 |
| | $\overline{\text{SCK11}}$ | Input | 0 | × | 0 | 1 | × | 0/1 | 0/1 |
| | | Output | 0 | 0/1 | 0 | 0 | 1 | × | × |
| | SCL11 | Output | 0 | 0/1 | 0 | 0 | 1 | × | × |
| | SNZOUT5 | Output | 0 | 0 | 0 | 0 | 0 | × | × |
| P72 | ANI28 | Input | × | × | 1 | 1 | × | – | – |
| | KR2 | Input | 0 | × | 0 | 1 | × | – | – |
| | SO11 | Output | 0 | 0/1 | 0 | 0 | 1 | – | – |
| | SNZOUT6 | Output | 0 | 0 | 0 | 0 | 0 | – | – |
| | (CTXD0) | Output | 1 | 0 | 0 | 0 | 1 | – | – |
| P73 | ANI29 | Input | × | – | 1 | 1 | × | × | × |
| | KR3 | Input | 0 | – | 0 | 1 | × | 0 | 0/1 |
| | $\overline{\text{SSI11}}$ | Input | 0 | – | 0 | 1 | × | 0/1 | 0/1 |
| | SNZOUT7 | Output | 0 | – | 0 | 0 | 0 | × | × |
| | (CRXD0) | Input | 1 | – | 0 | 1 | × | 0 | 0/1 |
| P74 | ANI30 | Input | × | – | 1 | 1 | × | – | – |
| | KR4 | Input | 0 | – | 0 | 1 | × | – | – |
| | (SO10) | Output | 1 | – | 0 | 0 | 1 | – | – |
| | (TXD1) | Output | 1 | – | 0 | 0 | 1 | – | – |
| P75 | KR5 | Input | 0 | – | – | 1 | × | – | 0/1 |
| | (SI10) | Input | 1 | – | – | 1 | × | – | 0/1 |
| | (RXD1) | Input | 1 | – | – | 1 | × | – | 0/1 |
| P76 | KR6 | Input | 0 | – | – | 1 | × | – | 0/1 |
| | ($\overline{\text{SCK10}}$) | Input | 1 | – | – | 1 | × | – | 0/1 |
| | | Output | 1 | – | – | 0 | 1 | – | × |
| P77 | INTP12 | Input | × | – | – | 1 | × | – | 0/1 |
| | KR7 | Input | 0 | – | – | 1 | × | – | 0/1 |
| | ($\overline{\text{SSI10}}$) | Input | 1 | – | – | 1 | × | – | 0/1 |
| P80 | ANI02 | Input | × | – | – | 1 | × | – | – |
| | ANO0 | Output | × | – | – | 1 | × | – | – |
| P81 | ANI03 | Input | × | – | – | 1 | × | – | – |
| | IVCMP00 | Input | × | – | – | 1 | × | – | – |

**Remarks 1.** ×: Don't care
PIORxx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch

**2.** The relationship between pins and their alternate functions shown in this table indicates the relationship when a 100-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIORxx, POMxx, PMCxx, PMxx, and Pxx set in the same way.

**3.** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

(The notes are described after the last table.)

**Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (7/8)**

| Pin Name | Alternate Function | | PIORXX | POMXX | PMCXX | PMXX | PXX | PIMXX | PITHLXX |
|---|---|---|---|---|---|---|---|---|---|
| | Function Name | I/O | | | | | | | |
| P82 | ANI04 | Input | × | – | – | 1 | × | – | – |
| | IVCMP01 | Input | × | – | – | 1 | × | – | – |
| P83 | ANI05 | Input | × | – | – | 1 | × | – | – |
| | IVCMP02 | Input | × | – | – | 1 | × | – | – |
| P84 | ANI06 | Input | × | – | – | 1 | × | – | – |
| | IVCMP03 | Input | × | – | – | 1 | × | – | – |
| P85 | ANI07 | Input | × | – | – | 1 | × | – | – |
| | IVREF0 | Input | × | – | – | 1 | × | – | – |
| P86-P87 | ANI08 to ANI09 | Input | × | – | – | 1 | × | – | – |
| P90-P95 | ANI10 to ANI15 | Input | × | – | – | 1 | × | – | – |
| P96-P97 | ANI16 to ANI17 | Input | × | – | – | 1 | × | – | – |
| | ANI26 to ANI27 Note | Input | × | – | 1 | 1 | × | – | – |
| P100-P105 | ANI18 to ANI23 | Input | × | – | – | 1 | × | – | – |
| P106 | (LTXD1) | Output | 1 | – | – | 0 | 1 | – | – |
| P107 | (LRXD1) | Input | 1 | – | – | 1 | × | – | 0/1 |
| P120 | ANI25 | Input | × | × | 1 | 1 | × | – | – |
| | TI07 | Input | 0 | × | 0 | 1 | × | – | – |
| | INTP4 | Input | × | × | 0 | 1 | × | – | – |
| | TRDIOD0 | Input | 0 | × | 0 | 1 | × | – | – |
| | | Output | 0 | 0 | 0 | 0 | 0 | – | – |
| | SO01 | Output | 0 | 0/1 | 0 | 0 | 1 | – | – |
| | TO07 | Output | 0 | 0 | 0 | 0 | 0 | – | – |
| P125 | ANI24 | Input | × | – | 1 | 1 | × | × | × |
| | TI03 | Input | 0 | – | 0 | 1 | × | 0 | 0/1 |
| | INTP1 | Input | × | – | 0 | 1 | × | 0 | 0/1 |
| | TO03 | Output | 0 | – | 0 | 0 | 0 | × | × |
| | TRDIOB0 | Input | 0 | – | 0 | 1 | × | 0 | 0/1 |
| | | Output | 0 | – | 0 | 0 | 0 | × | × |
| | $\overline{\text{SSI01}}$ | Input | 0 | – | 0 | 1 | × | 0/1 | 0/1 |
| | SNZOUT1 | Output | 0 | – | 0 | 0 | 0 | × | × |
| P126 | (TI01) | Input | 1 | – | – | 1 | × | – | – |
| | (TO01) | Output | 1 | – | – | 0 | 0 | – | – |
| P127 | (TI03) | Input | 1 | – | – | 1 | × | – | – |
| | (TO03) | Output | 1 | – | – | 0 | 0 | – | – |
| P130 | RESOUT | Output | × | – | – | – | 0 | – | – |
| P137 | INTP0 | Input | × | – | – | – | × | – | – |

**Note** In the products of Groups B to D.

**Remarks 1.** ×: Don't care
PIORxx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch

**2.** The relationship between pins and their alternate functions shown in this table indicates the relationship when a 100-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIORxx, POMxx, PMCxx, PMxx, and Pxx set in the same way.

**3.** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

(The notes are described after the last table.)

**Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (8/8)**

| Pin Name | Alternate Function | | PIORXX | POMXX | PMCXX | PMXX | PXX | PIMXX | PITHLXX |
|---|---|---|---|---|---|---|---|---|---|
| | Function Name | I/O | | | | | | | |
| P140 | PCLBUZ0 | Output | × | – | – | 0 | 0 | – | – |
| P150 | ($\overline{SSI11}$) | Input | 1 | – | – | 1 | × | – | 0/1 |
| P151 | (SO11) | Output | 1 | – | – | 0 | 1 | – | × |
| P152 | (SI11) | Input | 1 | – | – | 1 | × | – | 0/1 |
| P153 | ($\overline{SCK11}$) | Input | 1 | – | – | 1 | × | – | 0/1 |
| | | Output | 1 | – | – | 0 | 1 | – | × |
| P154 | (SNZOUT7) | Output | 1 | – | – | 0 | 0 | – | × |
| P155 | (SNZOUT6) | Output | 1 | – | – | 0 | 0 | – | × |
| P156 | (SNZOUT5) | Output | 1 | – | – | 0 | 0 | – | × |
| P157 | (SNZOUT4) | Output | 1 | – | – | 0 | 0 | – | × |

**Remarks 1.** ×: Don't care
PIORxx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch

**2.** The relationship between pins and their alternate functions shown in this table indicates the relationship when a 100-pin product is used. In other products, alternate functions might be assigned to different pins, but even in this case, the PIORxx, POMxx, PMCxx, PMxx, and Pxx set in the same way.

**3.** Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

(The notes are described after the last table.)

## 4.6 Cautions When Using Port Function

### 4.6.1 Cautions on 1-bit manipulation instruction for port register n (Pn)

When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the output latch value of an input port that is not subject to manipulation may be written in addition to the targeted bit.

Therefore, it is recommended to rewrite the output latch when switching a port from input mode to output mode.

<Example> When P10 is an output port, P11 to P17 are input ports (all pin statuses are high level), and the port latch value of port 1 is 00H, if the output of output port P10 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 1 is FFH.

Explanation: The targets of writing to and reading from the Pn register of a port whose PMnm bit is 1 are the output latch and pin status, respectively.

A 1-bit manipulation instruction is executed in the following order in the RL78/F13 and RL78/F14.

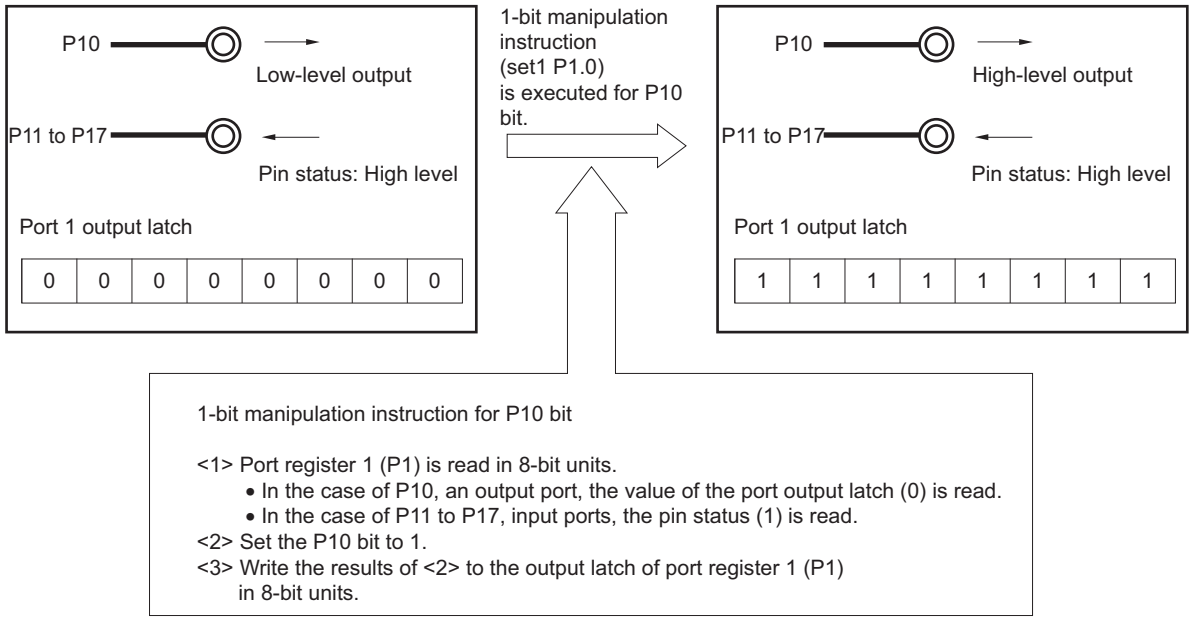
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P17, which are input ports, are read. If the pin statuses of P11 to P17 are high level at this time, the read value is FEH.

The value is changed to FFH by the manipulation in <2>.

FFH is written to the output latch by the manipulation in <3>.

**Figure 4-97. Bit Manipulation Instruction (P10)**



### 4.6.2 Notes on specifying the pin settings

For an output pin to which multiple functions are assigned, the output of the unused alternate functions must be set to its initial state so as to prevent conflicting outputs. This also applies to the functions assigned by using the peripheral I/O redirection register (PIOR). For details about the alternate function output, see **4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function**.

No specific setting is required for input pins because the output of their alternate functions is disabled (the buffer output is Hi-Z).

Disabling the unused functions, including blocks that are only used for input or do not have I/O, is recommended for lower power consumption.

# CHAPTER 5 CLOCK GENERATOR

Use the clock generator within a range that satisfies the values stipulated in **CHAPTER 34** to **CHAPTER 36 ELECTRICAL SPECIFICATIONS**.

The presence or absence of connecting resonator pin for subsystem clock and external clock input pin for subsystem clock depends on the product.

| | 20-, 30-, and 32-pin products | 48-, 64-, 80-, and 100-pin products |
|---|---|---|
| X1, X2 pins | √ | √ |
| EXCLK pin | √ | √ |
| XT1, XT2 pins | – | √ |
| EXCLKS pin | – | √ |

**Cautions 1. The 20-, 30-, and 32-pin products don't have the subsystem clock.**

**2. Do not use the XT1 and XT2 pin functions in grade-Y products.**

## 5.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.

The following three kinds of system clocks and clock oscillators are selectable.

**(1) Main system clock**

**<1> X1 oscillator**

This circuit oscillates a clock of $f_X$ = 1 to 20 MHz by connecting a resonator to X1 and X2.

Oscillation can be stopped by executing the STOP instruction or setting of the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

**<2> High-speed on-chip oscillator (High-speed OCO)**

The frequency at which to oscillate can be selected from among $f_{IH}$ = 64, 48, 32, 24, 16, 12, 8, 4, or 1 MHz (TYP.) by using the user option byte (000C2H/020C2H). When 64 MHz or 48 MHz is selected as $f_{IH}$, $f_{CLK}$ is set to 32 MHz or 24 MHz, respectively, after a reset release. The CPU always starts operating with this high-speed on-chip oscillator clock **Note**. Oscillation can be stopped by executing the STOP instruction or setting of the HIOSTOP bit (bit 0 of the CSC register).

The frequency set by using the user option byte can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV). For the frequency, see **Figure 5-15 Format of High-speed on-chip oscillator frequency select register (HOCODIV)**.

An external main system clock ($f_{EX}$ = 1 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of the MSTOP bit.

As the high-speed system clock, an X1 clock or external main system clock can be selected by setting of the OSCSEL bit (bit 6 of the clock operation mode control register (CMC)) and the EXCLK bit (bit 7 of the clock operation mode control register (CMC)).

As the main system clock, a high-speed system clock (X1 clock or external main system clock) or high-speed on-chip oscillator clock can be selected by setting of the MCM0 bit (bit 4 of the system clock control register (CKC)).

**Note** When selecting 64 MHz or 48 MHz, the selected clock ($f_{IH}$) is supplied to timer RD.
When supplying 64 MHz or 48 MHz to timer RD, set $f_{CLK}$ to $f_{IH}$.

**(2) PLL clock**

This clock oscillates the clock whose $f_{PLL}$ is 24 MHz, 32 MHz, or 64 MHz by oscillating the main system clock at 4 MHz or 8 MHz and multiplying by 3, 4, 6, or 8 times. When setting $f_{PLL}$ to 64 MHz or 48 MHz, the division of $f_{CLK}$ should be set to 32 MHz or 24 MHz by the MDIV2 to MDIV0 bits in the $f_{MP}$ clock division register (MDIV). Oscillation can be stopped by setting the PLLON bit (bit 0 of the PLLCTL register). Before entering STOP mode, the PLLON bit should be cleared to 0 (Stops PLL operation).

**Remarks 1.** The PLL input clock frequency can be set to 4 MHz or 8 MHz. When setting the high-speed on-chip oscillator clock as the PLL input clock, the on-chip oscillator clock can be set to 4 MHz or 8 MHz depending on the setting of bits 4 to 0 (FRQSEL4 to FRQSEL0) of the user option byte (000C2H/020C2H). For details of the user option byte, see **CHAPTER 29 OPTION BYTE**.

**2.** Set the multiplier of the PLL clock by bits 1 (PLLMUL) and 4 (PLLDIV0) in the PLL control register (PLLCTL).

**(3) Subsystem clock**

- **XT1 clock oscillator**

This circuit oscillates a clock of $f_{XT}$ = 32.768 kHz by connecting a 32.768 kHz resonator to XT1 and XT2. Oscillation can be stopped by setting the XTSTOP bit (bit 6 of the clock operation status control register (CSC)).

An external subsystem clock ($f_{EXS}$ = 32.768 kHz) can also be supplied from the EXCLKS/XT2/P124 pin. An external subsystem clock input can be disabled by setting the XTSTOP bit (bit 6 of the clock operation status control register (CSC)).

As the subsystem clock, an XT1 clock or external subsystem clock can be selected by setting of the OSCSELS bit (bit 4 of the clock operation mode control register (CMC)), the EXCLKS bit (bit 5 of the clock operation mode control register (CMC)), and the SELLOSC bit (bit 0 of the clock select register (CKSEL)).

**(4) Low-speed on-chip oscillator (Low-speed OCO)**

This circuit oscillates a clock of $f_{IL}$ = 15 kHz (TYP.).

The low-speed on-chip oscillator clock can be used as the CPU/peripheral hardware clock.

Only the following hardware circuits operate by the low-speed on-chip oscillator clock.

- Clock monitor ($f_{IL}$)
- Timer RJ ($f_{IL}$ and $f_{SL}$)
- Timer RD ($f_{SL}$)
- Clock output/buzzer output control circuit ($f_{SL}$)

This circuit operates when at least bit 4 in the operation speed mode control register (OSMC) or bit 6 in the clock select register (SELLOSC) is 1. When stopping the oscillation of the low-speed on-chip oscillator, set the WUTMMCK0 and SELLOSC bits to 0.

As the main/PLL select clock ($f_{MP}$), a main system clock ($f_{MAIN}$) or PLL clock ($f_{PLL}$) can be selected by setting of the SELPLL bit (bit 2 of the PLL control register (PLLCTL)).

As the subsystem/low-speed on-chip oscillator select clock ($f_{SL}$), a subsystem clock ($f_{SUB}$) or low-speed on-chip oscillator ($f_{IL}$) can be selected by setting of the CKSEL bit (bit 0 of the clock select register (CKSEL)).

**Remark** $f_X$: X1 clock oscillation frequency
$f_{IH}$: High-speed on-chip oscillator clock frequency (64 MHz max.) **Note**
$f_{EX}$: External main system clock frequency
$f_{XT}$: XT1 clock oscillation frequency
$f_{EXS}$: External subsystem clock frequency
$f_{IL}$: Low-speed on-chip oscillator clock frequency
$f_{SL}$: Subsystem/low-speed on-chip oscillator select clock frequency
$f_{PLL}$: PLL clock frequency (64 MHz max.) Notes 2 and 3
$f_{MP}$: Main/PLL select clock frequency (64 MHz max.)

**Notes 1.** $f_{IH}$ is controlled by hardware so that the MDIV register is set to 01H ($f_{MP}$ = two frequency division) when $f_{IH}$ is set to 64 MHz or 48 MHz after a reset release. When supplying 64 MHz or 48 MHz to timer RD, set $f_{CLK}$ to $f_{IH}$.

**2.** When setting $f_{PLL}$ to 64 MHz or 48 MHz, the division of $f_{MP}$ should be set within the range of 1 MHz to 32 MHz (or 1 MHz to 24 MHz for grade-K and grade-Y products) by the MDIV2 to MDIV0 bits in the $f_{MP}$ clock division register (MDIV). When supplying 64 MHz or 48 MHz to timer RD, set $f_{CLK}$ to $f_{PLL}$.

**3.** When supplying 64 MHz or 48 MHz to timer RD, set the MDIV register to 01H ($f_{MP}$/2 is selected).

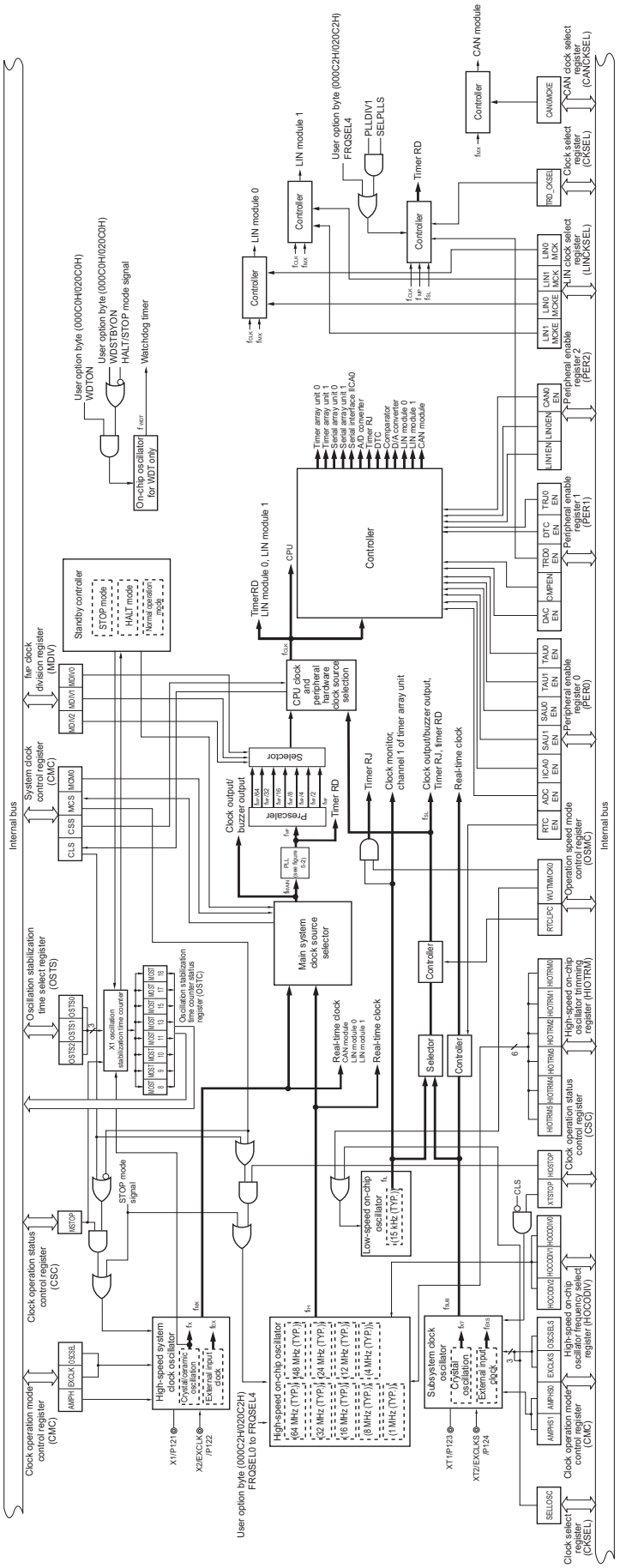
## 5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

**Table 5-1. Configuration of Clock Generator**

| Item | Configuration |
|---|---|
| Control registers | Clock operation mode control register (CMC)<br>System clock control register (CKC)<br>Clock operation status control register (CSC)<br>Oscillation stabilization time counter status register (OSTC)<br>Oscillation stabilization time select register (OSTS)<br>Peripheral enable registers 0, 1, 2 (PER0, PER1, PER2)<br>Operation speed mode control register (OSMC)<br>High-speed on-chip oscillator frequency select register (HOCODIV)<br>High-speed on-chip oscillator trimming register (HIOTRM)<br>CAN clock select register (CANCKSEL)<br>LIN clock select register (LINCKSEL)<br>Clock select register (CKSEL)<br>PLL control register (PLLCTL)<br>PLL status register (PLLSTS)<br>$f_{MP}$ clock division register (MDIV) |
| Oscillators | X1 oscillator<br>XT1 oscillator<br>High-speed on-chip oscillator clock<br>Low-speed on-chip oscillator clock |

**Figure 5-1. Block Diagram of Clock Generator**

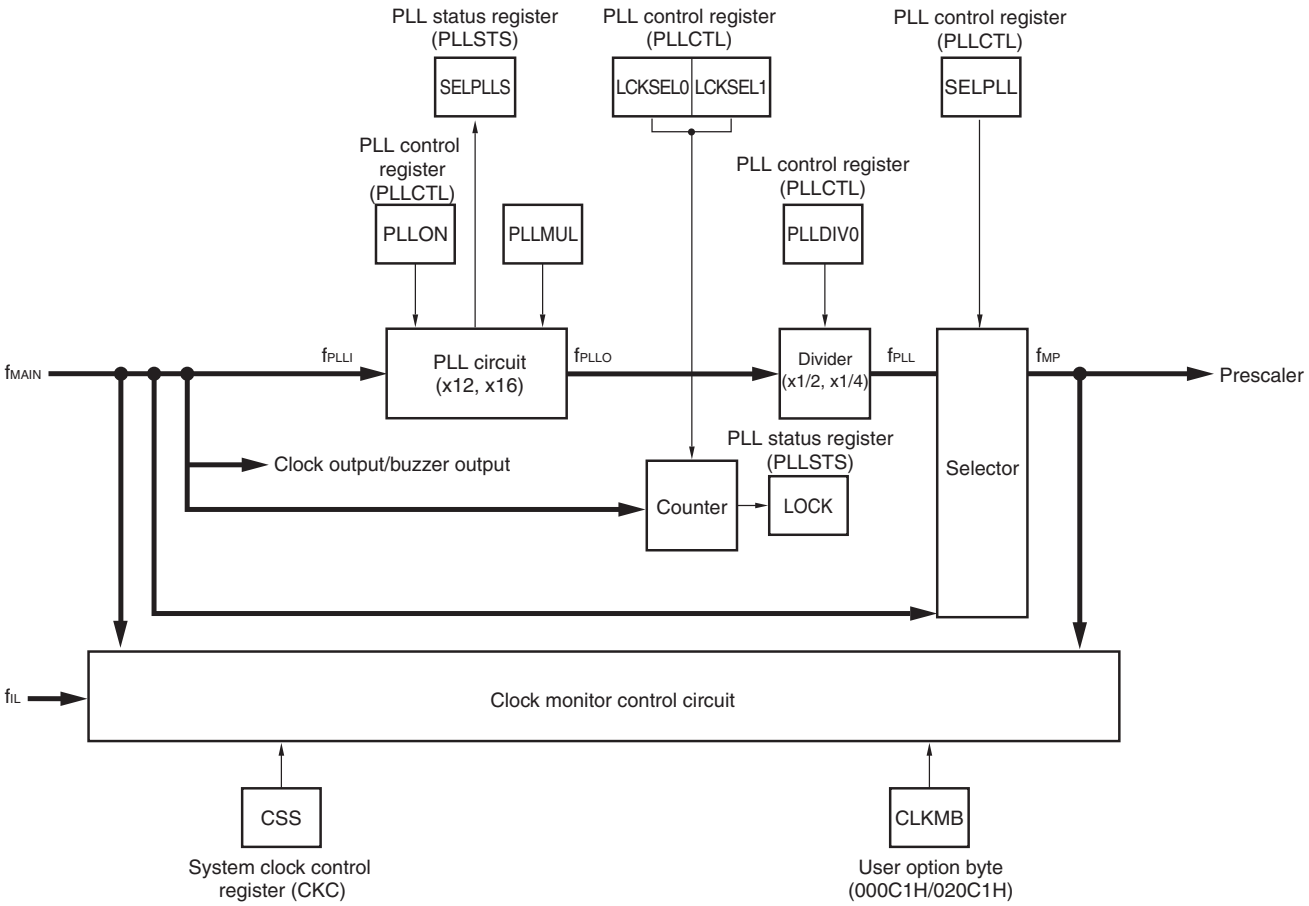



(**Remark** is listed on the next page.)

**Remark** $f_X$: X1 clock oscillation frequency
$f_{IH}$: High-speed on-chip oscillator clock frequency (64 MHz max.) **Note**
$f_{EX}$: External main system clock frequency
$f_{MX}$: High-speed system clock frequency
$f_{MAIN}$: Main system clock frequency
$f_{XT}$: XT1 clock oscillation frequency
$f_{EXS}$: External subsystem clock frequency
$f_{SUB}$: Subsystem clock frequency
$f_{CLK}$: CPU/peripheral hardware clock frequency
$f_{IL}$: Low-speed on-chip oscillator clock frequency
$f_{SL}$: Subsystem/low-speed on-chip oscillator select clock frequency
$f_{MP}$: Main system/PLL select clock frequency
$f_{PLL}$: PLL clock frequency

**Note** $f_{IH}$ is controlled by hardware so that the MDIV register is set to 01H ($f_{MP}$ = two frequency division) when $f_{IH}$ is set to 64 MHz or 48 MHz. When supplying 64 MHz or 48 MHz to timer RD, set $f_{CLK}$ to $f_{IH}$.

**Figure 5-2. Block Diagram of PLL Circuit**




(**Remark** is listed on the next page.)

**Remark** $f_{MAIN}$: Main system clock
$f_{IL}$: Low-speed on-chip oscillator clock
$f_{PLLI}$: PLL input clock
$f_{PLLO}$: PLL output clock
$f_{MP}$: Main system/PLL select clock
$f_{PLL}$: PLL clock

## 5.3 Registers Controlling Clock Generator

The following registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Peripheral enable registers 0, 1, 2 (PER0, PER1, PER2)
- Operation speed mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- High-speed on-chip oscillator trimming register (HIOTRM)
- CAN clock select register (CANCKSEL)
- LIN clock select register (LINCKSEL)
- Clock select register (CKSEL)
- PLL control register (PLLCTL)
- PLL status register (PLLSTS)
- $f_{MP}$ clock division register (MDIV)

### 5.3.1 Clock Operation Mode Control Register (CMC)

This register is used to set the operation mode of the X1/P121, X2/EXCLK/P122, XT1/P123, and XT2/EXCLKS/P124 pins, and to select a gain of the oscillator.

The CMC register can be written only once by an 8-bit memory manipulation instruction after reset release. This register can be read by an 8-bit memory manipulation instruction.

Writing to the CMC register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation sets this register to 00H.

**Figure 5-3. Format of Clock Operation Mode Control Register (CMC)**

Address: FFFA0H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CMC | EXCLK | OSCSEL | EXCLKS | OSCSELS Note 1 | 0 | AMPHS1 | AMPHS0 | AMPH |

| EXCLK | OSCSEL | High-speed system clock pin operation mode | X1/P121 pin | X2/EXCLK/P122 pin |
|---|---|---|---|---|
| 0 | 0 | Input port mode | Input port | |
| 0 | 1 | X1 oscillation mode | Crystal/ceramic resonator connection | |
| 1 | 0 | Input port mode | Input port | |
| 1 | 1 | External clock input mode | Input port | External clock input |

| CKSEL register | CMC register | | Subsystem clock pin operation mode | XT1/P123 pin | XT2/EXCLKS/ P124 pin |
|---|---|---|---|---|---|
| SELLOSC Note 1, 2 | EXCLKS | OSCSELS Note 1 | | | |
| x | 0 | 0 | Input port mode | Input port | |
| 0 | 0 | 1 | XT1 oscillation mode | Crystal/ceramic resonator connection | |
| 1 | 0 | 1 | Input port mode (low-speed on-chip oscillator operation mode) | Input port | |
| x | 1 | 0 | Input port mode | Input port | |
| 0 | 1 | 1 | External clock input mode | Input port | External clock input |
| 1 | 1 | 1 | Input port mode (low-speed on-chip oscillator operation mode) | Input port | |

| AMPHS1 | AMPHS0 | XT1 oscillator oscillation mode selection Note 3 | |
|---|---|---|---|
| 0 | 0 | Low power consumption oscillation (default) | Oscillation margin: Medium |
| 0 | 1 | Normal oscillation | Oscillation margin: high |
| 1 | 0 | Ultra-low power consumption oscillation | Oscillation margin: Low |
| 1 | 1 | Setting prohibited | |

| AMPH | Control of X1 clock oscillation frequency |
|---|---|
| 0 | 1 MHz ≤ $f_X$ ≤ 10 MHz |
| 1 | 1 MHz ≤ $f_X$ ≤ 20 MHz |

**Notes 1.** The 20-, 30-, and 32-pin products do not have a subsystem clock ($f_{SUB}$). If the low-speed on-chip oscillator is selected as the source of the clock signal for the CPU/peripheral hardware clock ($f_{CLK}$) or for a peripheral function, set the SELLOSC bit to 1.

**2.** When the SELLOSC bit is set to 1, the subsystem clock ($f_{SUB}$) cannot be supplied to the input clock ($f_{RTC}$) of the real-time clock.

**3.** As the XT oscillator becomes oscillation mode with lower power consumption, then its oscillation margin becomes smaller.

**Cautions 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. When the CMC register is used at the default value (00H), be sure to set 00H to this register after reset release in order to prevent malfunctioning during a program loop.**

2. **After reset release, set the CMC register before X1 or XT1 oscillation is started as set by the clock operation status control register (CSC).**
3. **Be sure to set the AMPH bit to 1 if the X1 clock oscillation frequency exceeds 10 MHz.**
   **When the X1 clock oscillation frequency is in the range from 1 to 10 MHz, setting the AMPH bit to 1 improves the oscillation margin.**
4. **The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.**

   - **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
   - **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
   - **Configure the circuit of the circuit board, using material with little parasitic capacitance and wiring resistance.**
   - **Place a ground pattern that has the same potential as $V_{SS}$ as much as possible near the XT1 oscillator.**
   - **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
   - **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
   - **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

### 5.3.2 System Clock Control Register (CKC)

This register is used to select a CPU/peripheral hardware clock and a main system clock.

Set the CKC register by a 1-bit or 8-bit memory manipulation instruction.

Writing to the CKC register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation sets this register to 00H.

**Figure 5-4. Format of System Clock Control Register (CKC)**

Address: FFFA4H After reset: 00H R/W Note 1

| Symbol | <7> | <6> | <5> | <4> | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CKC | CLS | CSS | MCS | MCM0 | 0 | 0 | 0 | 0 |

| CLS | Status of CPU/peripheral hardware clock ($f_{CLK}$) |
|---|---|
| 0 | Main system/PLL select clock ($f_{MP}$) |
| 1 | Subsystem/low-speed on-chip oscillator select clock ($f_{SL}$) |

| CSS Notes 2, 3 | Selection of CPU/peripheral hardware clock ($f_{CLK}$) |
|---|---|
| 0 | Main system/PLL select clock ($f_{MP}$) |
| 1 | Subsystem/low-speed on-chip oscillator select clock ($f_{SL}$) |

| MCS | Status of main system clock ($f_{MAIN}$) |
|---|---|
| 0 | High-speed on-chip oscillator clock ($f_{IH}$) |
| 1 | High-speed system clock ($f_{MX}$) |

| MCM0 Notes 2, 4, 5 | Main system clock ($f_{MAIN}$) operation control |
|---|---|
| 0 | Selects the high-speed on-chip oscillator clock ($f_{IH}$) as the main system clock ($f_{MAIN}$) |
| 1 | Selects the high-speed system clock ($f_{MX}$) as the main system clock ($f_{MAIN}$) |

**Notes 1.** Bits 7 and 5 are read-only.

**2.** Changing the value of the MCM0 bit is prohibited while the CSS bit is set to 1.

**3.** When setting the CSS bit in the 20-, 30-, or 32-pin products, set bit 0 (SELLOSC) in the clock select register (CKSEL) to 1 (Selects $f_{IL}$).

**4.** Changing the value of the MCM0 bit is prohibited while the PLLON bit is set to 1.

**5.** To change the MCM0 bit from 0 to 1 while FRQSEL4 = 1 in the corresponding user option byte (at 000C2H or 020C2H), stop counting by the timer RD (setting the TSTART0 and TSTART1 bits in the TRDSTR register to 0) and disable clock or buzzer output (by setting the PCLOE0 bit in the CKS0 register to 0) before changing the MCM0 bit.

**Cautions 1. Be sure to set bits 0 to 3 of the CKC register to 0.**

**2. The clock set by the CSS bit is supplied to the CPU and peripheral hardware. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the real-time clock, clock output/buzzer output, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral hardware clock.**

**3. If the subsystem clock or low-speed on-chip oscillator clock is used as the peripheral hardware clock, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 34 to CHAPTER 36 ELECTRICAL SPECIFICATIONS.**

**4. When selecting $f_{IH}$ as the count source for timer RD, set $f_{CLK}$ to $f_{MP}$ before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing $f_{CLK}$ to a clock other than $f_{MP}$, clear bit 4 (TRD0EN) in peripheral enable register 1 (PER1) before changing.**

**Remark** For setting of the PLL clock, refer to **5.6.4 Examples of Setting PLL Circuit**.

### 5.3.3 Clock Operation Status Control Register (CSC)

This register is used to control the operations of the high-speed system clock, high-speed on-chip oscillator clock, and subsystem clock (except the low-speed on-chip oscillator clock).

Set the CSC register by a 1-bit or 8-bit memory manipulation instruction.

Writing to the CSC register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation sets this register to C0H.

**Figure 5-5. Format of Clock Operation Status Control Register (CSC)**

Address: FFFA1H After reset: C0H R/W

| Symbol | <7> | <6> | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| CSC | MSTOP | XTSTOP | 0 | 0 | 0 | 0 | 0 | HIOSTOP |

| MSTOP | High-speed system clock operation control | | |
|---|---|---|---|
| | X1 oscillation mode | External clock input mode | Input port mode |
| 0 | X1 oscillator operating | External clock from EXCLK pin is valid | Input port |
| 1 | X1 oscillator stopped | External clock from EXCLK pin is invalid | |

| XTSTOP Note | Subsystem clock operation control | | |
|---|---|---|---|
| | XT1 oscillation mode | External clock input mode | Input port mode |
| 0 | XT1 oscillator operating | External clock from EXCLKS pin is valid | Input port |
| 1 | XT1 oscillator stopped | External clock from EXCLKS pin is invalid | |

| HIOSTOP | High-speed on-chip oscillator clock operation control |
|---|---|
| 0 | High-speed on-chip oscillator operating |
| 1 | High-speed on-chip oscillator stopped |

**Note** When setting the CSC register in the 20-, 30-, or 32-pin products, use the register with the XTSTOP bit set to 1 without changing from the default.

**Cautions 1. After reset release, set the clock operation mode control register (CMC) before setting the CSC register.**

**2. Set the oscillation stabilization time select register (OSTS) before setting the MSTOP bit to 0 after releasing reset. Note that if the OSTS register is being used with its default settings, the OSTS register is not required to be set here.**

**3. To start X1 oscillation as set by the MSTOP bit, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).**

**4. When starting XT1 oscillation by setting the XTSTOP bit, wait for oscillation of the subsystem clock to stabilize by setting a wait time using software.**

**5. Do not stop the clock selected for the CPU peripheral hardware clock ($f_{CLK}$) with the OSC register.**

**6. The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) and the condition before clock oscillation is to be stopped are as Table 5-2.**

**Table 5-2. Condition Before Stopping Clock Oscillation and Flag Setting**

| Clock | Condition Before Stopping Clock (Invalidating External Clock Input) | Setting of CSC Register Flags |
|---|---|---|
| X1 clock<br>External main system clock | CPU and peripheral hardware clocks operate with a clock other than the high-speed system clock or PLL clock (source clock = high-speed system clock).<br>(CLS (bit 7 of the CKC register) = 0 and MCS (bit 5 of the CKC register) = 0, or CLS = 1) | MSTOP = 1 |
| XT1 clock<br>External subsystem clock | CPU and peripheral hardware clocks operate with a clock other than the subsystem clock.<br>(CLS = 0, or CLS = 1 and SELLOSC (bit 0 of the CKSEL register) = 1) | XTSTOP = 1 |
| High-speed on-chip oscillator clock | CPU and peripheral hardware clocks operate with a clock other than the high-speed on-chip oscillator clock or PLL clock (source clock = high-speed on-chip oscillator clock).<br>(CLS = 0 and MCS = 1, or CLS = 1) | HIOSTOP = 1 |

### 5.3.4 Oscillation Stabilization Time Counter Status Register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.

The X1 clock oscillation stabilization time can be checked in the following case,

- If the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem/low-speed on-chip oscillator select clock is being used as the CPU clock.
- If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.

The generation of reset signal, the STOP instruction and MSTOP (bit 7 of clock operation status control register (CSC)) = 1 clear the OSTC register to 00H.

**Remark** The oscillation stabilization time counter starts counting in the following cases.

- When oscillation of the X1 clock starts (when EXCLK of the CMC register = 0 and OSCSEL of the CMC register = 1, MSTOP of the CSC register = 0)
- When the STOP mode is released

**Figure 5-6. Format of Oscillation Stabilization Time Counter Status Register (OSTC)**

Address: FFFA2H After reset: 00H R

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSTC | MOST 8 | MOST 9 | MOST 10 | MOST 11 | MOST 13 | MOST 15 | MOST 17 | MOST 18 |

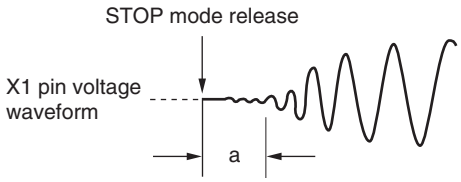
| MOST 8 | MOST 9 | MOST 10 | MOST 11 | MOST 13 | MOST 15 | MOST 17 | MOST 18 | Oscillation stabilization time status | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | $f_X$ = 10 MHz | $f_X$ = 20 MHz |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $2^8/f_X$ max. | 25.6 $\mu$s max. | 12.8 $\mu$s max. |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $2^8/f_X$ min. | 25.6 $\mu$s min. | 12.8 $\mu$s min. |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | $2^9/f_X$ min. | 51.2 $\mu$s min. | 25.6 $\mu$s min. |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | $2^{10}/f_X$ min. | 102.4 $\mu$s min. | 51.2 $\mu$s min. |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | $2^{11}/f_X$ min. | 204.8 $\mu$s min. | 102.4 $\mu$s min. |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | $2^{13}/f_X$ min. | 819.2 $\mu$s min. | 409.6 $\mu$s min. |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | $2^{15}/f_X$ min. | 3.27 ms min. | 1.63 ms min. |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | $2^{17}/f_X$ min. | 13.10 ms min. | 6.55 ms min. |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | $2^{18}/f_X$ min. | 26.21 ms min. | 13.10 ms min. |

**Cautions 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.**

**2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS). In the following cases, set the oscillation stabilization time of the OSTS register to the value equal to or greater than the count value which is to be checked by the OSTC register.**

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating. (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)**

**3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).**



### 5.3.5 Oscillation Stabilization Time Select Register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time.

When the X1 clock is oscillated, the operation automatically waits for the time set using the OSTS register.

When oscillation of the X1 clock starts, confirm with the oscillation stabilization time counter status register (OSTC) that the desired oscillation stabilization time has elapsed. The oscillation stabilization time can be checked up to the time set using the OSTC register.

Set the OSTS register by an 8-bit memory manipulation instruction.

Writing to the OSTS register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation sets the OSTS register to 07H.

**Figure 5-7. Format of Oscillation Stabilization Time Select Register (OSTS)**

Address: FFFA3H After reset: 07H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSTS | 0 | 0 | 0 | 0 | 0 | OSTS2 | OSTS1 | OSTS0 |

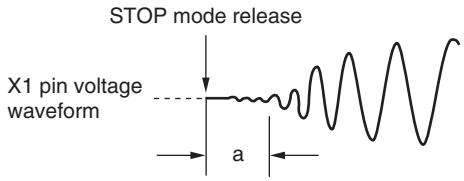
| OSTS2 | OSTS1 | OSTS0 | Oscillation stabilization time selection | | |
|---|---|---|---|---|---|
| | | | | $f_X$ = 10 MHz | $f_X$ = 20 MHz |
| 0 | 0 | 0 | $2^8/f_X$ | 25.6 μs | 12.8 μs |
| 0 | 0 | 1 | $2^9/f_X$ | 51.2 μs | 25.6 μs |
| 0 | 1 | 0 | $2^{10}/f_X$ | 102.4 μs | 51.2 μs |
| 0 | 1 | 1 | $2^{11}/f_X$ | 204.8 μs | 102.4 μs |
| 1 | 0 | 0 | $2^{13}/f_X$ | 819.2 μs | 409.6 μs |
| 1 | 0 | 1 | $2^{15}/f_X$ | 3.27 ms | 1.63 ms |
| 1 | 1 | 0 | $2^{17}/f_X$ | 13.10 ms | 6.55 ms |
| 1 | 1 | 1 | $2^{18}/f_X$ | 26.21 ms | 13.10 ms |

**Cautions**

1. **To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTS register before executing the STOP instruction.**
2. **Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.**
3. **Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.**
4. **The oscillation stabilization time counter counts up to the oscillation stabilization time set by the OSTS register.**

   **In the following cases, set the oscillation stabilization time of the OSTS register to the value equal to or greater than the count value which is to be checked by the OSTC register after the oscillation starts.**

   - **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem clock is being used as the CPU clock.**
   - **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating. (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)**
5. **The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).**



### 5.3.6 Peripheral Enable Registers 0, 1, 2 (PER0, PER1, PER2)

These registers are used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use the peripheral functions below, which are controlled by these registers, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

- Real-time clock
- A/D converter
- Serial interface IICA0
- Serial array unit 1
- Serial array unit 0
- Timer array unit 1
- Timer array unit 0
- D/A converter
- Comparator
- Timer RD
- DTC
- Timer RJ
- LIN0
- LIN1
- CAN

The PER0, PER1, and PER2 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Writing to the PER0, PER1, and PER2 registers is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation clears these registers to 00H.

**Figure 5-8. Format of Peripheral Enable Register 0 (PER0) (1/3)**

Address: F00F0H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PER0 | RTCEN Note 1 | 0 | ADCEN | IICA0EN | SAU1EN | SAU0EN | TAU1EN | TAU0EN |

| RTCEN Note 1 | Control of supplying input clock Note 2 for real-time clock (RTC) |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by the real-time clock (RTC) cannot be written.<br>• The real-time clock (RTC) is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by the real-time clock (RTC) can be read and written. |

**Notes 1.** The RTCCL register should be set before setting the RTCEN bit to 1.

**2.** The input clock that can be controlled by the RTCEN bit is used when the register that is used by the real-time clock (RTC) is accessed from the CPU. The RTCEN bit cannot control supply of the operating clock to the RTC.

**Caution Be sure to clear the following bits to 0.**

**Bits 1, 3, 4, and 6 in the 20-, 30-, 32-, 48-, and 64-pin products of the RL78/F13 (LIN incorporated) whose code flash memory is in the range from 16 Kbytes and 64 Kbytes**

**Bits 4 and 6 in the 30-pin products of the RL78/F13 (CAN and LIN incorporated) and 30-pin products of the RL78/F14**

**Bit 6 in the products other than above**

**Figure 5-9. Format of Peripheral Enable Register 0 (PER0) (2/3)**

Address: F00F0H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PER0 | RTCEN | 0 | ADCEN | IICA0EN Notes 1, 2 | SAU1EN Note 1 | SAU0EN | TAU1EN | TAU0EN |

| ADCEN | Control of A/D converter input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by the A/D converter cannot be written.<br>• The A/D converter is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by the A/D converter can be read and written. |

| IICA0EN Notes 1, 2 | Control of serial interface IICA0 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by the serial interface IICA0 cannot be written.<br>• The serial interface IICA0 is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by the serial interface IICA0 can be read and written. |

| SAU1EN Note 1 | Control of serial array unit 1 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by the serial array unit 1 cannot be written.<br>• The serial array unit 1 is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by the serial array unit 1 can be read and written. |

| SAU0EN | Control of serial array unit 0 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by the serial array unit 0 cannot be written.<br>• The serial array unit 0 is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by the serial array unit 0 can be read and written. |

**Notes 1.** Not available in the 20-, 30-, 32-, 48-, and 64-pin products of the RL78/F13 (LIN incorporated) whose code flash memory is in the range from 16 Kbytes and 64 Kbytes.

**2.** Not available in the 30-pin products of the RL78/F13 (CAN and LIN incorporated) and the 30-pin products of the RL78/F14.

**Figure 5-10. Format of Peripheral Enable Register 0 (PER0) (3/3)**

Address: F00F0H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PER0 | RTCEN | 0 | ADCEN | IICA0EN | SAU1EN | SAU0EN | TAU1EN Note | TAU0EN |

| TAU1EN Note | Control of timer array unit 1 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by timer array unit 1 cannot be written.<br>• Timer array unit 1 is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by timer array unit 1 can be read and written. |

| TAU0EN | Control of timer array unit 0 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by timer array unit 0 cannot be written.<br>• Timer array unit 0 is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by timer array unit 0 can be read and written. |

**Note** Not available in the 20-, 30-, 32-, 48-, and 64-pin products of the RL78/F13 (LIN incorporated) whose code flash memory is in the range from 16 Kbytes and 64 Kbytes.

**Figure 5-11. Format of Peripheral Enable Register 1 (PER1) (1/2)**

Address: F02C0H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PER1 | DACEN Note 1 | 0 | CMPEN Note 1 | TRD0EN Note 2 | DTCEN | 0 | 0 | TRJ0EN |

| DACEN Note 1 | Control of D/A converter input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by the D/A converter cannot be written.<br>• The D/A converter is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by the D/A converter can be read and written. |

| CMPEN Note 1 | Control of comparator input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by comparator cannot be written.<br>• Comparator is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by comparator can be read and written. |

| TRD0EN Note 2 | Control of timer RD input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by timer RD cannot be written.<br>• Timer RD is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by timer RD can be read and written. |

**Notes 1.** Only in the RL78/F14.

**2.** When FRQSEL4 = 1 in the user option byte (000C2H/020C2H), set $f_{CLK}$ to $f_{IH}$ before setting bit 4 (TRD0EN) in peripheral enable register 1 (PER1). When changing $f_{CLK}$ to a clock other than $f_{IH}$, clear bit 4 (TRD0EN) of peripheral enable register 1 (PER1) before changing.

**Caution Be sure to clear the following bits to 0.**
**Bits 1, 2, 5, 6, and 7 in the RL78/F13 products**
**Bits 1, 2, and 6 in the RL78/F14 products**

**Figure 5-12. Format of Peripheral Enable Register 1 (PER1) (2/2)**

Address: F02C0H After reset: 00H R/W

| Symbol | <7> | 6 | <5> | <4> | <3> | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PER1 | DACEN | 0 | CMPEN | TRD0EN | DTCEN | 0 | 0 | TRJ0EN |

| DTCEN | Control of DTC input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• DTC cannot run. |
| 1 | Enables input clock supply.<br>• DTC can run. |

| TRJ0EN | Control of timer RJ0 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by timer RJ0 cannot be written.<br>• Timer RJ0 is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by timer RJ0 can be read and written. |

**Figure 5-13. Format of Peripheral Enable Register 2 (PER2)**

Address: F02C1H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | <3> | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PER2 | 0 | 0 | 0 | 0 | LIN1EN Note 1 | LIN0EN | 0 | CAN0EN Note 2 |

| LIN1EN Note 1 | Control of LIN1 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• Disables writing to the SFR used by LIN1.<br>• LIN1 is in the reset state. |
| 1 | Enables input clock supply.<br>• Enables reading from and writing to the SFR used by LIN1. |

| LIN0EN | Control of LIN0 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• Disables writing to the SFR used by LIN0.<br>• LIN0 is in the reset state. |
| 1 | Enables input clock supply.<br>• Enables reading from and writing to the SFR used by LIN0. |

| CAN0EN Note 2 | Control of CAN input clock supply/control of CANi wakeup interrupt |
|---|---|
| 0 | Stops input clock supply.<br>• Disables writing to the SFR used by CAN.<br>• CAN is in the reset state.<br>Disables CANi wakeup interrupt. |
| 1 | Enables input clock supply.<br>• Enables reading from and writing to the SFR used by CAN.<br>Enables CANi wakeup interrupt. |

**Notes 1.** Only in the RL78/F14 products with at least 128 Kbytes of code flash memory and the 100-pin products of the RL78/F14.

**2.** Only in the RL78/F13 (CAN and LIN incorporated) and RL78/F14 products.

**Caution Be sure to clear the following bits to 0.**
**Bits 0, 1, 3, 4, 5, 6, and 7 in the RL78/F13 (LIN incorporated) products**
**Bits 1, 3, 4, 5, 6, and 7 in the RL78/F13 (CAN and LIN incorporated) products and the RL78/F14 products with 30, 32, 48, 64, or 80 pins and up to 96 Kbytes of code flash memory**
**Bits 1, 4, 5, 6, and 7 in the RL78/F14 products with at least 128 Kbytes of code flash memory and the 100-pin products of the RL78/F14**

### 5.3.7 Operation Speed Mode Control Register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions is stopped in STOP mode or HALT mode while subsystem/low-speed on-chip oscillator select clock is selected as CPU clock.

Set the OSMC register by an 8-bit memory manipulation instruction.

Writing to the OSMC register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation clears this register to 00H.

**Figure 5-14. Format of Operation Speed Mode Control Register (OSMC)**

Address: F00F3H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSMC | RTCLPC | 0 | 0 | WUTMMCK0 Note | 0 | 0 | 0 | 0 |

| RTCLPC | Setting in STOP mode or HALT mode while subsystem/low-speed on-chip oscillator select clock is selected as CPU clock |
|---|---|
| 0 | Enables supply of subsystem/low-speed on-chip oscillator select clock to peripheral functions<br>(See **Table 23-1** for peripheral functions whose operations are enabled.) |
| 1 | Stops supply of subsystem/low-speed on-chip oscillator select clock to peripheral functions |

| WUTMMCK0 Note | Low-speed on-chip oscillator operation control |
|---|---|
| 0 | Low-speed on-chip oscillator stopped |
| 1 | Low-speed on-chip oscillator operating |

**Note** To stop the low-speed on-chip oscillator, set bit 4 (WUTMMCK0) to 0 and bit 1 (SELLOSC) of the clock select register (CKSEL) to 0.

**Caution The STOP mode current or HALT mode current when the subsystem/low-speed on-chip oscillator select clock is used can be reduced by setting the RTCLPC bit to 1. However, no clock can be supplied to the peripheral functions during HALT mode while subsystem/low-speed on-chip oscillator select clock is selected as CPU clock. Set bit 7 (RTCEN) of peripheral enable registers 0 (PER0) to 1 and bits 0 to 6 of the PER0 register to 0 before setting HALT mode while the subsystem/low-speed on-chip oscillator clock is selected as CPU clock.**

### 5.3.8 High-Speed On-Chip Oscillator Frequency Select Register (HOCODIV)

The frequency of the high-speed on-chip oscillator which is set by a user option byte (000C2H/020C2H) can be changed by using high-speed on-chip oscillator frequency select register (HOCODIV). However, the selectable frequency depends on the FRQSEL4 and FRQSEL3 bits of the user option byte (000C2H/020C2H).

Set the HOCODIV register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to default value (the value set with the FRQSEL2 to FRQSEL0 bits of the user option byte (000C2H/020C2H)).

**Figure 5-15. Format of High-speed on-chip oscillator frequency select register (HOCODIV)**

Address: F00A8H After reset: value set with the FRQSEL2 to FRQSEL0 bits of the user option byte (000C2H/020C2H) R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| HOCODIV | 0 | 0 | 0 | 0 | 0 | HOCODIV2 | HOCODIV1 | HOCODIV0 |

| HOCODIV2 | HOCODIV1 | HOCODIV0 | Selection of high-speed on-chip oscillator clock frequency | | | |
|---|---|---|---|---|---|---|
| | | | 24-MHz base | 32-MHz base | 48-MHz base | 64-MHz base |
| | | | FRQSEL4 = 0 | | FRQSEL4 = 1 | |
| | | | FRQSEL3 = 0 | FRQSEL3 = 1 | FRQSEL3 = 0 | FRQSEL3 = 1 |
| 0 | 0 | 0 | $f_{IH}$ = 24 MHz | $f_{IH}$ = 32 MHz | $f_{IH}$ = 48 MHz | $f_{IH}$ = 64 MHz |
| 0 | 0 | 1 | $f_{IH}$ = 12 MHz | $f_{IH}$ = 16 MHz | $f_{IH}$ = 24 MHz | $f_{IH}$ = 32 MHz |
| 0 | 1 | 0 | $f_{IH}$ = 6 MHz | $f_{IH}$ = 8 MHz | $f_{IH}$ = 12 MHz | $f_{IH}$ = 16 MHz |
| 0 | 1 | 1 | $f_{IH}$ = 3 MHz | $f_{IH}$ = 4 MHz | $f_{IH}$ = 6 MHz | $f_{IH}$ = 8 MHz |
| 1 | 0 | 0 | Setting prohibited | $f_{IH}$ = 2 MHz | Setting prohibited | $f_{IH}$ = 4 MHz |
| 1 | 0 | 1 | Setting prohibited | $f_{IH}$ = 1 MHz | Setting prohibited | $f_{IH}$ = 2 MHz |
| Other than above | | | Setting prohibited | | | |

**Cautions**

1. **When setting of high-speed on-chip oscillator clock as system clock, the device operates at the old frequency for the duration of 3 clocks after the frequency value has been changed by using the HOCODIV register.**
2. **To change the frequency of the high-speed on-chip oscillator when X1 oscillation, external oscillation input, subsystem clock, or low-speed on-chip oscillator clock is set for the system clock, stop the high-speed on-chip oscillator by setting bit 0 (HIOSTOP) of the CSC register to 1 and then change the frequency.**
3. **To change the frequency of the high-speed on-chip oscillator when X1 oscillation or external oscillation input is set for the clock source of the PLL clock, and PLL clock is set for the system clock, stop the high-speed on-chip oscillator by setting bit 0 (HIOSTOP) of the CSC register to 1 and then change the frequency.**
4. **Do not change the setting of the HOCODIV register when the high-speed on-chip oscillator clock is used as the clock source of the PLL clock and the PLL clock is used as the system clock.**

### 5.3.9 High-Speed On-Chip Oscillator Trimming Register (HIOTRM)

This register is used to adjust the accuracy of the high-speed on-chip oscillator.

With self-measurement of the high-speed on-chip oscillator frequency via a timer using high-accuracy external clock input, and so on, the accuracy can be adjusted.

Set the HIOTRM register by an 8-bit memory manipulation instruction.

**Caution The frequency will vary if the temperature and $V_{DD}$ pin voltage change after accuracy adjustment. When the temperature and $V_{DD}$ voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.**

**Figure 5-16. Format of High-Speed On-Chip Oscillator Trimming Register (HIOTRM)**

Address: F00A0H After reset: **Note** R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| HIOTRM | 0 | 0 | HIOTRM5 | HIOTRM4 | HIOTRM3 | HIOTRM2 | HIOTRM1 | HIOTRM0 |

| HIOTRM5 | HIOTRM4 | HIOTRM3 | HIOTRM2 | HIOTRM1 | HIOTRM0 | High-speed on-chip oscillator |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | Minimum speed |
| 0 | 0 | 0 | 0 | 0 | 1 | |
| 0 | 0 | 0 | 0 | 1 | 0 | |
| 0 | 0 | 0 | 0 | 1 | 1 | |
| 0 | 0 | 0 | 1 | 0 | 0 | |
| • | • | • | • | • | • | |
| 1 | 1 | 1 | 1 | 1 | 0 | |
| 1 | 1 | 1 | 1 | 1 | 1 | Maximum speed |

**Note** The reset value differs for each chip.

### 5.3.10 CAN Clock Select Register (CANCKSEL)

This register is used to control the X1 clock (fx) supplied to the CAN.

Set the CANCKSEL register by a 1-bit or 8-bit memory manipulation instruction.

Writing to the CANCKSEL register is disabled when the GCSC bit of the IAWCTL register is set to 1.

**Figure 5-17. Format of CAN Clock Select Register (CANCKSEL)**

Address: F02C2H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| CANCKSEL | 0 | 0 | 0 | 0 | 0 | 0 | 0 | CAN0MCKE Note |

| CAN0MCKE Note | Control of supplying or stopping CAN X1 clock (fx) |
|---|---|
| 0 | Stops CAN X1 clock (fx) supply. |
| 1 | Enables CAN X1 clock (fx) supply. |

**Note** Only in the RL78/F13 (CAN and LIN incorporated) and RL78/F14 products.

### 5.3.11 LIN Clock Select Register (LINCKSEL)

This register is used to control the communication clock source supplied to the LIN.

Set the LINCKSEL register by a 1-bit or 8-bit memory manipulation instruction.

Writing to the LINCKSEL register is disabled when the GCSC bit of the IAWCTL register is set to 1.

**Figure 5-18. Format of LIN Clock Select Register (LINCKSEL)**

Address: F02C3H After reset: 00H R/W

| Symbol | 7 | 6 | <5> | <4> | 3 | 2 | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| LINCKSEL | 0 | 0 | LIN1MCKE Note | LIN0MCKE | 0 | 0 | LIN1MCK Note | LIN0MCK |

| LIN1MCKE Note | Control of supplying or stopping LIN1 communication clock source |
|---|---|
| 0 | Stops LIN communication clock source supply. |
| 1 | Enables LIN communication clock source supply. |

| LIN0MCKE | Control of supplying or stopping LIN0 communication clock source |
|---|---|
| 0 | Stops LIN communication clock source supply. |
| 1 | Enables LIN communication clock source supply. |

| LIN1MCK Note | Control of selecting LIN1 communication clock source |
|---|---|
| 0 | Selects the $f_{CLK}$ clock. |
| 1 | Selects the $f_{MX}$ clock. |

| LIN0MCK | Control of selecting LIN0 communication clock source |
|---|---|
| 0 | Selects the $f_{CLK}$ clock. |
| 1 | Selects the $f_{MX}$ clock. |

**Note** Only in the RL78/F14 products with 48 pins and at least 128 Kbytes of code flash memory and the 100-pin products of the RL78/F14.

**Cautions 1. Select the LINn operating clock with the LINnMCK bit before setting the LINnMCKE (n = 0, 1) bit to 1.**

**2. When operating LINn in SNOOZE mode, set the LINnMCK bit to 0.**

<R> **3. In case of LINnMCK is set to 1, do not use the timeout error detection.**
**In that case, set at least 1.2 times the frequency of the LIN communication clock source to the $f_{CLK}$ clock.**

### 5.3.12 Clock Select Register (CKSEL)

This register is used to select the CPU clock ($f_{SUB}$/$f_{IL}$) and the clocks for the timer RJ, the timer RD, and clock output/buzzer output. Together with the CMC register, the SELLOSC bit is used to set the operation mode of the subsystem clock. For details, see **Figure 5-3 Format of Clock Operation Mode Control Register (CMC)**.

Set the CKSEL register by a 1-bit or 8-bit memory manipulation instruction.

Writing to the CKSEL register is disabled when the GCSC bit of the IAWCTL register is set to 1.

**Figure 5-19. Format of Clock Select Register (CKSEL)**

Address: F02C4H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| CKSEL | 0 | 0 | 0 | 0 | 0 | TRD_ CKSEL | 0 | SELLOSC Note 5, 6, 7 |

| TRD_CKSEL | Control of TDR clock selection |
|---|---|
| 0 | Selects $f_{CLK}$ or $f_{MP}$ Note 1 |
| 1 | Selects $f_{SL}$ Note 2 |

| SELLOSC Note 5, 6, 7 | Control of subsystem/low-speed on-chip oscillator selection clock ($f_{SL}$) selection |
|---|---|
| 0 | Selects $f_{SUB}$ Note 3 and stopping the low-speed on-chip oscillator |
| 1 | Selects $f_{IL}$ Note 4 and running the low-speed on-chip oscillator |

**Notes 1.** When FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and PLLDIV1 = 1 ($f_{PLL}$ > 32 MHz) in the PLLCTL register, set the TRD_CKSEL bit to 0.
When FRQSEL4 = 1 in the user option byte (000C2H/020C2H) or PLLDIV1 = 1 ($f_{PLL}$ > 32 MHz) in the PLLCTL register, the timer RD clock becomes $f_{MP}$.

**2.** When $f_{SL}$ is selected as the timer RD clock, $f_{SL}$ should be selected as the CPU clock (set the CSS bit in the CKC register to 1) before setting the TRD0EN bit in the peripheral enable register 1 (PER1) to 1.

**3.** When setting $f_{SUB}$ as the CPU/peripheral hardware clock, first set the SELLOSC bit in the CKSEL register to 0 and then set the CSS bit in the CKC register to 1 .

**4.** When setting $f_{IL}$ as the CPU/peripheral hardware clock, first set the SELLOSC bit in the CKSEL register to 1 and then set the CSS bit in the CKC register to 1.

**5.** When the SELLOSC bit is set to 1, the low-speed on-chip oscillator operates. To stop the low-speed on-chip oscillator, set the WUTMMCK0 bit in the OSMC register to 0 and the SELLOSC bit to 0.

**6.** The 20-, 30-, and 32-pin products do not have a subsystem clock ($f_{SUB}$). If the low-speed on-chip oscillator is selected as the source of the clock signal for the CPU/peripheral hardware clock ($f_{CLK}$) or for a peripheral function, set the SELLOSC bit to 1.

**7.** When the SELLOSC bit is set to 1, the subsystem clock ($f_{SUB}$) cannot be supplied to the input clock ($f_{RTC}$) of the real-time clock.

### 5.3.13 PLL Control Register (PLLCTL)

This register is used to control the PLL function. The system clock multiplied by 3, 4, 6, or 8 times or not multiplied at all can be selected as the CPU clock and peripheral hardware clock.

Set the PLLCTL register by a 1-bit or 8-bit memory manipulation instruction.

Writing to the PLLCTL register is disabled when the GCSC bit of the IAWCTL register is set to 1.

**Figure 5-20. Format of PLL Control Register (PLLCTL)**

Address: F02C5H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | 3 | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PLLCTL | LCKSEL1 | LCKSEL0 | PLLDIV1 | PLLDIV0 | 0 | SELPLL | PLLMUL | PLLON |

| LCKSEL1 | LCKSEL0 | Control of setting lock-up wait counter |
|---|---|---|
| 0 | 0 | Selects 128/$f_{MAIN}$. |
| 0 | 1 | Selects 256/$f_{MAIN}$. |
| 1 | 0 | Selects 512/$f_{MAIN}$. |
| 1 | 1 | Setting prohibited |

| PLLDIV1 | Control of PLL output clock selection |
|---|---|
| 0 | When $f_{PLL} \leq 32$ MHz |
| 1 | When $f_{PLL} > 32$ MHz |

| PLLDIV0 | Control of PLL division selection |
|---|---|
| 0 | Divides the clock frequency by 2. |
| 1 | Divides the clock frequency by 4. |

| SELPLL | Control of clock mode selection |
|---|---|
| 0 | Clock through mode ($f_{MAIN}$) |
| 1 | PLL-clock-selected mode ($f_{PLL}$) |

| PLLMUL | Control of PLL multiplication selection |
|---|---|
| 0 | Multiplies the clock frequency by 12. |
| 1 | Multiplies the clock frequency by 16. |

| PLLON | Control of PLL operation |
|---|---|
| 0 | Stops PLL operation. |
| 1 | Starts PLL operation. |
| After PLL operation starts, the lock-up wait time for frequency stabilization is required. | |

**Cautions 1.** Writing to the SELPLL bit is disabled when the PLL output is not stable (LOCK bit of the PLLSTS register = 0).

**2.** When the clock monitor detects that the main system/PLL select clock has been stopped, the SELPLL bit is not automatically cleared.

**3.** When the clock monitor detects that the main system/PLL select clock has been stopped, the SELPLLS bit in the PLLSTS register is automatically cleared.

4. When the clock monitor detects that the main system/PLL select clock has been stopped, even if the SELPLL bit is set to 1 (SELPLL = 1), the clock through mode is entered.
5. The counter for the lock-up wait time should be set to a period of at least 40 μs.
6. When PLL operation starts, a wait time for the PLL to be locked is required.
7. When the PLL circuit is used, the PLL input clock and multiplication value can be set only in the combinations shown in the following. When the PLL circuit is not used (PLLON = 0 or SELPLL = 0), an input clock of any frequency between 1 to 32 MHz can be selected.

| PLLCTL Register | | | Inputtable Frequency ($f_{MAIN}$) | Multiplication | Division | Outputtable Frequency ($f_{PLL}$) |
|---|---|---|---|---|---|---|
| PLLMUL | PLLDIV1 | PLLDIV0 | | | | |
| 0 | 0 | 0 | 4MHz ± 2% | ×12 | 1/2 | 24MHz ± 2% |
| 0 | 0 | 1 | 8MHz ± 2% | ×12 | 1/4 | 24MHz ± 2% |
| 0 | 1 | 0 | 8MHz ± 2% | ×12 | 1/2 | 48MHz ± 2% |
| 1 | 0 | 0 | 4MHz ± 2% | ×16 | 1/2 | 32MHz ± 2% |
| 1 | 0 | 1 | 8MHz ± 2% | ×16 | 1/4 | 32MHz ± 2% |
| 1 | 1 | 0 | 8MHz ± 2% | ×16 | 1/2 | 64MHz ± 2% |
| Other than above | | | Setting prohibited | | | |

8. When PLLON = 0, simultaneously changing the PLLON bit and SELPLL bit through 8-bit access is disabled.
9. When the PLLON bit is cleared (becomes 0), the SELPLL bit is also automatically cleared (clock through mode).
10. Before entering STOP mode, the PLLON bit should be cleared to 0.
11. Do not change the value of the MCM0 bit of the CKC register while the PLLON bit is set to 1.
12. When FRQSEL4 = 1 in the user option byte (000C2H/020C2H), set the PLLDIV1 bit to 0 ($f_{PLL} \leq 32$ MHz).
13. To change the SELPLL bit from 1 to 0 while PLLDIV1 = 1 ($f_{PLL} > 32$ MHz), stop counting by the timer RD (setting the TSTART0 and TSTART1 bits in the TRDSTR register to 0) before changing the SELPLL bit.

**Remark** When the PLLON and SELPLL bits are set, the clock selected for $f_{PLL}$ is determined according to the state of the LOCK and SELPLLS bits of the PLLSTS register. $f_{PLL}$ for each state of the PLLON, SELPLL, LOCK, and SELPLL bits is shown in the following.

| PLLON | SELPLL | LOCK | SELPLLS | Selected Clock ($f_{PLL}$) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Main system clock ($f_{MAIN}$) |
| 1 | 0 | 0 | 0 | Main system clock ($f_{MAIN}$) |
| 1 | 0 | 1 | 0 | Main system clock ($f_{MAIN}$) |
| 1 | 1 | 1 | 0 | Main system clock ($f_{MAIN}$)<br>State in which after the SELPLL bit is set to 1, the clock has not switched to the multiplied clock. |
| 1 | 1 | 1 | 1 | PLL clock ($f_{PLL}$) |
| Other than above | | | | Setting prohibited |

### 5.3.14 PLL Status Register (PLLSTS)

This register is used to indicate the operation status of the PLL clock.

Read the PLLSTS register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

**Figure 5-21. Format of PLL Status Register (PLLSTS)**

Address: F02C6H After reset: 00H R

| Symbol | <7> | 6 | 5 | 4 | <3> | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PLLSTS | LOCK | 0 | 0 | 0 | SELPLLS | 0 | 0 | 0 |

| LOCK | PLL locked state |
|---|---|
| 0 | Unlocked state |
| 1 Note | Locked state |
| This bit is set to 1 when the lock-up wait counter overflows. | |

| SELPLLS | CLock mode state |
|---|---|
| 0 | Clock through mode ($f_{MAIN}$) |
| 1 | PLL-clock-selected mode ($f_{PLL}$) |

**Note** When PLL operation starts, a wait time for the PLL to be locked (LOCK = 1) is required.

### 5.3.15 $f_{MP}$ Clock Division Register (MDIV)

This register is used to divide the frequency of the $f_{MP}$ clock (1/2, 1/4, 1/8, 1/16, 1/32, or 1/64).

Set the MDIV register by an 8-bit memory manipulation instruction.

Writing to the MDIV register is disabled when the GCSC bit of the IAWCTL register is set to 1.

**Figure 5-22. Format of $f_{MP}$ Clock Division Register (MDIV)**

Address: F02C7H After reset: 00H/01H Note R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| MDIV | 0 | 0 | 0 | 0 | 0 | MDIV2 | MDIV1 | MDIV0 |

| MDIV2 | MDIV1 | MDIV0 | |
|---|---|---|---|
| 0 | 0 | 0 | Selects $f_{MP}$. |
| 0 | 0 | 1 | Selects $f_{MP}/2$. |
| 0 | 1 | 0 | Selects $f_{MP}/4$. |
| 0 | 1 | 1 | Selects $f_{MP}/8$. |
| 1 | 0 | 0 | Selects $f_{MP}/16$. |
| 1 | 0 | 1 | Selects $f_{MP}/32$. |
| 1 | 1 | 0 | Selects $f_{MP}/64$. |
| Other than above | | | Setting prohibited |

**Note** The value of the FRQSEL4 bit in the user option byte (000C2H/020C2H) becomes the initial value of the MDIV0 bit in the MDIV register.

**Cautions 1. When setting the MDIV register, make the frequency after division of $f_{MP}$ be within the range of 1 MHz to 32 MHz (or 1 MHz to 24 MHz for grade-K and grade-Y products).**

**2. When FRQSEL4 = 1 in the user option byte (000C2H/020C2H), set the MDIV2 to MDIV0 bits to 001 (division by 2). Setting these bits to 001 (division by 2) is unnecessary in the clock through mode by PLL oscillation stop detection.**

**3. When the PLLDIV1 bit in the PLLCTL register is 1 ($f_{PLL}$ > 32 MHz), set the MDIV2 to MDIV0 bits to 001 (division by 2). Setting these bits to 001 (division by 2) is unnecessary in the clock through mode by PLL oscillation stop detection.**

**4. When 64 MHz or 48 MHz is selected as $f_{IH}$, the initial setting of the MDIV register is "division by 2" so that $f_{CLK}$ is set to 32 MHz or 24 MHz, respectively.**

## 5.4 System Clock Oscillator

### 5.4.1 X1 Oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (1 to 20 MHz) connected to the X1 and X2 pins.

An external clock can also be input. In this case, input the clock signal to the EXCLK pin.

To use the X1 oscillator, set bits 7 and 6 (EXCLK, OSCSEL) of the clock operation mode control register (CMC) as follows.

- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

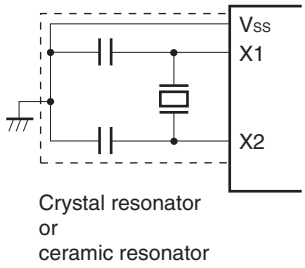
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

When the pins are not even used as input port pins, see **2.3 Processing of Unused Pins**.

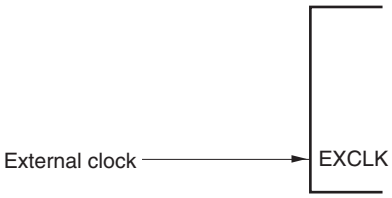
Figure 5-23 shows an example of the external circuit of the X1 oscillator.

**Figure 5-23. Example of External Circuit of X1 Oscillator**

**(a) Crystal or ceramic oscillation**




**(b) External clock**




**Cautions** are listed on the next page.

### 5.4.2 XT1 Oscillator

The XT1 oscillator oscillates with a crystal resonator (standard: 32.768 kHz) connected to the XT1 and XT2 pins.

To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

An external clock can also be input. In this case, input the clock signal to the EXCLKS pin.

To use the XT1 oscillator, set bits 5 and 4 (EXCLKS, OSCSELS) of the clock operation mode control register (CMC) as follows.

- Crystal oscillation: EXCLKS, OSCSELS = 0, 1
- External clock input: EXCLKS, OSCSELS = 1, 1

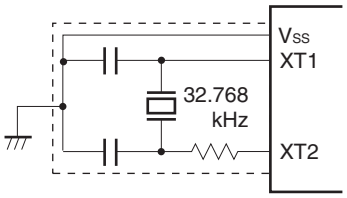
When the XT1 oscillator is not used, set the input port mode (EXCLKS, OSCSELS = 0, 0).

When the pins are not even used as input port pins, see **2.3 Processing of Unused Pins**.

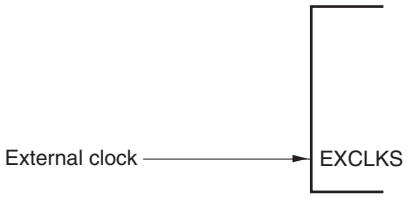
Figure 5-24 shows an example of the external circuit of the XT1 oscillator.

**Figure 5-24. Example of External Circuit of XT1 Oscillator**

**(a) Crystal oscillation**




**(b) External clock**




**Caution When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in Figures 5-23 and 5-24 to avoid an adverse effect from wiring capacitance.**

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as Vss. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

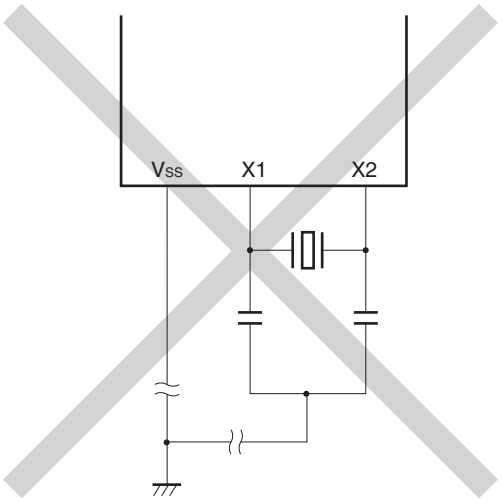
**The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.**

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential as Vss as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

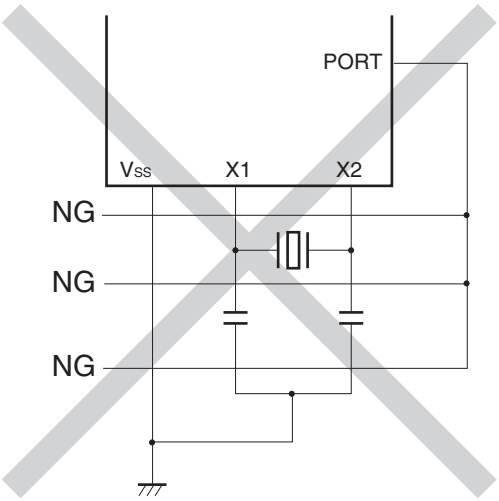
Figure 5-25 shows examples of incorrect resonator connection.

**Figure 5-25. Examples of Incorrect Resonator Connection (1/2)**

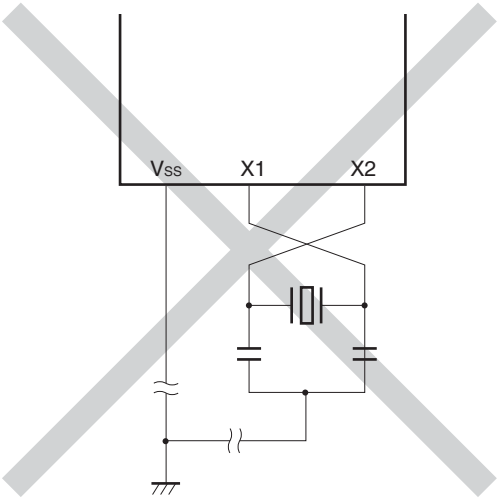
**(a) Too long wiring**



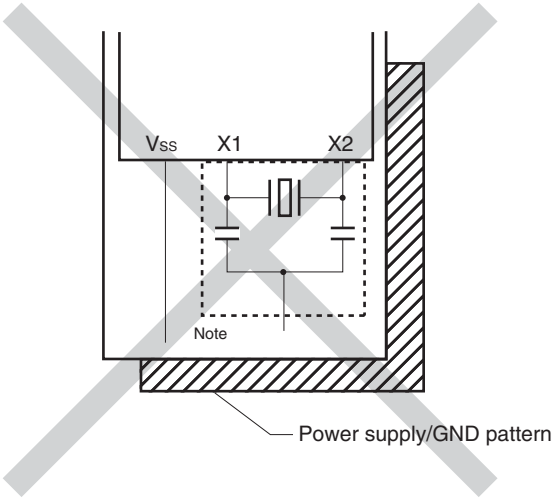

**(b) Crossed signal line**




**(c) The X1 and X2 signal line wires cross.**




**(d) A power supply/GND pattern exists under the X1 and X2 wires.**

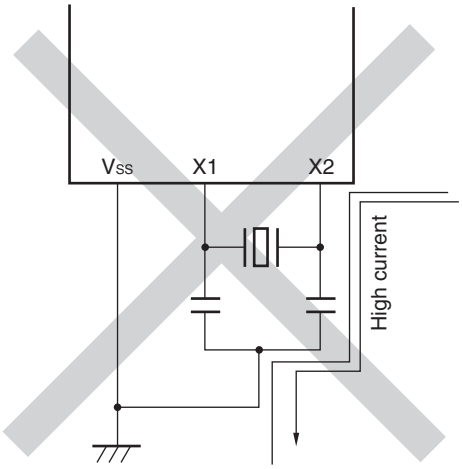



**Note** Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

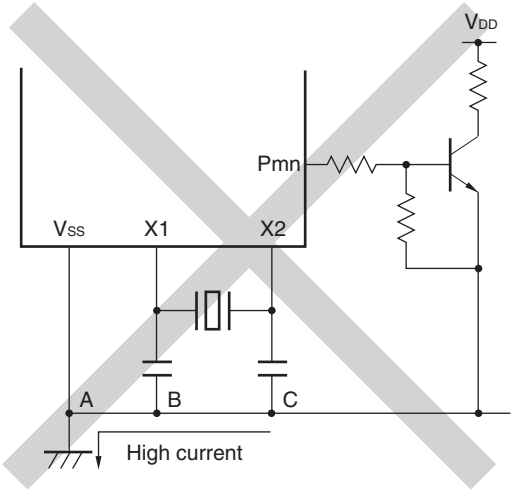
**Remark** When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

**Figure 5-25. Examples of Incorrect Resonator Connection (2/2)**

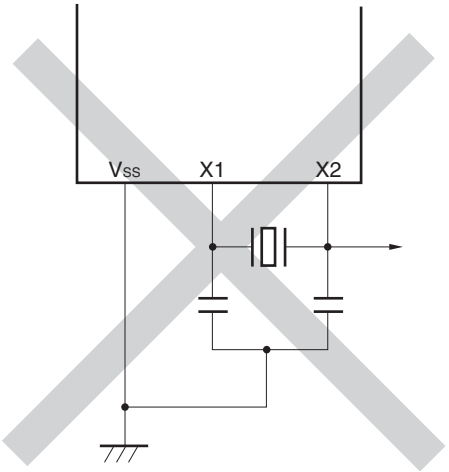
**(e) Wiring near high alternating current**




**(f) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)**




**(g) Signals are fetched**




**Caution When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.**

**Remark** When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

### 5.4.3 High-Speed On-Chip Oscillator

The high-speed on-chip oscillator is incorporated in the RL78/F13 and RL78/F14. The frequency can be selected from among 64, 48, 32, 24, 16, 12, 8, 4, or 1 MHz by using the user option byte (000C2H/020C2H). When 64 MHz or 48 MHz is selected, the frequency obtained by dividing the selected clock by 2 by the $f_{MP}$ clock division register (MDIV) is supplied as the CPU clock after a reset release. Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC).

The high-speed on-chip oscillator automatically starts oscillating after reset release.

### 5.4.4 PLL Circuit

The PLL circuit is incorporated in the RL78/F13 and RL78/F14. Operation of the PLL circuit can be controlled by bit 0 (PLLON) of the PLL control register (PLLCTL).

### 5.4.5 Low-Speed On-Chip Oscillator

The low-speed on-chip oscillator which can be used for the CPU/peripheral hardware clock is incorporated in the RL78/F13 and RL78/F14.

### 5.4.6 WDT-Dedicated Low-Speed On-Chip Oscillator

The WDT-dedicated low-speed on-chip oscillator is incorporated in the RL78/F13 and RL78/F14.

The WDT-dedicated low-speed on-chip oscillator clock is used as the watchdog timer clock. This clock cannot be used as the CPU clock.

The WDT-dedicated low-speed on-chip oscillator operates when bit 4 (WDTON) of the user option byte (000C0H/020C0H) is set to 1. The WDT-dedicated low-speed on-chip oscillator continues oscillating while the watchdog timer is operating. The WDT-dedicated low-speed on-chip oscillator does not stop while the watchdog timer is operating even though the program goes out of control.

## 5.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 5-1**).

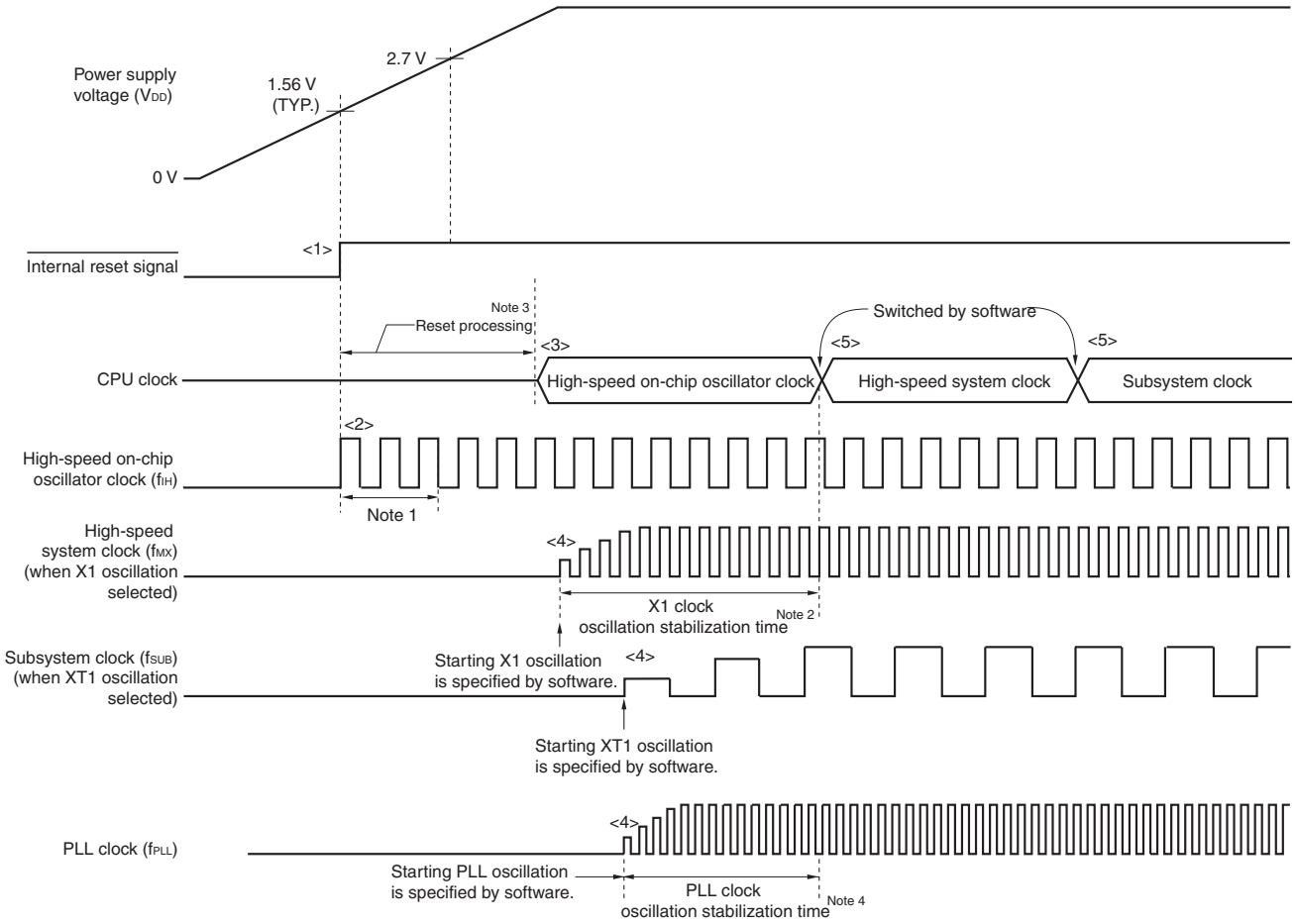
- Main system clock $f_{MAIN}$
  - High-speed system clock $f_{MX}$
    X1 clock $f_X$
    External main system clock $f_{EX}$
  - High-speed on-chip oscillator clock $f_{IH}$
- Subsystem clock $f_{SUB}$
  - XT1 clock $f_{XT}$
  - External subsystem clock $f_{EXS}$
- PLL clock $f_{PLL}$
- Low-speed on-chip oscillator clock $f_{IL}$
- CPU/peripheral hardware clock $f_{CLK}$

**Caution The subsystem clock is only in products with at least 48 pins.**

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the RL78/F13 and RL78/F14.

When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-26.

**Figure 5-26. Clock Generator Operation When Power Supply Voltage Is Turned On**




<1> When the power is turned on, an internal reset signal is generated by the power-on-reset (POR) circuit.

<2> When the power supply voltage exceeds 1.56 V (TYP.), the reset is released and the high-speed on-chip oscillator automatically starts oscillation.

<3> The CPU starts operation on the high-speed on-chip oscillator clock after a reset processing such as waiting for the voltage of the power supply or regulator to stabilize has been performed after reset release.

<4> Set the start of oscillation of the X1 clock, XT1 clock, low-speed on-chip oscillator, or PLL clock via software (see **5.6.2 Example of Setting X1 Oscillator**, **5.6.3 Example of Setting XT1 Oscillator**, **5.6.4 Examples of Setting PLL Circuit**, or **5.6.5 Example of Setting Low-Speed On-Chip Oscillator**).

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **5.6.2 Example of Setting X1 Oscillator** and **5.6.3 Example of Setting XT1 Oscillator**).

**Notes 1.** The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

**2.** When releasing a reset, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC).

**3.** For details about the reset processing time, see **CHAPTER 25 POWER-ON-RESET (POR) CIRCUIT**.

**4.** When the PLL circuit starts operation, time is required so that the PLL circuit becomes locked (LOCK = 1).

**Caution It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.**

## 5.6 Controlling Clock

### 5.6.1 Example of Setting High-Speed On-Chip Oscillator

After a reset release, the CPU/peripheral hardware clock ($f_{CLK}$) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected from 64, 48, 32, 24, 16, 12, 8, 4, and 1 MHz by using FRQSEL0 to FRQSEL4 of the user option byte (000C2H/020C2H). In addition, Oscillation can be changed by the internal high-speed on-chip oscillator frequency select register (HOCODIV).

[User option byte setting]

Address: 000C2H/020C2H After reset: - (user setting value)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Option byte | 1 | 1 | RESOUTB<br>0/1 | FRQSEL4<br>0/1 | FRQSEL3<br>0/1 | FRQSEL2<br>0/1 | FRQSEL1<br>0/1 | FRQSEL0<br>0/1 |

| FRQSEL4 | FRQSEL3 | FRQSEL2 | FRQSEL1 | FRQSEL0 | Frequency of the high-speed on-chip oscillator<br>$f_{IH}$ |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 64 MHz |
| 1 | 0 | 0 | 0 | 0 | 48 MHz |
| 0 | 1 | 0 | 0 | 0 | 32 MHz |
| 0 | 0 | 0 | 0 | 0 | 24 MHz |
| 0 | 1 | 0 | 0 | 1 | 16 MHz |
| 0 | 0 | 0 | 0 | 1 | 12 MHz |
| 0 | 1 | 0 | 1 | 0 | 8 MHz |
| 0 | 1 | 0 | 1 | 1 | 4 MHz |
| 0 | 1 | 1 | 0 | 1 | 1 MHz |
| Other than above | | | | | Setting prohibited |

[High-speed on-chip oscillator frequency select register (HOCODIV) setting]

Address: F00A8H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| HOCODIV | 0 | 0 | 0 | 0 | 0 | HOCODIV2 | HOCODIV1 | HOCODIV0 |

| HOCODIV2 | HOCODIV1 | HOCODIV0 | Selection of high-speed on-chip oscillator clock frequency | | | |
|---|---|---|---|---|---|---|
| | | | 24-MHz base | 32-MHz base | 48-MHz base | 64-MHz base |
| | | | FRQSEL4 = 0 | | FRQSEL4 = 1 | |
| | | | FRQSEL3 = 0 | FRQSEL3 = 1 | FRQSEL3 = 0 | FRQSEL3 = 1 |
| 0 | 0 | 0 | $f_{IH}$ = 24 MHz | $f_{IH}$ = 32 MHz | $f_{IH}$ = 48 MHz | $f_{IH}$ = 64 MHz |
| 0 | 0 | 1 | $f_{IH}$ = 12 MHz | $f_{IH}$ = 16 MHz | $f_{IH}$ = 24 MHz | $f_{IH}$ = 32 MHz |
| 0 | 1 | 0 | $f_{IH}$ = 6 MHz | $f_{IH}$ = 8 MHz | $f_{IH}$ = 12 MHz | $f_{IH}$ = 16 MHz |
| 0 | 1 | 1 | $f_{IH}$ = 3 MHz | $f_{IH}$ = 4 MHz | $f_{IH}$ = 6 MHz | $f_{IH}$ = 8 MHz |
| 1 | 0 | 0 | Setting prohibited | $f_{IH}$ = 2 MHz | Setting prohibited | $f_{IH}$ = 4 MHz |
| 1 | 0 | 1 | Setting prohibited | $f_{IH}$ = 1 MHz | Setting prohibited | $f_{IH}$ = 2 MHz |
| Other than above | | | Setting prohibited | | | |

### 5.6.2 Example of Setting X1 Oscillator

After a reset release, the CPU/peripheral hardware clock ($f_{CLK}$) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the X1 oscillation clock, set the oscillator and start oscillation by using the clock operation mode control register (CMC) and clock operation status control register (CSC) and wait for oscillation to stabilize by using the oscillation stabilization time counter status register (OSTC). After the oscillation stabilizes, set the X1 oscillation clock to $f_{CLK}$ by using the system clock control register (CKC).

[Register settings] Set the registers in the following order.

<1> Set the OSCSEL bit of the CMC register to 1, except for the cases where the frequency is equal or more than 10MHz, in such cases set the AMPH bit to 1, to operate the X1 oscillator.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CMC | EXCLK | OSCSEL | EXCLKS | OSCSELS | | AMPHS1 | AMPHS0 | AMPH |
| | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

<2> Using the OSTS register, select the oscillation stabilization time of the X1 oscillator at releasing of the STOP mode.
Example: Setting values when a wait of at least 102.4 $\mu$s is set based on a 10 MHz resonator.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSTS | | | | | | OSTS2 | OSTS1 | OSTS0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

<3> Clear (0) the MSTOP bit of the CSC register to start oscillating the X1 oscillator.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CSC | MSTOP | XTSTOP | | | | | | HIOSTOP |
| | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

<4> Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize.
Example: Wait until the bits reach the following values when a wait of at least 102.4 $\mu$s is set based on a 10 MHz resonator.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSTC | MOST8 | MOST9 | MOST10 | MOST11 | MOST13 | MOST15 | MOST17 | MOST18 |
| | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

<5> Use the MCM0 bit of the CKC register to specify the X1 oscillation clock as the CPU/peripheral hardware clock.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CKC | CLS | CSS | MCS | MCM0 | | | | |
| | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

<6> Use the MCS bit of the CKC register to confirm that $f_{MX}$ (X1 oscillation clock) is selected as the CPU/peripheral hardware clock (MCS = 1).

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CKC | CLS | CSS | MCS | MCM0 | | | | |
| | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

### 5.6.3 Example of Setting XT1 Oscillator

After a reset release, the CPU/peripheral hardware clock ($f_{CLK}$) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the XT1 oscillation clock, set the oscillator and start oscillation by using the operation speed mode control register (OSMC), clock select register (CKSEL), clock operation mode control register (CMC), and clock operation status control register (CSC), set the XT1 oscillation clock to $f_{CLK}$ by using the system clock control register (CKC).

[Register settings] Set the registers in the following order.

<1> The RTCLPC bit in the OSMC register can be used to enable or disable supply of the clock to the peripheral functions in STOP mode or HALT mode while sub/low-speed on-chip oscillator selection clock is selected as CPU clock.
When RTCLPC = 0, the supply of the subsystem/low-speed on-chip oscillator select clock to peripheral functions is enabled. When RTCLPC = 1, the supply of the subsystem/low-speed on-chip oscillator select clock to peripheral functions except for the real-time clock is stopped.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSMC | RTCLPC<br>0/1 | 0 | 0 | WUTMMCK0<br>0 | 0 | 0 | 0 | 0 |

<2> Select $f_{SUB}$ with the SELLOSC bit of the CKSEL register.
Clear the SELLOSC bit to 0 to set $f_{SL}$ to the XT1 oscillation clock.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CKSEL | 0 | 0 | 0 | 0 | 0 | TRD_CKSEL<br>0 | 0 | SELLOSC<br>0 |

<3> Select the operation mode of the subsystem clock with the OSCSELS bit of the CMC register.
Set the OSCSELS bit to 1 to select the XT1 oscillation mode or external clock input mode.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CMC | EXCLK<br>0 | OSCSEL<br>0 | EXCLKS<br>0 | OSCSELS<br>1 | 0 | AMPHS1<br>0/1 | AMPHS0<br>0/1 | AMPH<br>0 |

AMPHS0 and AMPHS1 bits: These bits are used to specify the oscillation mode of the XT1 oscillator.

<4> Clear the XTSTOP bit of the CSC register to 0 to start oscillating the XT1 oscillator.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CSC | MSTOP<br>0 | XTSTOP<br>0 | 0 | 0 | 0 | 0 | 0 | HIOSTOP<br>0 |

<5> Use the timer function or another function to wait for oscillation of the subsystem clock to stabilize by using software.

<6> Select the CPU/peripheral hardware clock with the CSS bit of the CKC register.
Set the CSS bit to 1 to specify CPU clock = $f_{SL}$ (XT1 oscillation clock).

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CKC | CLS<br>0 | CSS<br>1 | MCS<br>0 | MCM0<br>0 | 0 | 0 | 0 | 0 |

<7> Confirm that $f_{SL}$ (XT1 oscillation clock) is selected as the CPU/peripheral hardware clock (CLS = 1) with the CLS bit of the CKC register.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CKC | CLS<br>1 | CSS<br>1 | MCS<br>0 | MCM0<br>0 | 0 | 0 | 0 | 0 |

### 5.6.4 Examples of Setting PLL Circuit

The following PLL setting procedures are described here.

- Oscillating the PLL clock and setting it as the CPU clock
- Stopping the PLL clock

[Register settings] Set the registers in the following order.

**(1) Example of procedure for setting oscillation of PLL clock**

<1> Select the frequency of the PLL output clock with the PLLDIV1 bit of the PLLCTL register.

When PLL clock ≤ 32 MHz, clear the PLLDIV1 bit to 0. When PLL clock > 32 MHz, set the PLLDIV1 bit to 1.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PLLCTL | LCKSEL1<br>0/1 | LCKSEL0<br>0/1 | PLLDIV1<br>0/1 | PLLDIV0<br>0/1 | <br>0 | SELPLL<br>0 | PLLMUL<br>0/1 | PLLON<br>0 |

<2> Set the PLL lock-up wait counter with the LCKSEL1 and LCKSEL0 bits of the PLLCTL register.

The counter for the PLL lock-up wait time is set to a period of at least 40 μs. When the PLL source clock ($f_{MAIN}$) is 4 MHz, set the LCKSEL1 and LCKSEL0 bits to 01 or 10. When the PLL source clock ($f_{MAIN}$) is 8 MHz, set the LCKSEL1 and LCKSEL0 bits to 10.

<3> Select the frequency division of the PLL clock with the PLLDIV0 bit of the PLLCTL register.

When PLLDIV0 = 0, the PLL division ratio is 2. When PLLDIV0 = 1, the PLL division ratio is 4.

<4> Select the multiplication value of the PLL clock with the PLLMUL bit of the PLLCTL register.

When PLLMUL = 0, the PLL multiplication value is 12. When PLLMUL = 1, the PLL multiplication value is 16.

<5> Wait for the selection of the PLL multiplication value to become effective. After setting the PLLMUL bit, wait for at least 1 μs.

<6> Set the PLLON bit of the PLLCTL register to 1 to start oscillation of the PLL clock.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PLLCTL | LCKSEL1<br>0/1 | LCKSEL0<br>0/1 | PLLDIV1<br>0/1 | PLLDIV0<br>0/1 | <br>0 | SELPLL<br>0 | PLLMUL<br>0/1 | PLLON<br>1 |

<7> Confirm that the PLL circuit is locked (LOCK = 1) with the LOCK bit of the PLLSTS register.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PLLSTS | LOCK<br>1 | <br>0 | <br>0 | <br>0 | <br>0 | <br>0 | <br>0 | <br>0 |

<8> Set the PLL clock between 1 MHz and 32 MHz with the MDIV bits of the MDIV register.

Example: To select $f_{MP}/2$, set the following value.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| MDIV | <br>0 | <br>0 | <br>0 | <br>0 | <br>0 | MDIV2<br>0 | MDIV1<br>0 | MDIV0<br>1 |

<9> Select the PLL clock mode with the SELPLL bit of the PLLCTL register.

Set the SELPLL bit to 1 to select the PLL-clock-selected mode ($f_{MP} = f_{PLL}$).

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PLLCTL | LCKSEL1<br>0/1 | LCKSEL0<br>0/1 | PLLDIV1<br>0/1 | PLLDIV0<br>0/1 | <br>0 | SELPLL<br>1 | PLLMUL<br>0/1 | PLLON<br>1 |

<10> Confirm that the PLL-clock-selected mode is selected (SELPLLS = 1) with the SELPLLS bit of the PLLSTS register.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PLLSTS | LOCK | | | | SELPLLS | | | |
| | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

**(2) Examples of procedure for stopping PLL clock**

There is the following method to stop the PLL clock.

- Set the PLLON bit to 0 to stop the PLL clock.

<1> Select the PLL clock mode with the SELPLL bit of the PLLCTL register.

Clear the SELPLL bit to 0 to select the clock through mode ($f_{PLL} = f_{MAIN}$).

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PLLCTL | LCKSEL1 | LCKSEL0 | PLLDIV1 | PLLDIV0 | | SELPLL | PLLMUL | PLLON |
| | 0/1 | 0/1 | 0/1 | 0/1 | 0 | 0 | 0/1 | 1 |

<2> Confirm that the clock through mode is selected (SELPLLS = 0) with the SELPLLS bit of the PLLSTS register.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PLLSTS | LOCK | | | | SELPLLS | | | |
| | 0/1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

<3> Clear the PLLON bit of the PLLCTL register to 0 to stop oscillation of the PLL clock.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PLLCTL | LCKSEL1 | LCKSEL0 | PLLDIV1 | PLLDIV0 | | SELPLL | PLLMUL | PLLON |
| | 0/1 | 0/1 | 0/1 | 0/1 | 0 | 0 | 0/1 | 0 |

**(3) Caution when restarting the PLL clock after being stopped**

In a case of restarting the PLL clock after it has been stopped, wait for at least 4 μs after the PLL circuit was stopped before restarting operation.

### 5.6.5 Example of Setting Low-Speed On-Chip Oscillator

An example of setting the low-speed on-chip oscillator as the CPU clock is shown below.

<1> Select $f_{IL}$ with the SELLOSC bit of the CKSEL register.

Set the SELLOSC bit to 1 to set $f_{SL}$ for the low-speed on-chip oscillator.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CKSEL | | | | | | TRD_CKSEL | | SELLOSC |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

<2> Select the operation mode of the subsystem clock with the OSCSELS bit of the CMC register.

Set the OSCSELS bit to 1 to select the input port mode (low-speed on-chip oscillator operation mode).

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CMC | EXCLK | OSCSEL | EXCLKS | OSCSELS | | AMPHS1 | AMPHS0 | AMPH |
| | 0 | 0 | 0 | 1 | 0 | 0/1 | 0/1 | 0 |

<3> Select the CPU/peripheral hardware clock with the CSS bit of the CKC register.

Set the CSS bit to 1 to specify CPU clock = $f_{SL}$ (low-speed on-chip oscillator).

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CKC | CLS | CSS | MCS | MCM0 | | | | |
| | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

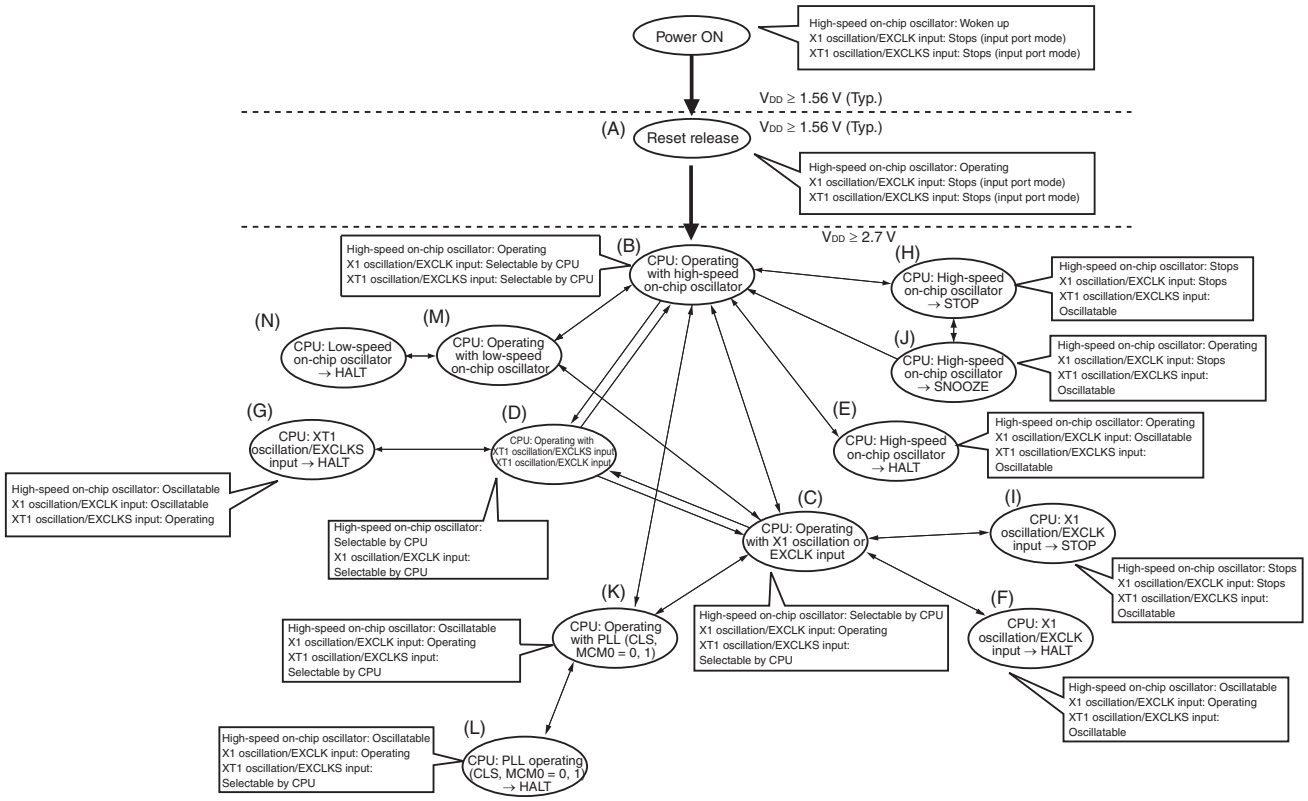
<4> Confirm that $f_{SL}$ (low-speed on-chip oscillator) is selected as the CPU/peripheral hardware clock (CLS = 1) with the CLS bit of the CKC register.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CKC | CLS | CSS | MCS | MCM0 | | | | |
| | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

### 5.6.6 CPU Clock Status Transition Diagram

Figure 5-27 shows the CPU clock status transition diagram of this product.

**Figure 5-27. CPU Clock Status Transition Diagram**




**Caution Transitions in the order of (B) → (D) → (C) or (C) → (D) → (B) are prohibited.**

The following shows an example of changing the CPU clock and setting the SFR register.

**(1) After reset release (A), change the CPU to operating with the high-speed on-chip oscillator clock (B).**

(A)→(B): Setting the SFR register is not required (initial status after reset release).

**(2) Change the CPU from operating with the high-speed on-chip oscillator clock (B) to operating with the high-speed system clock (C).**

- Set the CMC register (EXCLK = 0, OSCSEL = 1, AMPH = x). **Note 1**
- Set the OSTS register. **Note 2**
- Set the MSTOP bit of the CSC register to 0.
- Check the oscillation stabilization time by using the OSTC register. **Note 2**
- Set the MCM0 bit of the CKC register to 1.
- Set the MCS bit of the CKC register to 1.

**Notes 1.** The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release.

**2.** Set the oscillation stabilization time of the oscillation stabilization time select register (OSTS) as shown below: OSTS register setting value ≥ Expected oscillation stabilization time counter status register (OSTC)

**(3) Change the CPU from operating with the high-speed on-chip oscillator clock (B) or operating with the high-speed system clock (C) to operating with the subsystem clock (D).**

- Set the RTCLPC bit of the OSMC register.
- Set the SELLOSC bit of the CKSEL register to 0.
- Set the CMC register (EXCLKS = x, OSCSELS = 1, AMPHS[1:0] = xx). **Note**
- Set the XTSTOP bit of the CSC register to 0.
- Wait for oscillation stabilization.
- Set the CSS bit of the CKC register to 1.
- Confirm that the CLS bit of the CKC register is set to 1.

**Note** The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release.

**(4) Change the CPU from operating with the high-speed on-chip oscillator clock (B) or operating with the high-speed system clock (C) to operating with the low-speed on-chip oscillator clock (M).**

- Set the SELLOSC bit of the CKSEL register to 1.
- Set the CMC register (EXCLKS = x, OSCSELS = 1). **Note**
- Set the CSS bit of the CKC register to 1.
- Confirm that the CLS bit of the CKC register is set to 1.

**Note** The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release.

**(5) Change the CPU from operating with the high-speed on-chip oscillator clock (B) or operating with the high-speed system clock (C) to operating with the PLL clock (K).**

- Set the PLLCTL register (PLLDIV1 = x, LCKSEL[1:0] = xx, PLLDIV0 = x, PLLMUL = x).
- Wait for the selection of the PLL multiplication value to become effective (After setting the PLLMUL bit, wait for at least 1 μs).
- Set the PLLON bit of the PLLCTL register to 1.
- Confirm that the LOCK bit of the PLLSTS register is set to 1 (checking PLL locked state).
- Set the MDIV [2:0] bits of the MDIV register.
- Set the SELPLL bit of the PLLCTL register to 1.
- Confirm that the SELPLLS bit of the PLLSTS register is set to 1.

**(6) Change the CPU from operating with the high-speed system clock (C) to operating with the high-speed on-chip oscillator clock (B).**

- Set the HIOSTOP bit of the CSC register to 0. **Note**
- Set the MCM0 bit of the CKC register to 0.
- Confirm that the MCS bit of the CKC register is set to 0.

**Note** When oscillation starts from a high-speed on-chip oscillator clock stop state (HIOSTOP = 1), have the software wait for the following oscillation accuracy stabilization time, and then change the clock.
FRQSEL4 of the user option byte (000C2H/020C2H) = 0: 18 μs to 65 μs
FRQSEL4 of the user option byte (000C2H/020C2H) = 1: 18 μs to 105 μs

**(7) Change the CPU from operating with the subsystem clock (D) or operating with the low-speed on-chip oscillator clock (M) to operating with the high-speed on-chip oscillator clock (B).**

- Set the HIOSTOP bit of the CSC register to 0. Note
- Set the CSS bit of the CKC register to 0.
- Confirm that the CLS bit of the CKC register is set to 0.

**Note** When oscillation starts from a high-speed on-chip oscillator clock stop state (HIOSTOP = 1), have the software wait for the following oscillation accuracy stabilization time, and then change the clock.
FRQSEL4 of the user option byte (000C2H/020C2H) = 0: 18 μs to 65 μs
FRQSEL4 of the user option byte (000C2H/020C2H) = 1: 18 μs to 105 μs

**(8) Change the CPU from operating with the PLL clock (K) to operating with the high-speed system clock (C) or operating with the high-speed on-chip oscillator clock (B).**

- Set the SELPLL bit of the PLLCTL register to 0.
- Confirm that the SELPLLS bit of the PLLSTS register is set to 0.

**(9) Change the CPU from operating with the subsystem clock (D) or operating with the low-speed on-chip oscillator clock (M) to operating with the high-speed system clock (C).**

- Set the CMC register (EXCLK = 0, OSCSEL = 1, AMPH = x). Note 1
- Set the OSTS register. Note 2
- Check the oscillation stabilization time by using the OSTC register. Note 2
- Set the CSS bit of the CKC register to 0.
- Confirm that the CLS bit of the CKC register is set to 0.

**Notes 1.** The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release.

**2.** Set the oscillation stabilization time of the oscillation stabilization time select register (OSTS) as shown below:
OSTS register setting value > Expected oscillation stabilization time counter status register (OSTC)

**(10) Change the CPU from each operation mode to HALT mode.**

- The CPU changes from operating with the high-speed on-chip oscillator clock (B) to HALT mode (E).
- The CPU changes from operating with the high-speed system clock (C) to HALT mode (F).
- The CPU changes from operating with the subsystem clock (D) to HALT mode (G).
- The CPU changes from operating with the PLL clock (K) to HALT mode (L).
- The CPU changes from operating with the low-speed on-chip oscillator clock (M) to HALT mode (N).
  - Execute the HALT instruction.

**(11) The CPU changes from operating with the high-speed on-chip oscillator clock (B) to STOP mode (H).**

- Stop peripheral functions that are not operated in STOP mode.
- Execute the STOP instruction.

**(12) The CPU changes from operating with the high-speed system clock (C) to STOP mode (I).**

- Stop peripheral functions that are not operated in STOP mode.
- Set the OSTS register. Note
- Execute the STOP instruction.

**Note** Set the oscillation stabilization time of the oscillation stabilization time select register (OSTS) as shown below:
OSTS register setting value > Expected oscillation stabilization time counter status register (OSTC)

**(13) Change the CPU from STOP mode (H) to SNOOZE mode (J).**

For details about the settings for entering SNOOZE mode, see 23.3.3 SNOOZE Mode and peripheral functions that are used.

**Remarks 1**. "x" shown in the settings of the SFR register for changing each mode represents an arbitrary value (the settings to be used).

**2**. For details about transition and recovery to the standby function (HALT mode, STOP mode, and SNOOZE mode), see **CHAPTER 23 STANDBY FUNCTION** and peripheral functions that are used.

### 5.6.7 Conditions before Changing CPU Clock and Processing after Changing CPU Clock

The following table shows the conditions before changing the CPU clock and the processing after changing the CPU clock.

**Table 5-3. Changing CPU Clock (1/7)**

| CPU Clock | | Conditions before Change | Processing after Change |
|---|---|---|---|
| Before Change | After Change | | |
| High-speed on-chip oscillator clock | X1 clock | X1 oscillation is stable.<br>• OSCSEL = 1, EXCLK = 0, MSTOP = 0<br>• After elapse of oscillation stabilization time | Stopping the high-speed on-chip oscillator (HIOSTOP = 1) can reduce the operating current. |
| | External main system clock | External clock input from the EXCLK pin is enabled.<br>• OSCSEL = 1, EXCLK = 1, MSTOP = 0 | |
| | XT1 clock | XT1 oscillation is stable, and the subsystem clock is selected as the subsystem/low-speed on-chip oscillator select clock.<br>• OSCSELS = 1, EXCLKS = 0, XTSTOP = 0<br>• SELLOSC = 0<br>• After elapse of oscillation stabilization time | |
| | External subsystem clock | External clock input from the EXCLKS pin is enabled, and the subsystem clock is selected as the subsystem/low-speed on-chip oscillator select clock.<br>• OSCSELS = 1, EXCLKS = 1, XTSTOP = 0<br>• SELLOSC = 0 | |
| | Low-speed on-chip oscillator clock | The low-speed on-chip oscillator starts oscillation, and the low-speed on-chip oscillator clock is selected as the subsystem/low-speed on-chip oscillator select clock.<br>• OSCSELS = 1, SELLOSC = 1 | |
| | PLL clock | PLL oscillation is stable.<br>• LOCK = 1, PLLON = 1 | The high-speed on-chip oscillator cannot be stopped because it is the PLL input clock. |

**Remark** For details about the register flag settings for stopping the target clock during the processing after change and conditions before the clock is stopped, see **5.6.9 Conditions Before Clock Oscillation Is Stopped**.

**Table 5-3. Changing CPU Clock (2/7)**

| CPU Clock | | Conditions before Change | Processing after Change |
|---|---|---|---|
| Before Change | After Change | | |
| X1 clock | High-speed on-chip oscillator clock | The high-speed on-chip oscillator starts oscillation.<br>• HIOSTOP = 0 | X1 oscillation can be stopped.<br>(MSTOP = 1) |
| | External main system clock | Prohibited to change. (To change the CPU clock, clear the settings first and then reset the settings.) | — |
| | XT1 clock | XT1 oscillation is stable, and the subsystem clock is selected as the subsystem/low-speed on-chip oscillator select clock.<br>• OSCSELS = 1, EXCLKS = 0, XTSTOP = 0<br>• SELLOSC = 0<br>• After elapse of oscillation stabilization time | X1 oscillation can be stopped.<br>(MSTOP = 1) |
| | External subsystem clock | External clock input from the EXCLKS pin is enabled, and the subsystem clock is selected as the subsystem/low-speed on-chip oscillator select clock.<br>• OSCSELS = 1, EXCLKS = 1, XTSTOP = 0<br>• SELLOSC = 0 | |
| | Low-speed on-chip oscillator clock | The low-speed on-chip oscillator starts oscillation, and the low-speed on-chip oscillator clock is selected as the subsystem/low-speed on-chip oscillator select clock.<br>• OSCSELS = 1, SELLOSC = 1 | |
| | PLL clock | PLL oscillation is stable.<br>• LOCK = 1, PLLON = 1 | The X1 clock cannot be stopped because it is the PLL input clock. |

**Remark** For details about the register flag settings for stopping the target clock during the processing after change and conditions before the clock is stopped, see **5.6.9 Conditions Before Clock Oscillation Is Stopped**.

**Table 5-3. Changing CPU Clock (3/7)**

| CPU Clock | | Conditions before Change | Processing after Change |
|---|---|---|---|
| Before Change | After Change | | |
| External main system clock | High-speed on-chip oscillator clock | The high-speed on-chip oscillator starts oscillation.<br>• HIOSTOP = 0 | The external main system clock input can be disabled.<br>(MSTOP = 1) |
| | X1 clock | Prohibited to change. (To change the CPU clock, clear the settings first and then reset the settings.) | — |
| | XT1 clock | XT1 oscillation is stable, and the subsystem clock is selected as the subsystem/low-speed on-chip oscillator select clock.<br>• OSCSELS = 1, EXCLKS = 0, XTSTOP = 0<br>• SELLOSC = 0<br>• After elapse of oscillation stabilization time | The external main system clock input can be disabled.<br>(MSTOP = 1) |
| | External subsystem clock | External clock input from the EXCLKS pin is enabled, and the subsystem clock is selected as the subsystem/low-speed on-chip oscillator select clock.<br>• OSCSELS = 1, EXCLKS = 1, XTSTOP = 0<br>• SELLOSC = 0 | |
| | Low-speed on-chip oscillator clock | The low-speed on-chip oscillator starts oscillation, and the low-speed on-chip oscillator clock is selected as the subsystem/low-speed on-chip oscillator select clock.<br>• OSCSELS = 1, SELLOSC = 1 | |
| | PLL clock | PLL oscillation is stable.<br>• LOCK = 1, PLLON = 1 | The external main system clock cannot be stopped because it is the PLL input clock. |

**Remark** For details about the register flag settings for stopping the target clock during the processing after change and conditions before the clock is stopped, see **5.6.9 Conditions Before Clock Oscillation Is Stopped**.

**Table 5-3. Changing CPU Clock (4/7)**

| CPU Clock | | Conditions before Change | Processing after Change |
|---|---|---|---|
| Before Change | After Change | | |
| XT1 clock | High-speed on-chip oscillator clock | The high-speed on-chip oscillator starts oscillation, and the high-speed on-chip oscillator clock is selected as the main system clock.<br>• HIOSTOP = 0, MCS = 0 | XT1 oscillation can be stopped.<br>(XTSTOP = 1) |
| | X1 clock | X1 oscillation is stable, and the high-speed system clock is selected as the main system clock.<br>• OSCSEL = 1, EXCLK = 0, MSTOP = 0<br>• After elapse of oscillation stabilization time<br>• MCS = 1 | |
| | External main system clock | External clock input from the EXCLK pin is enabled, and the high-speed system clock is selected as the main system clock.<br>• OSCSEL = 1, EXCLK = 1, MSTOP = 0<br>• SELLOSC = 0, MCS = 1 | |
| | External subsystem clock | Prohibited to change. (To change the CPU clock, clear the settings first and then reset the settings.) | — |
| | Low-speed on-chip oscillator clock | Prohibited to change. (To change the CPU clock, specify the main system/PLL select clock as the CPU clock first and then reset the settings.) | |

**Remark** For details about the register flag settings for stopping the target clock during the processing after change and conditions before the clock is stopped, see **5.6.9 Conditions Before Clock Oscillation Is Stopped**.

**Table 5-3. Changing CPU Clock (5/7)**

| CPU Clock | | Conditions before Change | Processing after Change |
|---|---|---|---|
| Before Change | After Change | | |
| External subsystem clock | High-speed on-chip oscillator clock | The high-speed on-chip oscillator starts oscillation, and the high-speed on-chip oscillator clock is selected as the main system clock.<br>• HIOSTOP = 0, MCS = 0 | The external subsystem clock input can be disabled.<br>(XTSTOP = 1) |
| | X1 clock | X1 oscillation is stable, and the high-speed system clock is selected as the main system clock.<br>• OSCSEL = 1, EXCLK = 0, MSTOP = 0<br>• After elapse of oscillation stabilization time<br>• MCS = 1 | |
| | External main system clock | External clock input from the EXCLK pin is enabled, and the high-speed system clock is selected as the main system clock.<br>• OSCSEL = 1, EXCLK = 1, MSTOP = 0<br>• SELLOSC = 0, MCS = 1 | |
| | XT1 clock | Prohibited to change. (To change the CPU clock, clear the settings first and then reset the settings.) | — |
| | Low-speed on-chip oscillator clock | Prohibited to change. (To change the CPU clock, specify the main system/PLL select clock as the CPU clock first and then reset the settings.) | |

**Remark** For details about the register flag settings for stopping the target clock during the processing after change and conditions before the clock is stopped, see **5.6.9 Conditions Before Clock Oscillation Is Stopped**.

**Table 5-3. Changing CPU Clock (6/7)**

| CPU Clock | | Conditions before Change | Processing after Change |
|---|---|---|---|
| Before Change | After Change | | |
| Low-speed on-chip oscillator clock | High-speed on-chip oscillator clock | The high-speed on-chip oscillator starts oscillation, and the high-speed on-chip oscillator clock is selected as the main system clock.<br>• HIOSTOP = 0, MCS = 0 | The low-speed on-chip oscillator can be stopped.<br>(SELLOSC = 0, WUTMMCK0 = 0) |
| | X1 clock | X1 oscillation is stable, and the high-speed system clock is selected as the main system clock.<br>• OSCSEL = 1, EXCLK = 0, MSTOP = 0<br>• After elapse of oscillation stabilization time<br>• MCS = 1 | |
| | External main system clock | External clock input from the EXCLK pin is enabled, and the high-speed system clock is selected as the main system clock.<br>• OSCSEL = 1, EXCLK = 1, MSTOP = 0<br>• SELLOSC = 0, MCS = 1 | |
| | XT1 clock | Prohibited to change. (To change the CPU clock, specify the main system/PLL select clock as the CPU clock first and then reset the settings.) | — |
| | External subsystem clock | Prohibited to change. (To change the CPU clock, specify the main system/PLL select clock as the CPU clock first and then reset the settings.) | |

**Remark** For details about the register flag settings for stopping the target clock during the processing after change and conditions before the clock is stopped, see **5.6.9 Conditions Before Clock Oscillation Is Stopped**.

**Table 5-3. Changing CPU Clock (7/7)**

| CPU Clock | | Conditions before Change | Processing after Change |
|---|---|---|---|
| Before Change | After Change | | |
| PLL clock | High-speed on-chip oscillator clock | The high-speed on-chip oscillator starts oscillation, and the high-speed on-chip oscillator clock is selected as the main system clock.<br>• HIOSTOP = 0, MCS = 0 | The PLL clock can be stopped.<br>(PLLON = 0) |
| | X1 clock | X1 oscillation is stable, and the high-speed system clock is selected as the main system clock.<br>• OSCSEL = 1, EXCLK = 0, MSTOP = 0<br>• After elapse of oscillation stabilization time<br>• MCS = 1 | |
| | External main system clock | External clock input from the EXCLK pin is enabled, and the high-speed system clock is selected as the main system clock.<br>• OSCSEL = 1, EXCLK = 1, MSTOP = 0<br>• SELLOSC = 0, MCS = 1 | |

**Remark** For details about the register flag settings for stopping the target clock during the processing after change and conditions before the clock is stopped, see **5.6.9 Conditions Before Clock Oscillation Is Stopped**.

### 5.6.8 Time Required for Switchover of CPU Clock, Main System/PLL Select Clock, and Main System Clock

By setting bits 4 and 6 (MCM0, CSS) of the system clock control register (CKC), bits 0 to 2 (MDIV0 to MDIV2) of the $f_{MP}$ clock division register (MDIV), and bit 0 (SELLOSC) of the clock select register (CKSEL), the CPU clock can be switched (between the main system/PLL select clock and the subsystem/low-speed on-chip oscillator select clock), the main system/PLL select clock can be switched (between the main system clock and the PLL clock), the main system clock can be switched (between the high-speed on-chip oscillator clock and the high-speed system clock), the subsystem/low-speed on-chip oscillator select clock can be switched (between the subsystem clock and the low-speed on-chip oscillator clock), and the frequency division ratio of the main system/PLL select clock can be changed.

The actual switchover operation is not performed immediately after rewriting to the CKC or MDIV register; operation continues on the pre-switchover clock for several clocks. The subsystem/low-speed on-chip oscillator select clock is switched immediately after rewriting to the CKSEL register.

Whether the CPU is operating on the main system/PLL select clock or the subsystem/low-speed on-chip oscillator select clock can be ascertained using bit 7 (CLS) of the CKC register.

Whether the main system/PLL select clock is operating on the main system clock or the PLL clock can be ascertained using bit 3 (SELPLLS) of the PLL status register (PLLSTS).

Whether the main system clock is operating on the high-speed on-chip oscillator clock or the high-speed system clock can be ascertained using bit 5 (MCS) of the CKC register.

When the CPU clock is switched, the peripheral hardware clock is also switched.

**Table 5-4. Maximum Time Required for Clock Switchover**

| Clock A | Switching directions | Clock B | Type |
|---|---|---|---|
| $f_{MP}$ | ⟷ (change of the frequency division ratio) | $f_{MP}$ | Type 1 (Table 5-5) |
| $f_{IH}$ | ⟷ | $f_{MX}$ | Type 2 (Table 5-6) |
| $f_{MP}$ | ⟷ | $f_{SL}$ | Type 3 (Table 5-7) |
| $f_{MAIN}$ | ⟷ | $f_{PLL}$ | Type 4 (Table 5-8) |

**Table 5-5. Maximum Time Required for Type 1**

| Set Value Before Switchover | Set Value After Switchover | |
|---|---|---|
| | Clock A | Clock B |
| Clock A | | 1 + $f_A/f_B$ clock |
| Clock B | 1 + $f_B/f_A$ clock | |

**Table 5-6. Maximum Time Required for Type 2 (1)[Note1]**

| Set Value Before Switchover | | Set Value After Switchover | |
|---|---|---|---|
| MCM0 | | MCM0 | |
| | | 0 ($f_{MAIN} = f_{IH}$) | 1 ($f_{MAIN} = f_{MX}$) |
| 0 ($f_{MAIN} = f_{IH}$) | $f_{MX} \geq f_{IH}$ | | 3 clocks |
| | $f_{MX} < f_{IH}$ | | 3 $f_{IH}/f_{MX}$ clock |
| 1 ($f_{MAIN} = f_{MX}$) | $f_{MX} > f_{IH}$ | 3 $f_{MX}/f_{IH}$ clock | |
| | $f_{MX} \leq f_{IH}$ | 3 clocks | |

**Note 1.** For $f_{IH} \leq 32$ MHz

(**Remarks** are listed on the next page.)

**Table 5-6. Maximum Time Required for Type 2 (2)[Note 1]**

| Set Value Before Switchover | Set Value After Switchover | |
|---|---|---|
| MCM0 | MCM0 | |
| | 0 ($f_{MAIN} = f_{IH}$) | 1 ($f_{MAIN} = f_{MX}$) |
| 0 ($f_{MAIN} = f_{IH}$) | | 6 $f_{IH}/f_{MX}$ clock |
| 1 ($f_{MAIN} = f_{MX}$) | 3 clocks | |

**Note 1.** For $f_{IH} > 32$ MHz

**Table 5-7. Maximum Time Required for Type 3**

| Set Value Before Switchover | Set Value After Switchover | |
|---|---|---|
| CSS | CSS | |
| | 0 ($f_{CLK} = f_{MP}$) | 1 ($f_{CLK} = f_{SL}$) |
| 0 ($f_{CLK} = f_{MP}$) | | 1 + 2 $f_{MP}/f_{SL}$ clock |
| 1 ($f_{CLK} = f_{SL}$) | 3 clocks | |

**Table 5-8. Maximum Time Required for Type 4**

| Set Value Before Switchover | Set Value After Switchover | |
|---|---|---|
| SELPLL | SELPLL | |
| | 0 ($f_{MP} = f_{MAIN}$) | 1 ($f_{MP} = f_{PLL}$) |
| 0 ($f_{MP} = f_{MAIN}$) | | 2 clocks |
| 1 ($f_{MP} = f_{PLL}$) | 2 $f_{PLL}/f_{MAIN}$ clock | |

**Remarks 1.** The number of clocks listed in Tables 5-5 to 5-8 is the number of CPU clocks before switchover.

**2.** Calculate the number of clocks in Tables 5-5 to 5-8 by removing the decimal portion.

**Example** When switching the main system clock from the high-speed on-chip oscillator clock (with 16 MHz) to the high-speed system clock
(@ oscillation with $f_{IH}$ = 16 MHz, $f_{MX}$ = 10 MHz)
3 $f_{IH}/f_{MX}$ = 3 × 1.6 = 4.8 → 5 clocks

### 5.6.9 Conditions Before Clock Oscillation Is Stopped

The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped.

**Table 5-9. Conditions Before the Clock Oscillation Is Stopped and Flag Settings**

| Clock | Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled) | Flag Settings of SFR Register |
|---|---|---|
| High-speed on-chip oscillator clock | MCS = 1 or CLS = 1<br>(The CPU is operating on a clock other than the high-speed on-chip oscillator clock.) | HIOSTOP = 1 |
| X1 clock<br>External main system clock | MCS = 0 or CLS = 1<br>(The CPU is operating on a clock other than the high-speed system clock.) | MSTOP = 1 |
| PLL clock | SELPLLS = 0<br>(The CPU is operating on a clock other than the PLL clock.) | PLLON = 0 |
| XT1 clock<br>External subsystem clock | CLS = 0<br>(The CPU is operating on a clock other than the subsystem/low-speed on-chip oscillator clock.) | XTSTOP = 1 |
| Low-speed on-chip oscillator clock | CLS = 0<br>(The CPU is operating on a clock other than the subsystem/low-speed on-chip oscillator clock.) | SELLOSC = 0 and<br>WUTMMCK0 = 0 |

**Remark** MCS: Bit 5 of the system clock control register (CKC)
CLS: Bit 7 of the system clock control register (CKC)
HIOSTOP: Bit 0 of the clock operation status control register (CSC)
XTSTOP: Bit 6 of the clock operation status control register (CSC)
MSTOP: Bit 7 of the clock operation status control register (CSC)
SELPLLS: Bit 3 of the PLL status register (PLLSTS)
PLLON: Bit 0 of the PLL control register (PLLCTL)
SELLOSC: Bit 1 of the clock select register (CKSEL)
WUTMMCK0: Bit 4 of the operation speed mode control register (OSMC)

## 5.7 Usage Notes

### 5.7.1 CPU/Peripheral Hardware Clock

The clock set by the CSS, MCM0, SELPLL, and MDIV2 to MDIV0 bits is supplied to the CPU and peripheral hardware modules. If the CPU clock is changed, the clock supplied to the peripheral hardware modules is simultaneously changed. Therefore, when changing the CPU/peripheral hardware clock, operation of the peripheral hardware modules needs to be stopped before the change.

### 5.7.2 High-Speed On-Chip Oscillator

<R> When the FRQSEL3 bit is set to 0 (high-speed on-chip oscillator = 48/24/12/6/3 MHz), and moreover the CPU/peripheral hardware clock is selected as the PLL clock, the CPU/peripheral hardware clock frequency ($f_{CLK}$) must not be set to 32 MHz.

# CHAPTER 6 TIMER ARRAY UNIT

The number of units or channels of the timer array unit differs, depending on the product.

| Units | Channels | Group A | Groups B, C, and D | Group E |
|---|---|---|---|---|
| Unit 0 | Channel 0 | √ | √ | √ |
| | Channel 1 | √ | √ | √ |
| | Channel 2 | √ | √ | √ |
| | Channel 3 | √ | √ | √ |
| | Channel 4 | √ | √ | √ |
| | Channel 5 | √ | √ | √ |
| | Channel 6 | √ | √ | √ |
| | Channel 7 | √ | √ | √ |
| Unit 1 | Channel 0 | – | √ | √ |
| | Channel 1 | – | √ | √ |
| | Channel 2 | – | √ | √ |
| | Channel 3 | – | √ | √ |
| | Channel 4 | – | – | √ |
| | Channel 5 | – | – | √ |
| | Channel 6 | – | – | √ |
| | Channel 7 | – | – | √ |

**Remark** Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory

Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or RL78/F13 (LIN incorporated) products with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory

Group C: RL78/F13 (CAN incorporated) products with 30, 32, 48, 64, or 80 pins and 32 Kbytes to 128 Kbytes of code flash memory

Group D: RL78/F14 products with 30, 32, 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory

Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or RL78/F14 products with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

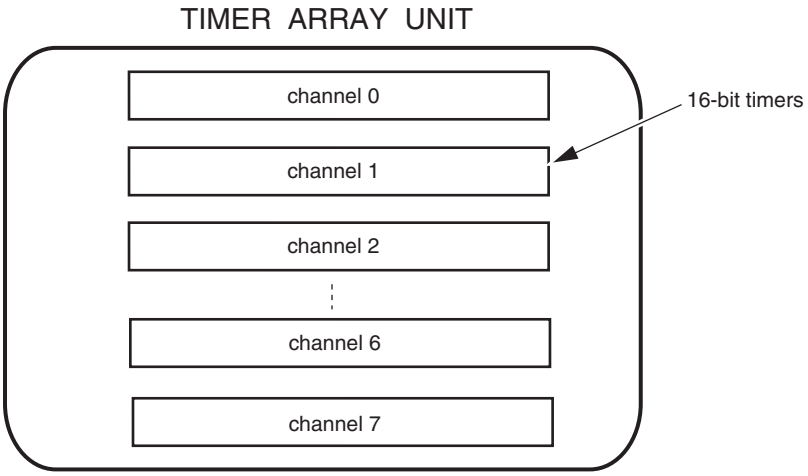
**Cautions**

**1. The presence or absence of timer I/O pins depends on the product. See Table 6-2 Timer I/O Pins provided in Each Product for details.**

**2. Most of the following descriptions in this chapter use the Group E products as an example.**

The timer array unit has eight 16-bit timers.

Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more “channels” can be used to create a high-accuracy timer.




For details about each function, see the table below.

| Independent channel operation function | Simultaneous channel operation function |
|---|---|
| • Interval timer (→ refer to 6.7.1)<br>• Square wave output (→ refer to 6.7.1)<br>• External event counter (→ refer to 6.7.2)<br>• Divider function (→ refer to 6.7.3)<br>• Input pulse interval measurement (→ refer to 6.7.4)<br>• Measurement of high-/low-level width of input signal (→ refer to 6.7.5)<br>• Delay counter (→ refer to 6.7.6) | • One-shot pulse output(→ refer to 6.8.1)<br>• PWM output(→ refer to 6.8.2)<br>• Multiple PWM output(→ refer to 6.8.3) |

It is possible to use the 16-bit timer of channels 1 and 3 of the units 0 and 1 as two 8-bit timers (higher and lower). The functions that can use channels 1 and 3 as 8-bit timers are as follows:

- Interval timer
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)

Channel 7 of timer array unit 0 can be used to realize LIN-bus communication operating in combination with UART0 of the serial array unit.

## 6.1 Functions of Timer Array Unit

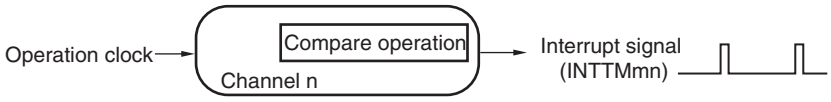
Timer array unit has the following functions.

### 6.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following purposes without being affected by the operation mode of other channels.

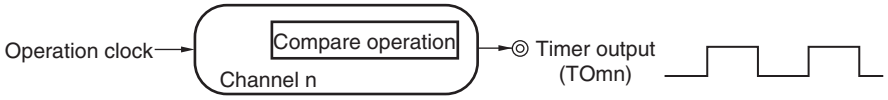
**(1) Interval timer**

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



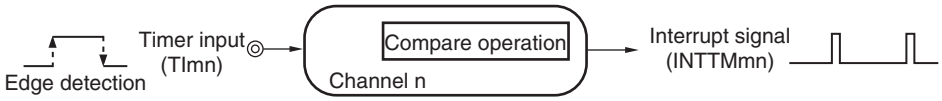

**(2) Square wave output**

A toggle operation is performed each time INTTMmn interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOmn).



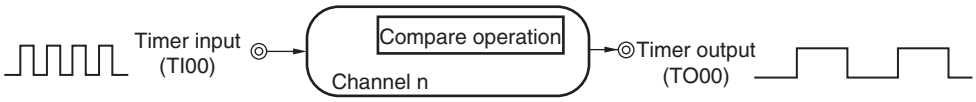

**(3) External event counter**

Each timer of a unit can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TImn) has reached a specific value.




**(4) Divider function**

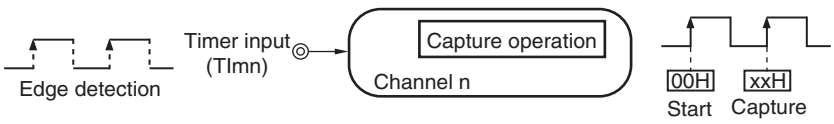
A clock input from a timer input pin (TImn) is divided and output from an output pin (TOmn).




Set the TImn and TOmn pins so that they are different from each other by the peripheral I/O redirection registers 0, 1, 2, and 3 (PIOR0, PIOR1, PIOPR2, and PIOR3).

**(5) Input pulse interval measurement**

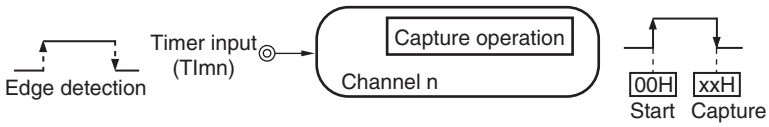
Counting is started by the valid edge of a pulse signal input to a timer input pin (TImn). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.




(**Remarks** are listed on the next page.)

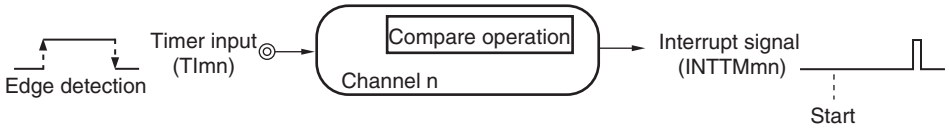
**(6) Measurement of high-/low-level width of input signal**

Counting is started by a single edge of the signal input to the timer input pin (TImn), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.




**(7) Delay counter**

Counting is started at the valid edge of the signal input to the timer input pin (TImn), and an interrupt is generated after any delay period.




**Remarks 1** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

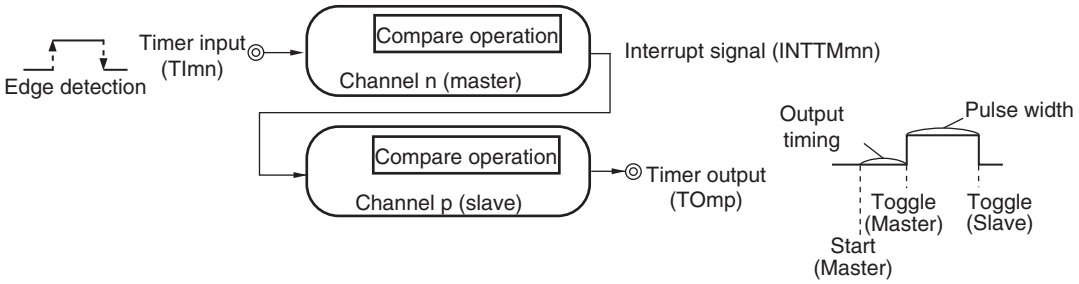
**2.** The presence or absence of timer I/O pins of channels 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

### 6.1.2 Simultaneous channel operation function

By using the combination of a master channel (a reference timer mainly controlling the cycle) and slave channels (timers operating according to the master channel), channels can be used for the following purposes.

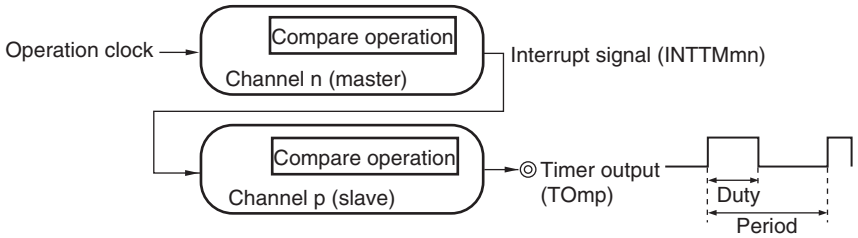
**(1) One-shot pulse output**

Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.




**(2) PWM (Pulse Width Modulation) output**

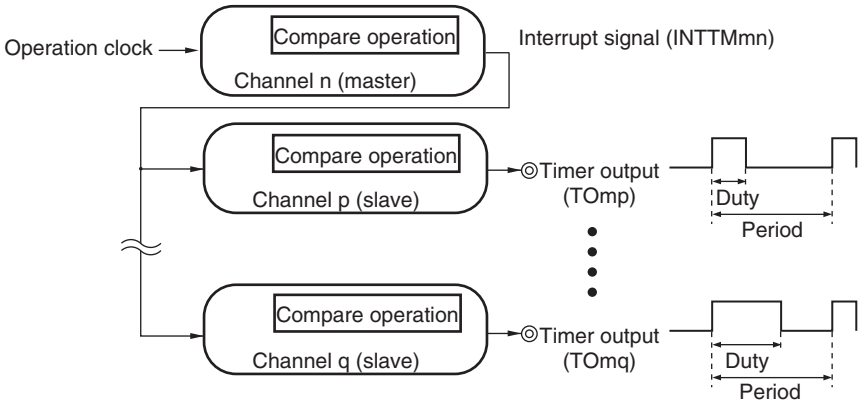
Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.




(**Caution** and **Remark** are is listed on the next page.)

**(3) Multiple PWM (Pulse Width Modulation) output**

By extending the PWM function and using one master channel and two or more slave channels, up to seven types of PWM signals that have a specific period and a specified duty factor can be generated.




**Caution There are several rules for using simultaneous channel operation function.**
**For details, see 6.4.1 Basic rules of simultaneous channel operation function.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
p, q: Slave channel number (n < p < q ≤ 7)

### 6.1.3 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels. This function can only be used for channels 1 and 3.

**Caution There are several rules for using 8-bit timer operation function.**
**For details, see 6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only).**

### 6.1.4 LIN-bus supporting function (channel 7 of unit 0 only)

Timer array unit is used to check whether signals received in LIN-bus communication match the LIN-bus communication format.

**(1) Detection of wakeup signal**

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD0) of UART0 and the count value of the timer is captured at the rising edge. In this way, a low-level width can be measured. If the low-level width is greater than a specific value, it is recognized as a wakeup signal.

**(2) Detection of sync break field**

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD0) of UART0 after a wakeup signal is detected, and the count value of the timer is captured at the rising edge. In this way, a low-level width is measured. If the low-level width is greater than a specific value, it is recognized as a sync break field.

**(3) Measurement of pulse width of sync field**

After a sync break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD0) of UART0 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

**Remark** For details about setting up the operations used to implement the LIN-bus, see **6.7.5 Operation as input signal high-/low-level width measurement**.

## 6.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

**Table 6-1. Configuration of Timer Array Unit**

| Item | Configuration |
|---|---|
| Timer/counter | Timer count register mn (TCRmn) |
| Register | Timer data register mn (TDRmn) |
| Timer input | TI00 to TI07, TI10 to TI17 Note 1, RxD0 pin (for LIN-bus) |
| Timer output | TO00 to TO07, TO10 to TO17 pins Note 1, output controller |
| Control registers | <Registers of unit setting block><br>• Peripheral enable register 0 (PER0)<br>• Timer clock select register m (TPSm)<br>• Timer channel enable status register m (TEm)<br>• Timer channel start register m (TSm)<br>• Timer channel stop register m (TTm)<br>• Timer input select register 0 (TIS0)<br>• Timer input select register 1 (TIS1)<br>• Timer input select register 2 (TIS2)<br>• Timer output enable register m (TOEm)<br>• Timer output register m (TOm)<br>• Timer output level register m (TOLm)<br>• Timer output mode register m (TOMm)<br>• PWM output delay control register 1 (PWMDLY1)<br>• PWM output delay control register 2 (PWMDLY2) Note 3 |
| | <Registers of each channel><br>• Timer mode register mn (TMRmn)<br>• Timer status register mn (TSRmn)<br>• Noise filter enable registers 1, 2 (NFEN1, NFEN2)<br>• Port mode register (PMxx) Note 2<br>• Port register (Pxx) Note 2 |

**Notes 1.** The presence or absence of timer I/O pins of channel 0 to 7 of the unit depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

**2.** The port mode registers (PMxx) and port registers (Pxx) to be set differ depending on the product. For details, **see 6.3.16 Port mode registers 1, 3, 4, 7, 12 (PM1, PM3, PM4, PM7, PM12).**

**3.** Bit allocation differs depending on the group.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

The presence or absence of timer I/O pins in each timer array unit channel depends on the product.

**Table 6-2. Timer I/O Pins provided in Each Product**

| Timer array unit channels | | I/O Pins of Each Product | | |
|---|---|---|---|---|
| | | Group E Products | Group B, C, and D Products | Group A Products |
| Unit 0 | Channel 0 | P17/TI00/TO00 | | |
| | Channel 1 | P30/TI01/TO01 | | |
| | Channel 2 | P16/TI02/TO02 | | |
| | Channel 3 | P125/TI03/TO03 | | |
| | Channel 4 | P13/TI04/TO04 | | |
| | Channel 5 | P15/TI05/TO05 | | |
| | Channel 6 | P14/TI06/TO06 | | |
| | Channel 7 | P120/TI07/TO07 | | |
| Unit 1 | Channel 0 | P41/TI10/TO10 | | × |
| | Channel 1 | P12/TI11/TO11 | | × |
| | Channel 2 | P11/TI12/TO12 | | × |
| | Channel 3 | P10/TI13/TO13 | | × |
| | Channel 4 | P31/TI14/TO14 | × | × |
| | Channel 5 | P70/TI15/TO15 | × | × |
| | Channel 6 | P32/TI16/TO16 | × | × |
| | Channel 7 | P71/TI17/TO17 | × | × |

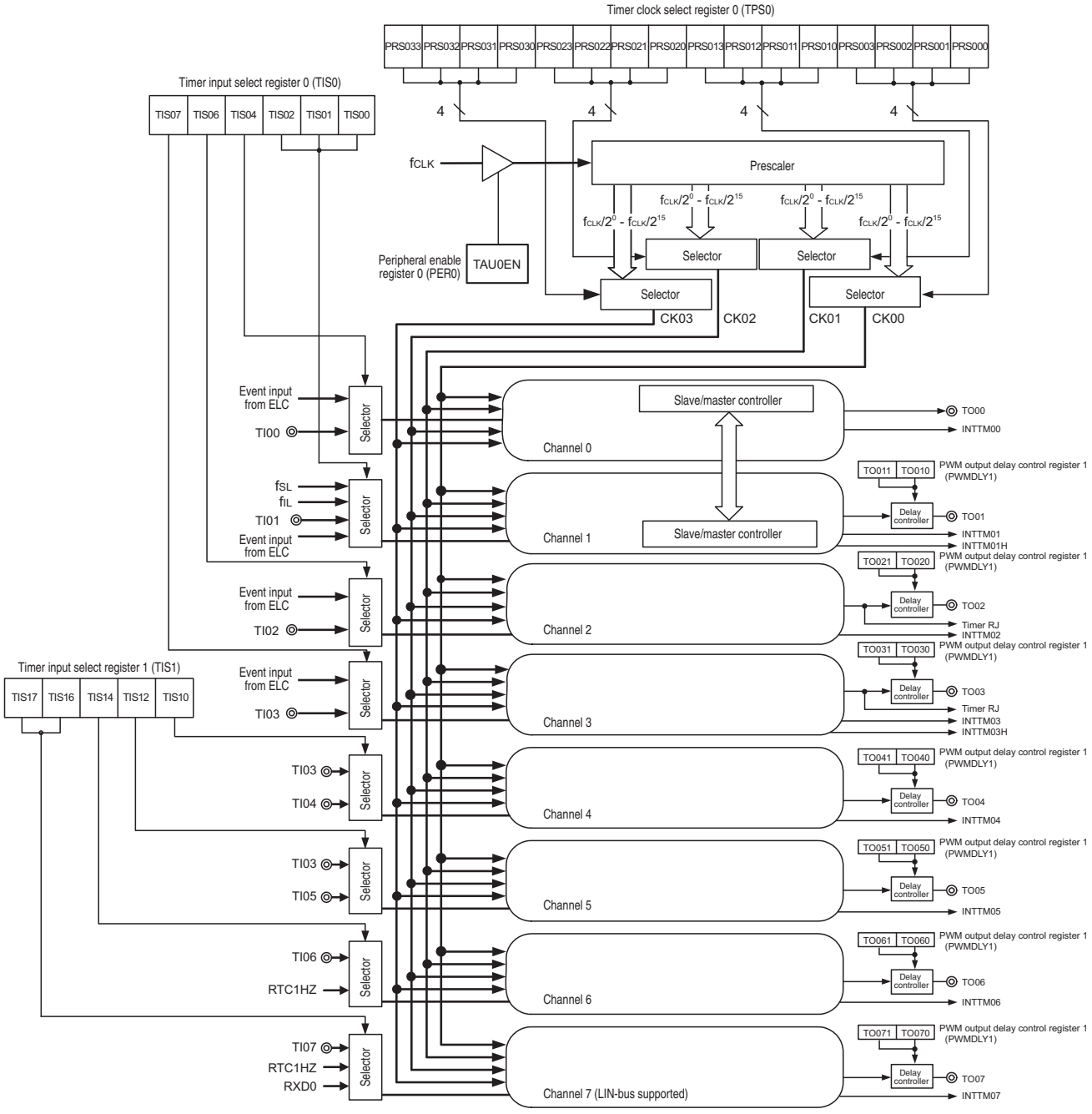
**Remarks 1.** When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

**2.** ×: The channel is not available.

**3.** For details on Groups A to E, see **the first page of CHAPTER 6.**

Figures 6-1 to 6-3 show the block diagrams of the timer array unit.

**Figure 6-1. Entire Configuration of Timer Array Unit 0 (Example: 100-pin products)**




**Remark** $f_{SL}$: Sub/low-speed on-chip oscillator clock
$f_{IL}$: Low-speed on-chip oscillator clock frequency

**Figure 6-2. Entire Configuration of Timer Array Unit 1 (Example: Groups B, C, and D)**

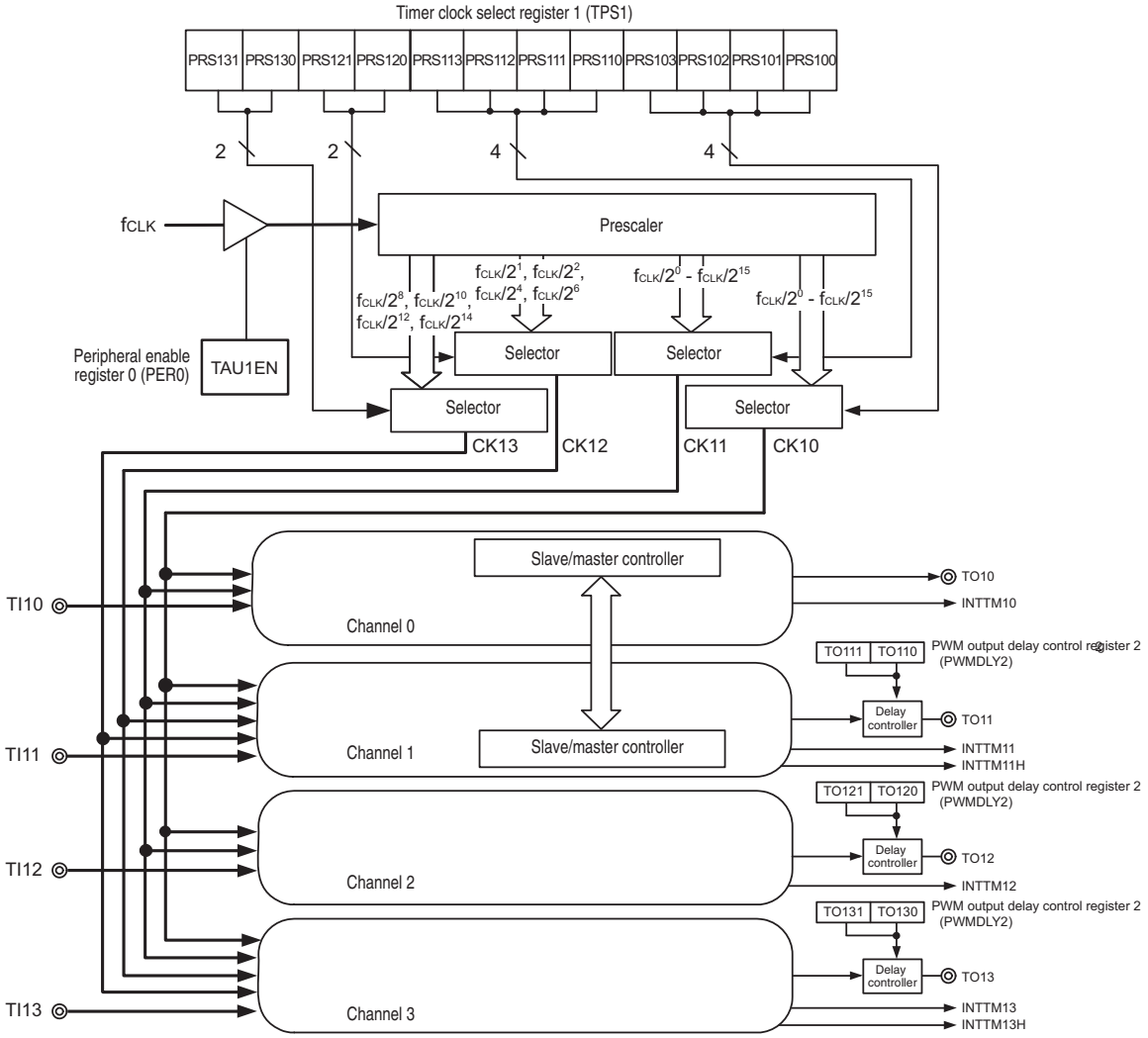


**Figure 6-3. Entire Configuration of Timer Array Unit 1 (Example: Group E)**

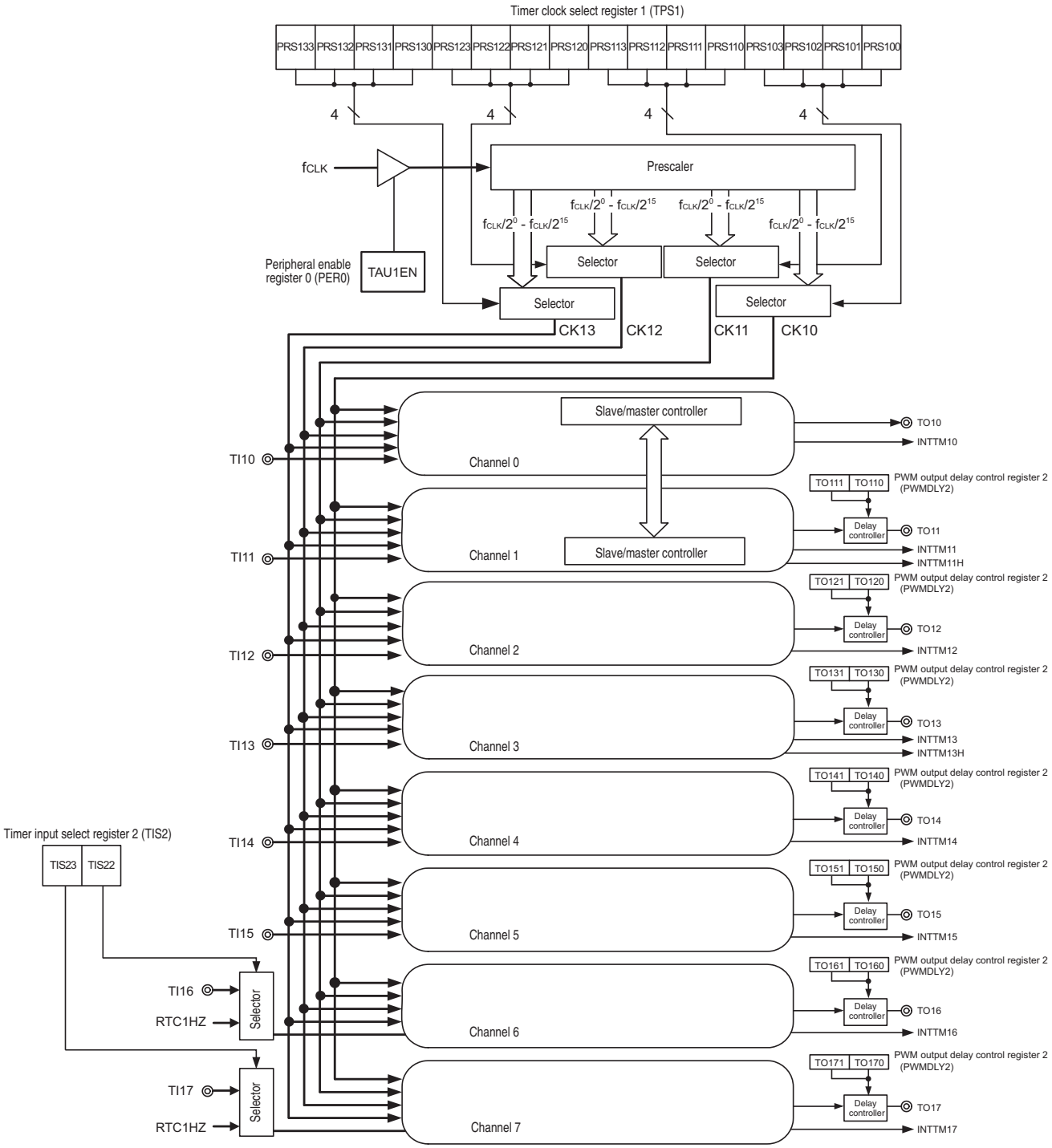
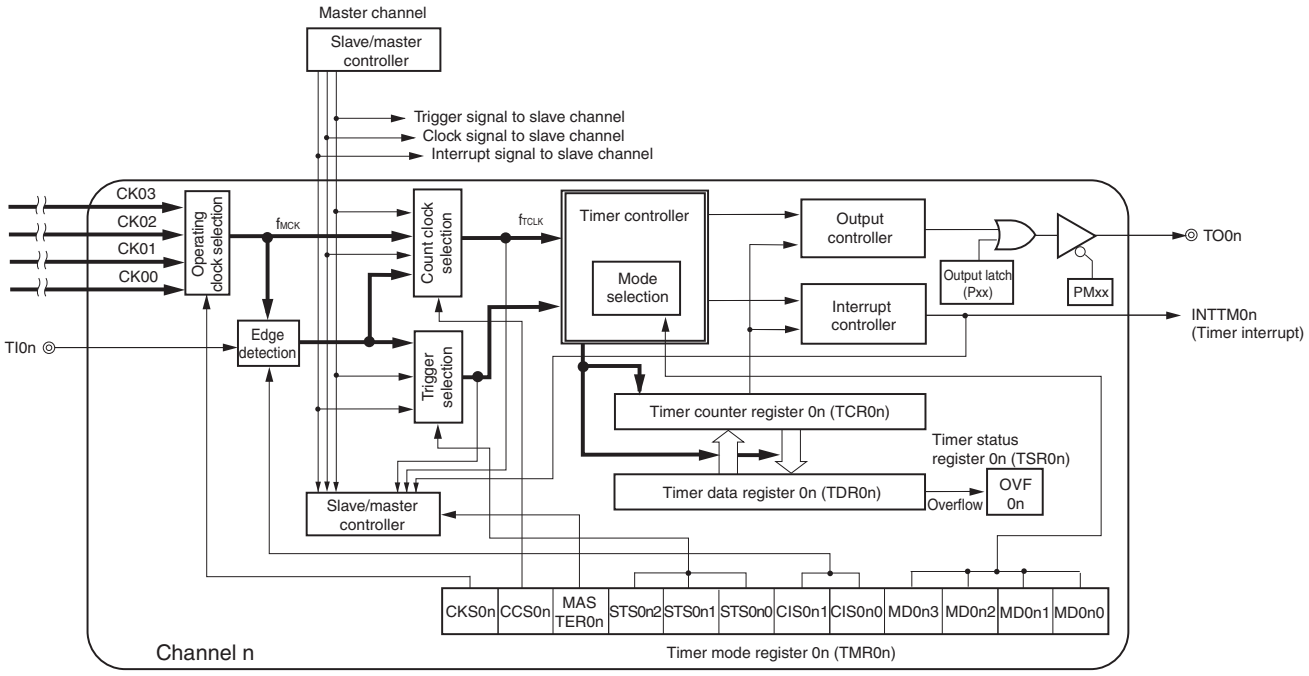
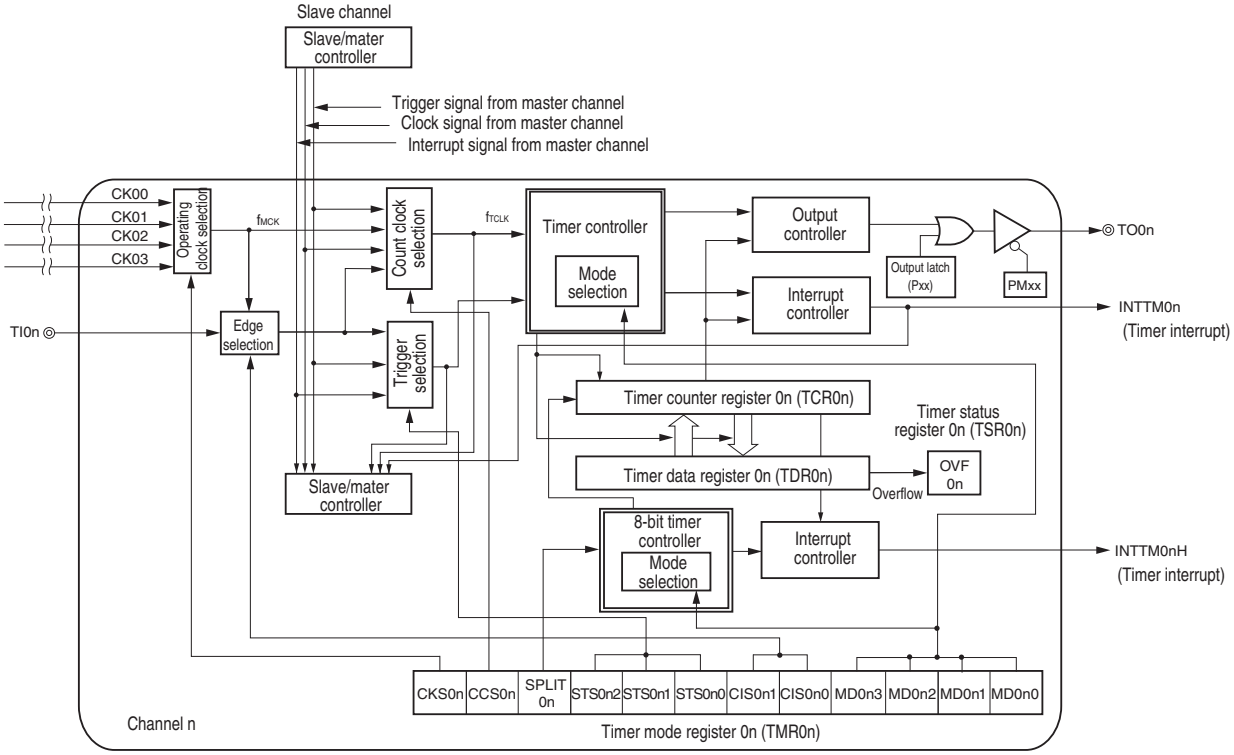


**Figure 6-4. Internal Block Diagram of Channel of Timer Array Unit 0**



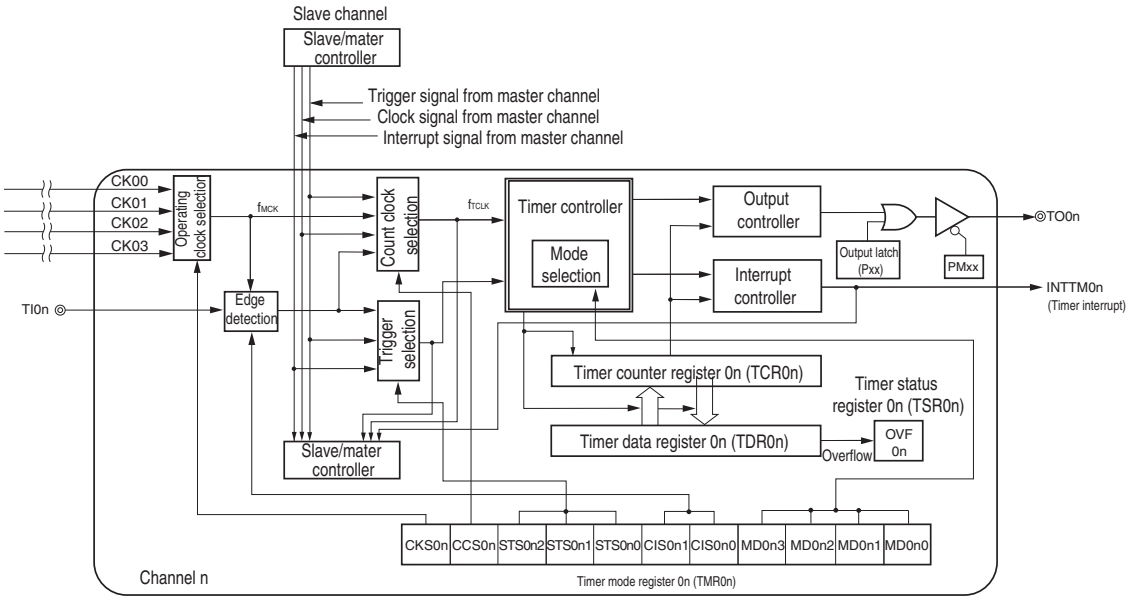

**Remark** n = 0, 2, 4, 6

**Figure 6-5. Internal Block Diagram of Channel of Timer Array Unit 0**




**Remark** n = 1, 3

**Figure 6-6. Internal Block Diagram of Channel of Timer Array Unit 0**




**Remark** n = 5, 7

### 6.2.1 Timer count register mn (TCRmn)

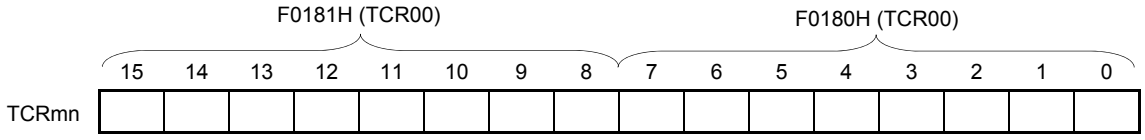
The TCRmn register is a 16-bit read-only register and is used to count clocks.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock.

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn) (refer to **6.3.3 Timer mode register mn (TMRmn)**).

**Figure 6-7. Format of Timer Count Register mn (TCRmn)**

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07), After reset: FFFFH R
F01C0H, F01C1H (TCR10) to F01CEH, F01CFH (TCR17)




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

The count value can be read by reading timer count register mn (TCRmn).

The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAUmEN bit of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the slave channel has been completed in the delay count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

The count value is cleared to 0000H in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

**Caution The count value is not captured to timer data register mn (TDRmn) even when the TCRmn register is read.**

The TCRmn register read value differs as follows according to operation mode changes and the operating status.

**Table 6-3. Timer Count Register mn (TCRmn) Read Value in Various Operation Modes**

| Operation Mode | Count Mode | Timer count register mn (TCRmn) Read Value[Note] | | | |
|---|---|---|---|---|---|
| | | Value if the operation mode was changed after releasing reset | Value if the operation mode was changed after count operation paused (TTmn = 1) | Value if the Operation was restarted after count operation paused (TTmn = 1) | Value when waiting for a start trigger after one count |
| Interval timer mode | Count down | FFFFH | Undefined | Stop value | – |
| Capture mode | Count up | 0000H | Undefined | Stop value | – |
| Event counter mode | Count down | FFFFH | Undefined | Stop value | – |
| One-count mode | Count down | FFFFH | Undefined | Stop value | FFFFH |
| Capture & one-count mode | Count up | 0000H | Undefined | Stop value | Capture value of TDRmn register + 1 |

**Note** This indicates the value read from the TCRmn register when channel n has stopped operating as a timer (TEmn = 0) and has been enabled to operate as a counter (TSmn = 1). The read value is held in the TCRmn register until the count operation starts.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

### 6.2.2 Timer data register mn (TDRmn)

This is a 16-bit register from which a capture function and a compare function can be selected.

The capture or compare function can be switched by selecting an operation mode by using the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn).

<R> When the TDRmn register is used for compare function, the value can be changed at any time.

This register can be read or written in 16-bit units.

In addition, for the TDRm1 and TDRm3 registers, while in the 8-bit timer mode (when the SPLIT bits of timer mode registers 01 and 03 (TMRm1, TMRm3) are 1), it is possible to rewrite the data in 8-bit units, with TDRm1H and TDRm3H used as the higher 8 bits, and TDRm1L and TDRm3L used as the lower 8 bits. However, reading is only possible in 16-bit units.

Reset signal generation clears this register to 0000H.

**Figure 6-8. Format of Timer Data Register mn (TDRmn) (n = 0, 2, 4 to 7)**

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02), FFF68H, FFF69H (TDR04) to FFF6EH, FFF6FH (TDR07), FFF70H, FFF71H (TDR10), FFF74H, FFF75H (TDR12), FFF78H, FFF79H (TDR14) to FFF7EH, FFF7FH (TDR17) After reset: 0000H R/W

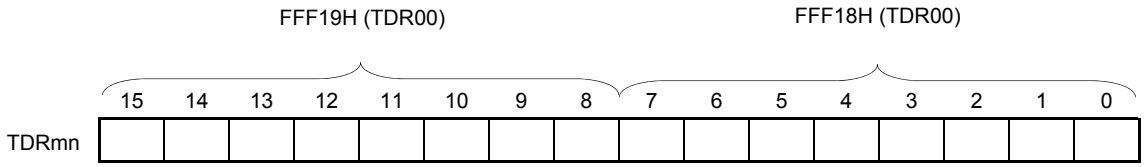
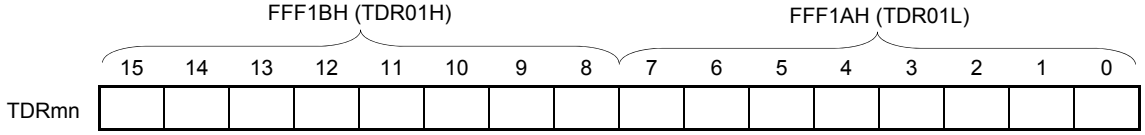



**Figure 6-9. Format of Timer Data Register mn (TDRmn) (n = 1, 3)**

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03), FFF72H, FFF73H (TDR11), FFF76H, FFF77H (TDR13) After reset: 0000H R/W




**(i) When timer data register mn (TDRmn) is used as compare register**

Counting down is started from the value set to the TDRmn register. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. The TDRmn register holds its value until it is rewritten.

**Caution The TDRmn register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.**

**(ii) When timer data register mn (TDRmn) is used as capture register**

The count value of timer count register mn (TCRmn) is captured to the TDRmn register when the capture trigger is input.

A valid edge of the TImn pin can be selected as the capture trigger. This selection is made by timer mode register mn (TMRmn).

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

## 6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer input select register 0 (TIS0)
- Timer input select register 1 (TIS1)
- Timer input select register 2 (TIS2)
- Timer output enable register m (TOEm)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- PWM output delay control register 1 (PWMDLY1)
- PWM output delay control register 2 (PWMDLY2) Note 2
- Noise filter enable registers 1, 2 (NFEN1, NFEN2)
- Port mode register (PMxx) Note 1
- Port register (Pxx) Note 1

**Notes 1.** The port mode registers (PMxx) and port registers (Pxx) to be set differ depending on the product. For details, **see 6.3.16 Port mode registers 1, 3, 4, 7, 12 (PM1, PM3, PM4, PM7, PM12).**

**2.** Bit allocation differs depending on the number of pins.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 (m = 1) is not provided in the Group A products.

**3.** Channel numbers 7 to 4 (n = 7 to 4) of unit 1 are not provided in the Group B, C, and D products.

### 6.3.1 Peripheral enable register 0 (PER0)

This registers is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the timer array unit 0 is used, be sure to set bit 0 (TAU0EN) of this register to 1.

When the timer array unit 1 is used, be sure to set bit 1 (TAU1EN) of this register to 1.

Set the PER0 register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 6-10. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

| Symbol | <7> | 6 | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PER0 | RTCEN | 0 | ADCEN | IICA0EN Notes1, 2 | SAU1EN Note1 | SAU0EN | TAU1EN Note1 | TAU0EN |

| TAU1EN | Control of timer array unit 1 input clock |
|---|---|
| 0 | Stops supply of input clock.<br>• SFR used by the timer array unit 1 cannot be written.<br>• The timer array unit 1 is in the reset status. |
| 1 | Supplies input clock.<br>• SFR used by the timer array unit 1 can be read/written. |

| TAU0EN | Control of timer array 0 unit input clock |
|---|---|
| 0 | Stops supply of input clock.<br>• SFR used by the timer array unit 0 cannot be written.<br>• The timer array unit 0 is in the reset status. |
| 1 | Supplies input clock.<br>• SFR used by the timer array unit 0 can be read/written. |

**Notes 1.** Not provided in RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory.

**2.** Not provided in RL78/F13 (CAN and LIN incorporated) with 30 pins and RL78/F14 with 30 pins.

**Cautions 1. When setting the timer array unit, be sure to set the TAUmEN bit to 1 first. If TAUmEN = 0, writing to a control register of timer array unit is ignored, and all read values are default values (except for the timer input select registers 0, 1, 2 (TIS0, TIS1, TIS2), noise filter enable registers 1, 2 (NFEN1, NFEN2), port mode registers 1, 3, 4, 7, 12 (PM1, PM3, PM4, PM7, PM12), port registers 1, 3, 4, 7, 12 (P1, P3, P4, P7, P12), and PWM output delay control registers 1, 2 (PWMDLY1, PWMDLY2)).**

**2. Be sure to clear the following bits to 0.**
**Bits 1, 3, 4, and 6 in RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory**
**Bits 4 and 6 in the RL78/F13 (CAN and LIN incorporated) products with 30 pins, or in the RL78/F14 products with 30 pins**
**Bit 6 in the products other than above**

### 6.3.2 Timer clock select register m (TPSm)

The TPSm register is a 16-bit register that is used to select the operation clocks that are supplied to each channel (CKm0, CKm1, CKm2, and CKm3) from the external prescaler. For unit 0 of the Group A, B, C, and D products, the clock frequency of CK00 is selected by using bits 3 to 0 of the TPS0 register, that of CK01 is selected by using bits 7 to 4, that of CK02 is selected by using bits 11 to 8, and that of CK03 is selected by using bits 15 to 12.

For unit 1 of the Group B, C, and D products, the clock frequency of CK10 is selected by using bits 3 to 0 of the TPS1 register, that of CK11 is selected by using bits 7 to 4, that of CK12 is selected by using bits 9 and 8, and that of CK13 is selected by using bits 13 and 12.

For the Group E products, the clock frequency of CKm0 is selected by using bits 3 to 0 of the TPSm register, that of CKm1 is selected by using bits 7 to 4, that of CKm2 is selected by using bits 11 to 8, and that of CKm3 is selected by using bits 15 to 12.

Rewriting of the TPSm register during timer operation is possible only in the following cases.

If the PRSm00 to PRSm03 bits can be rewritten (n = 0 to 7):
All channels for which CKm0 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 0) are stopped (TEmn = 0).
If the PRSm10 to PRSm13 bits can be rewritten (n = 0 to 7):
All channels for which CKm1 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 0) are stopped (TEmn = 0).
If the PRSm20 to PRSm23 bits can be rewritten (n = 0 to 7):
All channels for which CKm2 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 1) are stopped (TEmn = 0).
If the PRSm30 to PRSm33 bits can be rewritten (n = 0 to 7):
All channels for which CKm3 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 1) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

**Figure 6-11. Format of Timer Clock Select Register m (TPSm) (8-ch version)**

Address: F01B6H, F01B7H (TPS0), F01F6H, F01F7H (TPS1) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TPSm | PRS m33 | PRS m32 | PRS m31 | PRS m30 | PRS m23 | PRS m22 | PRS m21 | PRS m20 | PRS m13 | PRS m12 | PRS m11 | PRS m10 | PRS m03 | PRS m02 | PRS m01 | PRS m00 |

| PRS mk3 | PRS mk2 | PRS mk1 | PRS mk0 | Selection of operation clock (CKmk) Note (k = 0 to 3) | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | $f_{CLK}$ = 2 MHz | $f_{CLK}$ = 5 MHz | $f_{CLK}$ = 10 MHz | $f_{CLK}$ = 20 MHz | $f_{CLK}$ = 32 MHz |
| 0 | 0 | 0 | 0 | $f_{CLK}$ | 2 MHz | 5 MHz | 10 MHz | 20 MHz | 32 MHz |
| 0 | 0 | 0 | 1 | $f_{CLK}/2$ | 1 MHz | 2.5 MHz | 5 MHz | 10 MHz | 16 MHz |
| 0 | 0 | 1 | 0 | $f_{CLK}/2^2$ | 500 kHz | 1.25 MHz | 2.5 MHz | 5 MHz | 8 MHz |
| 0 | 0 | 1 | 1 | $f_{CLK}/2^3$ | 250 kHz | 625 kHz | 1.25 MHz | 2.5 MHz | 4 MHz |
| 0 | 1 | 0 | 0 | $f_{CLK}/2^4$ | 125 kHz | 312.5 kHz | 625 kHz | 1.25 MHz | 2 MHz |
| 0 | 1 | 0 | 1 | $f_{CLK}/2^5$ | 62.5 kHz | 156.2 kHz | 312.5 kHz | 625 kHz | 1 MHz |
| 0 | 1 | 1 | 0 | $f_{CLK}/2^6$ | 31.25 kHz | 78.1 kHz | 156.2 kHz | 312.5 kHz | 500 kHz |
| 0 | 1 | 1 | 1 | $f_{CLK}/2^7$ | 15.62 kHz | 39.1 kHz | 78.1 kHz | 156.2 kHz | 250 kHz |
| 1 | 0 | 0 | 0 | $f_{CLK}/2^8$ | 7.81 kHz | 19.5 kHz | 39.1 kHz | 78.1 kHz | 125 kHz |
| 1 | 0 | 0 | 1 | $f_{CLK}/2^9$ | 3.91 kHz | 9.76 kHz | 19.5 kHz | 39.1 kHz | 62.5 kHz |
| 1 | 0 | 1 | 0 | $f_{CLK}/2^{10}$ | 1.95 kHz | 4.88 kHz | 9.76 kHz | 19.5 kHz | 31.25 kHz |
| 1 | 0 | 1 | 1 | $f_{CLK}/2^{11}$ | 976 Hz | 2.44 kHz | 4.88 kHz | 9.76 kHz | 15.63 kHz |
| 1 | 1 | 0 | 0 | $f_{CLK}/2^{12}$ | 488 Hz | 1.22 kHz | 2.44 kHz | 4.88 kHz | 7.81 kHz |
| 1 | 1 | 0 | 1 | $f_{CLK}/2^{13}$ | 244 Hz | 610 Hz | 1.22 kHz | 2.44 kHz | 3.91 kHz |
| 1 | 1 | 1 | 0 | $f_{CLK}/2^{14}$ | 122 Hz | 305 Hz | 610 Hz | 1.22 kHz | 1.95 kHz |
| 1 | 1 | 1 | 1 | $f_{CLK}/2^{15}$ | 61 Hz | 153 Hz | 305 Hz | 610 Hz | 976 Hz |

**Note** When changing the clock selected for $f_{CLK}$ (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).

**Cautions 1. TPS1 is not provided in the Group A products, because the timer array unit 1 is not provided. This format cannot be applied to unit 1 of the Group B, C, and D products (see the specifications of 4-channel version in Figure 6-12).**

**2. When selecting $f_{CLK}$ (not divided) as the operation clock (CKmk) and setting TDRnm = 0000H (n = 0, 1; m = 0 to 7), set the interrupt mask flag to "interrupt processing disabled" (TMMKnm = 1).**

**Remarks 1.** $f_{CLK}$: CPU/peripheral hardware clock frequency

**2.** The above clock becomes high level for one period of $f_{CLK}$ from its rising edge (m = 0, 1). For details, see **6.5.1 Count clock ($f_{TCLK}$)**.

**Figure 6-12. Format of Timer Clock Select Register 1 (TPS1) (4-ch version)**

Address: F01F6H, F01F7H After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TPSm | 0 | 0 | PRS 131 | PRS 130 | 0 | 0 | PRS 121 | PRS 120 | PRS 113 | PRS 112 | PRS 111 | PRS 110 | PRS 103 | PRS 102 | PRS 101 | PRS 100 |

| PRS 121 | PRS 120 | | Selection of operation clock (CK12) Note | | | | |
|---|---|---|---|---|---|---|---|
| | | | $f_{CLK}$ = 2 MHz | $f_{CLK}$ = 5 MHz | $f_{CLK}$ = 10 MHz | $f_{CLK}$ = 20 MHz | $f_{CLK}$ = 32 MHz |
| 0 | 0 | $f_{CLK}/2$ | 1 MHz | 2.5 MHz | 5 MHz | 10 MHz | 16 MHz |
| 0 | 1 | $f_{CLK}/2^2$ | 500 kHz | 1.25 MHz | 2.5 MHz | 5 MHz | 8 MHz |
| 1 | 0 | $f_{CLK}/2^4$ | 125 kHz | 312.5 kHz | 625 kHz | 1.25 MHz | 2 MHz |
| 1 | 1 | $f_{CLK}/2^6$ | 31.25 kHZ | 78.1 kHz | 156.2 kHz | 312.5 kHz | 500 kHZ |

| PRS 131 | PRS 130 | | Selection of operation clock (CK13) Note | | | | |
|---|---|---|---|---|---|---|---|
| | | | $f_{CLK}$ = 2 MHz | $f_{CLK}$ = 5 MHz | $f_{CLK}$ = 10 MHz | $f_{CLK}$ = 20 MHz | $f_{CLK}$ = 32 MHz |
| 0 | 0 | $f_{CLK}/2^8$ | 7.81 kHz | 19.5 kHz | 39.1 kHz | 78.1 kHz | 125 kHz |
| 0 | 1 | $f_{CLK}/2^{10}$ | 1.95 kHz | 4.88 kHz | 9.76 kHz | 19.5 kHz | 31.25 kHz |
| 1 | 0 | $f_{CLK}/2^{12}$ | 488 Hz | 1.22 kHz | 2.44 kHz | 4.88 kHz | 7.81 kHz |
| 1 | 1 | $f_{CLK}/2^{14}$ | 122 HZ | 305 Hz | 610 Hz | 1.22 kHz | 1.95 kHZ |

**Note** The above format is applied to the TPS1 of the Group B, C, and D products.
When changing the clock selected for $f_{CLK}$ (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).
The timer array unit must also be stopped if the operating clock ($f_{MCK}$) specified by using the CKSmn0, and CKSmn1 bits or the valid edge of the signal input from the TImn pin is selected as the count clock ($f_{TCLK}$).

**Cautions 1. This format cannot be applied to the Group A and E products and unit 0 of the Group B, C, and D products (see the specifications of 8-channel version in Figure 6-11).**
**2. When selecting $f_{CLK}$ (not divided) as the operation clock (CKmk) and setting TDRnm = 0000H (n = 0, 1; m = 0 to 7), set the interrupt mask flag to "interrupt processing disabled" (TMMKnm = 1).**

By using channels 1 and 3 in the 8-bit timer mode and specifying CKm2 or CKm3 as the operation clock, the interval times shown in Table 6-4 can be achieved by using the interval timer function.

**Table 6-4. Interval Times Available for Operation Clock CKSm2 or CKSm3**

| Clock | | Interval time ($f_{CLK}$ = 32 MHz) Note | | | |
|---|---|---|---|---|---|
| | | 10 μs | 100 μs | 1 ms | 10 ms |
| CKm2 | $f_{CLK}/2$ | √ | – | – | – |
| | $f_{CLK}/2^2$ | √ | – | – | – |
| | $f_{CLK}/2^4$ | √ | √ | – | – |
| | $f_{CLK}/2^6$ | √ | √ | – | – |
| CKm3 | $f_{CLK}/2^8$ | – | √ | √ | – |
| | $f_{CLK}/2^{10}$ | – | √ | √ | – |
| | $f_{CLK}/2^{12}$ | – | – | √ | √ |
| | $f_{CLK}/2^{14}$ | – | – | √ | √ |

**Note** The margin is within 5 %.

**Remarks 1.** $f_{CLK}$: CPU/peripheral hardware clock frequency
**2.** For details of a signal of $f_{CLK}/2^i$ selected with the TPSm register, see **6.5.1 Count clock ($f_{TCLK}$).**

### 6.3.3 Timer mode register mn (TMRmn)

The TMRmn register sets an operation mode of channel n. This register is used to select the operation clock ($f_{MCK}$), select the count clock, select the master/slave, select the 16 or 8-bit timer (only for channels 1 and 3), specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture and one-count).

Rewriting the TMRmn register is prohibited when the register is in operation (when TEmn = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEmn = 1) (for details, see **6.7 Independent Channel Operation Function of Timer Array Unit** and **6.8 Simultaneous Channel Operation Function of Timer Array Unit**.

Set the TMRmn register by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

**Caution The bits mounted depend on the channels in the bit 11 of TMRmn register.**
**TMRm2, TMRm4, TMRm6: MASTERmn bit (n = 2, 4, 6)**
**TMRm1, TMRm3: SPLITmn bit (n = 1, 3)**
**TMRm0, TMRm5, TMRm7: Fixed to 0 (n = 0, 5, 7)**

**Figure 6-13. Format of Timer Mode Register mn (TMRmn) (1/4)**

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07), After reset: 0000H R/W
F01D0H, F01D1H (TMR10) to F01DEH, F01DFH (TMR17)

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn (n = 2, 4, 6) | CKS mn1 | CKS mn0 | 0 | CCS mn | MAST ERmn | STS mn2 | STS mn1 | STS mn0 | CIS mn1 | CIS mn0 | 0 | 0 | MD mn3 | MD mn2 | MD mn1 | MD mn0 |

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn (n = 1, 3) | CKS mn1 | CKS mn0 | 0 | CCS mn | SPLIT mn | STS mn2 | STS mn1 | STS mn0 | CIS mn1 | CIS mn0 | 0 | 0 | MD mn3 | MD mn2 | MD mn1 | MD mn0 |

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn (n = 0, 5, 7) | CKS mn1 | CKS mn0 | 0 | CCS mn | 0 Note | STS mn2 | STS mn1 | STS mn0 | CIS mn1 | CIS mn0 | 0 | 0 | MD mn3 | MD mn2 | MD mn1 | MD mn0 |

| CKS mn1 | CKS mn0 | Selection of operation clock ($f_{MCK}$) of channel n |
|---|---|---|
| 0 | 0 | Operation clock CKm0 set by timer clock select register m (TPSm) |
| 0 | 1 | Operation clock CKm2 set by timer clock select register m (TPSm) |
| 1 | 0 | Operation clock CKm1 set by timer clock select register m (TPSm) |
| 1 | 1 | Operation clock CKm3 set by timer clock select register m (TPSm) |
| Operation clock ($f_{MCK}$ ) is used by the edge detector. A count clock ($f_{TCLK}$) and a sampling clock are generated depending on the setting of the CCSmn bit. | | |

| CCS mn | Selection of count clock ($f_{TCLK}$) of channel n |
|---|---|
| 0 | Operation clock ($f_{MCK}$) specified by the CKSmn0 and CKSmn1 bits |
| 1 | Valid edge of input signal input from the TImn pin<br>Valid edge of input signal selected by TIS0 in channel 5 |
| Count clock ($f_{TCLK}$) is used for the timer/counter, output controller, and interrupt controller. | |

**Note** Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

**Cautions 1. Be sure to clear bits 13, 5, and 4 to "0".**

**2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for $f_{CLK}$ is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKSmn0 and CKSmn1 bits ($f_{MCK}$) or the valid edge of the signal input from the TImn pin is selected as the count clock ($f_{TCLK}$).**

**3. Be sure to clear CKS1n0 (n = 0, 2) to "0" in the Group B, C, and D products.**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** TMR1n is not provided in the Group A products.
TMR17 to TMR14 are not provided in the Group B, C, and D products.

**Figure 6-13. Format of Timer Mode Register mn (TMRmn) (2/4)**

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07), After reset: 0000H R/W
F01D0H, F01D1H (TMR10) to F01DEH, F01DFH (TMR17)

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn (n = 2, 4, 6) | CKS mn1 | CKS mn0 | 0 | CCS mn | MASTERmn | STS mn2 | STS mn1 | STS mn0 | CIS mn1 | CIS mn0 | 0 | 0 | MD mn3 | MD mn2 | MD mn1 | MD mn0 |

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn (n = 1, 3) | CKS mn1 | CKS mn0 | 0 | CCS mn | SPLIT mn | STS mn2 | STS mn1 | STS mn0 | CIS mn1 | CIS mn0 | 0 | 0 | MD mn3 | MD mn2 | MD mn1 | MD mn0 |

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn (n = 0, 5, 7) | CKS mn1 | CKS mn0 | 0 | CCS mn | 0 Note | STS mn2 | STS mn1 | STS mn0 | CIS mn1 | CIS mn0 | 0 | 0 | MD mn3 | MD mn2 | MD mn1 | MD mn0 |

(Bit 11 of TMRmn (n = 2, 4, 6))

| MASTERmn | Selection between using channel n independently or simultaneously with another channel(as a slave or master) |
|---|---|
| 0 | Operates in independent channel operation function or as slave channel in simultaneous channel operation function. |
| 1 | Operates as master channel in simultaneous channel operation function. |
| Only channels 2, 4, and 6 can be set as a master channel (MASTERmn = 1).<br>Channels 0, 5, and 7 are fixed to 0 (channel 0 always operates as master regardless of the bit setting, because it is the highest channel).<br>Clear the MASTERmn bit to 0 for a channel that is used with the independent channel operation function. | |

(Bit 11 of TMRmn (n = 1, 3))

| SPLITmn | Selection of 8 or 16-bit timer operation for channels 1 and 3 |
|---|---|
| 0 | Operates as 16-bit timer.<br>(Operates in independent channel operation function or as slave channel in simultaneous channel operation function.) |
| 1 | Operates as 8-bit timer. |

| STS mn2 | STS mn1 | STS mn0 | Setting of start trigger or capture trigger of channel n |
|---|---|---|---|
| 0 | 0 | 0 | Only software trigger start is valid (other trigger sources are unselected). |
| 0 | 0 | 1 | Valid edge of the TImn pin input is used as both the start trigger and capture trigger. |
| 0 | 1 | 0 | Both the edges of the TImn pin input are used as a start trigger and a capture trigger. |
| 1 | 0 | 0 | Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function). |
| Other than above | | | Setting prohibited |

**Note** Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored. In addition, channel 0 operates as master.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** TMR1n is not provided in the Group A products.
TMR17 to TMR14 are not provided in the Group B, C, and D products.

**Figure 6-13. Format of Timer Mode Register mn (TMRmn) (3/4)**

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07), F01D0H, F01D1H (TMR10) to F01DEH, F01DFH (TMR17) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn (n = 2, 4, 6) | CKS mn1 | CKS mn0 | 0 | CCS mn | MAST ERmn | STS mn2 | STS mn1 | STS mn0 | CIS mn1 | CIS mn0 | 0 | 0 | MD mn3 | MD mn2 | MD mn1 | MD mn0 |

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn (n = 1, 3) | CKS mn1 | CKS mn0 | 0 | CCS mn | SPLIT mn | STS mn2 | STS mn1 | STS mn0 | CIS mn1 | CIS mn0 | 0 | 0 | MD mn3 | MD mn2 | MD mn1 | MD mn0 |

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn (n = 0, 5, 7) | CKS mn1 | CKS mn0 | 0 | CCS mn | 0 Note | STS mn2 | STS mn1 | STS mn0 | CIS mn1 | CIS mn0 | 0 | 0 | MD mn3 | MD mn2 | MD mn1 | MD mn0 |

| CIS mn1 | CIS mn0 | Selection of TImn pin input valid edge |
|---|---|---|
| 0 | 0 | Falling edge |
| 0 | 1 | Rising edge |
| 1 | 0 | Both edges (when low-level width is measured)<br>Start trigger: Falling edge, Capture trigger: Rising edge |
| 1 | 1 | Both edges (when high-level width is measured)<br>Start trigger: Rising edge, Capture trigger: Falling edge |
| If both the edges are specified when the value of the STSmn2 to STSmn0 bits is other than 010B, set the CISmn1 to CISmn0 bits to 10B. | | |

**Note** Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

**Remarks** **1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** TMR1n is not provided in the Group A products.
TMR17 to TMR14 are not provided in the Group B, C, and D products.

**Figure 6-13. Format of Timer Mode Register mn (TMRmn) (4/4)**

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07), After reset: 0000H R/W
F01D0H, F01D1H (TMR10) to F01DEH, F01DFH (TMR17)

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn (n = 2, 4, 6) | CKS mn1 | CKS mn0 | 0 | CCS mn | MAST ERmn | STS mn2 | STS mn1 | STS mn0 | CIS mn1 | CIS mn0 | 0 | 0 | MD mn3 | MD mn2 | MD mn1 | MD mn0 |

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn (n = 1, 3) | CKS mn1 | CKS mn0 | 0 | CCS mn | SPLIT mn | STS mn2 | STS mn1 | STS mn0 | CIS mn1 | CIS mn0 | 0 | 0 | MD mn3 | MD mn2 | MD mn1 | MD mn0 |

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn (n = 0, 5, 7) | CKS mn1 | CKS mn0 | 0 | CCS mn | 0 Note 1 | STS mn2 | STS mn1 | STS mn0 | CIS mn1 | CIS mn0 | 0 | 0 | MD mn3 | MD mn2 | MD mn1 | MD mn0 |

| MD mn3 | MD mn2 | MD mn1 | Operation mode of channel n | Corresponding function | Count operation of TCR |
|---|---|---|---|---|---|
| 0 | 0 | 0 | Interval timer mode | Interval timer / Square wave output / Divider function / PWM output (master) | Counting down |
| 0 | 1 | 0 | Capture mode | Input pulse interval measurement | Counting up |
| 0 | 1 | 1 | Event counter mode | External event counter | Counting down |
| 1 | 0 | 0 | One-count mode | Delay counter / One-shot pulse output / PWM output (slave) | Counting down |
| 1 | 1 | 0 | Capture & one-count mode | Measurement of high-/low-level width of input signal | Counting up |
| Other than above | | | Setting prohibited | | |
| Each operation mode varies depending on the MDmn0 bit (see table below). | | | | | |

| Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above)) | MD mn0 | Setting of starting counting and interrupt |
|---|---|---|
| • Interval timer mode (0, 0, 0)<br>• Capture mode (0, 1, 0) | 0 | Timer interrupt is not generated when counting is started (timer output does not change, either). |
| | 1 | Timer interrupt is generated when counting is started (timer output also changes). |
| • Event counter mode (0, 1, 1) | 0 | Timer interrupt is not generated when counting is started (timer output does not change, either). |
| • One-count mode Note 2 (1, 0, 0) | 0 | Start trigger is invalid during counting operation. At that time, interrupt is not generated, either. |
| | 1 | Start trigger is valid during counting operation Note 3. At that time, interrupt is also generated. |
| • Capture & one-count mode (1, 1, 0) | 0 | Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated, either. |
| Other than above | | Setting prohibited |

**Notes 1.** Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

**2.** In one-count mode, interrupt output (INTTMmn) when starting a count operation and TOmn output are not controlled.

**3.** If the start trigger (TSmn = 1) is issued during operation, the counter is initialized and recounting is started (no interrupt request is generated).

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** TMR1n is not provided in the Group A products.

**3.** TMR17 to TMR14 are not provided in the Group B, C, and D products.

### 6.3.4 Timer status register mn (TSRmn)

The TSRmn register indicates the overflow status of the counter of channel n.

The TSRmn register is valid only in the capture mode (MDmn3 to MDmn1 = 010B) and capture & one-count mode (MDmn3 to MDmn1 = 110B). See **Table 6-5** for the operation of the OVF bit in each operation mode and set/clear conditions.

The TSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSRmn register can be set with an 8-bit memory manipulation instruction with TSRmnL.

Reset signal generation clears this register to 0000H.

**Figure 6-14. Format of Timer Status Register mn (TSRmn)**

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07), After reset: 0000H R
F01E0H, F01E1H (TSR10) to F01EEH, F01EFH (TSR17)

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TSRmn | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | OVF |

| OVF | Counter overflow status of channel n |
|---|---|
| 0 | Overflow does not occur. |
| 1 | Overflow occurs. |
| When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow. | |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** TSR1n is not provided in the Group A products.

**3.** TSR17 to TSR14 are not provided in the Group B, C, and D products.

**Table 6-5. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode**

| Timer operation mode | OVF bit | Set/clear conditions |
|---|---|---|
| • Capture mode<br>• Capture & one-count mode | clear | When no overflow has occurred upon capturing |
| | set | When an overflow has occurred upon capturing |
| • Interval timer mode<br>• Event counter mode<br>• One-count mode | clear | – (Use prohibited) |
| | set | |

**Remark** The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

### 6.3.5 Timer channel enable status register m (TEm)

The TEm register is used to enable or stop the timer operation of each channel.

Each bit of the TEm register corresponds to each bit of the timer channel start register m (TSm) and the timer channel stop register m (TTm). When a bit of the TSm register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TTm register is set to 1, the corresponding bit of this register is cleared to 0.

The TEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TEmL.

Reset signal generation clears this register to 0000H.

**Figure 6-15. Format of Timer Channel Enable Status Register m (TEm)**

Address: F01B0H, F01B1H (TE0), F01F0H, F01F1H (TE1) After reset: 0000H R

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TEm | 0 | 0 | 0 | 0 | TEHm3 | 0 | TEHm1 | 0 | TEm7 | TEm6 | TEm5 | TEm4 | TEm3 | TEm2 | TEm1 | TEm0 |

| TEHm3 | Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 3 is in the 8-bit timer mode |
|---|---|
| 0 | Operation is stopped. |
| 1 | Operation is enabled. |

| TEHm1 | Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode |
|---|---|
| 0 | Operation is stopped. |
| 1 | Operation is enabled. |

| TEmn | Indication of operation enable/stop status of channel n |
|---|---|
| 0 | Operation is stopped. |
| 1 | Operation is enabled. |
| This bit displays whether operation of the lower 8-bit timer for TEm1 and TEm3 is enabled or stopped when channel 1 or 3 is in the 8-bit timer mode. | |

**Cautions 1. Be sure to clear bits 15 to 12, 10, and 8 to "0".**

**2. Be sure to clear TE1n (n = 7 to 4) to “0” in the Group B, C, and D products.**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** TE1n is not provided in the Group A products.

TE17 to TE14 are not provided in the Group B, C, and D products**.**

### 6.3.6 Timer channel start register m (TSm)

The TSm register is a trigger register that is used to initialize timer count register mn (TCRmn) and start the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is set to 1. The TSmn, TSHm1, TSHm3 bits are immediately cleared when operation is enabled (TEmn, TEHm1, TEHm3 = 1), because they are trigger bits.

The TSm register can be set by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the TSm register with a 1-bit or 8-bit memory manipulation instruction with TSmL.

Reset signal generation clears this register to 0000H.

**Figure 6-16. Format of Timer Channel Start Register m (TSm)**

Address: F01B2H, F01B3H (TS0), F01F2H, F01F3H (TS1) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TSm | 0 | 0 | 0 | 0 | TSHm3 | 0 | TSHm1 | 0 | TSm7 | TSm6 | TSm5 | TSm4 | TSm3 | TSm2 | TSm1 | TSm0 |

| TSHm3 | Trigger to enable operation (start operation) of the higher 8-bit timer when channel 3 is in the 8-bit timer mode |
|---|---|
| 0 | No trigger operation |
| 1 | The TEHm3 bit is set to 1 and the count operation becomes enabled.<br>The TCRm3 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-6** in **6.5.2 Start timing of counter**). |

| TSHm1 | Trigger to enable operation (start operation) of the higher 8-bit timer when channel 1 is in the 8-bit timer mode |
|---|---|
| 0 | No trigger operation |
| 1 | The TEHm1 bit is set to 1 and the count operation becomes enabled.<br>The TCRm1 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-6** in **6.5.2 Start timing of counter**). |

| TSmn | Operation enable (start) trigger of channel n |
|---|---|
| 0 | No trigger operation |
| 1 | The TEmn bit is set to 1 and the count operation becomes enabled.<br>The TCRmn register count operation start in the count operation enabled state varies depending on each operation mode (see **Table 6-6** in **6.5.2 Start timing of counter**).<br>This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for TSm1 and TSm3 when channel 1 or 3 is in the 8-bit timer mode. |

**Cautions 1. Be sure to clear bits 15 to 12, 10, and 8 to "0".**

**2. Be sure to clear TS1n (n = 7 to 4) to "0" in the Group B, C, and D products.**

**3. When switching from a function that does not use TImn pin input to one that does, the following wait period is required from when timer mode register mn (TMRmn) is set until the TSmn (TSHm1, TSHm3) bit is set to 1.**

**When the TImn pin noise filter is enabled (TNFENmn = 1): Four cycles of the operation clock ($f_{MCK}$)**

**When the TImn pin noise filter is disabled (TNFENmn = 0): Two cycles of the operation clock ($f_{MCK}$)**

**Remarks 1.** When the TSm register is read, 0 is always read.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**3.** TS1n is not provided in the Group A products.

TS17 to TS14 are not provided in the Group B, C, and D products.

### 6.3.7 Timer channel stop register m (TTm)

The TTm register is a trigger register that is used to stop the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is cleared to 0. The TTmn, TTHm1, TTHm3 bits are immediately cleared when operation is stopped (TEmn, TTHm1, TTHm3 = 0), because they are trigger bits.

The TTm register can be set by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the TTm register with a 1-bit or 8-bit memory manipulation instruction with TTmL.

Reset signal generation clears this register to 0000H.

**Figure 6-17. Format of Timer Channel Stop Register m (TTm)**

Address: F01B4H, F01B5H (TT0), F01F4H, F01F5H (TT1) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TTm | 0 | 0 | 0 | 0 | TTHm3 | 0 | TTHm1 | 0 | TTm7 | TTm6 | TTm5 | TTm4 | TTm3 | TTm2 | TTm1 | TTm0 |

| TTHm3 | Trigger to stop operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode |
|---|---|
| 0 | No trigger operation |
| 1 | Operation is stopped (stop trigger is generated). |

| TTHm1 | Trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode |
|---|---|
| 0 | No trigger operation |
| 1 | Operation is stopped (stop trigger is generated). |

| TTmn | Operation stop trigger of channel n |
|---|---|
| 0 | No trigger operation |
| 1 | Operation is stopped (stop trigger is generated).<br>This bit is the trigger to stop operation of the lower 8-bit timer for TTm1 and TTm3 when channel 1 or 3 is in the 8-bit timer mode. |

**Cautions 1. Be sure to clear bits 15 to 12, 10, and 8 of the TTm register to "0".**

**2. Be sure to clear TT1n (n = 7 to 4) to "0" in the Group B, C, and D products.**

**Remarks 1.** When the TTm register is read, 0 is always read.

**2.** m: Unit number (m = 0, 1),n: Channel number (n = 0 to 7)

**3.** TT1n is not provided in the Group A products.
TT17 to TT14 are not provided in the Group B, C, and D products.

### 6.3.8 Timer input select register 0 (TIS0)

The TIS0 register selects an input source of the timer array unit 0.

Set the TIS0 register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 6-18. Format of Timer Input Select Register 0 (TIS0)**

Address: F0074H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TIS0 | TIS07 Note 1 | TIS06 Note 1 | 0 | TIS04 Note 2 | 0 | TIS02 | TIS01 | TIS00 |

| TIS07 Note 1 | Selection of timer input used with channel 3 of timer array unit 0 |
|---|---|
| 0 | Input signal of timer input pin (TI03) |
| 1 | Event input signal from ELC Note 3 |

| TIS06 Note 1 | Selection of timer input used with channel 2 of timer array unit 0 |
|---|---|
| 0 | Input signal of timer input pin (TI02) |
| 1 | Event input signal from ELC Note 3 |

| TIS04 Note 2 | Selection of timer input used with channel 0 of timer array unit 0 |
|---|---|
| 0 | Input signal of timer input pin (TI00) |
| 1 | Event input signal from ELC Note 3 |

| TIS02 | TIS01 | TIS00 | Selection of timer input used with channel 1 of timer array unit 0 |
|---|---|---|---|
| 0 | 0 | 0 | Input signal of timer input pin (TI01) |
| 0 | 0 | 1 | Event input signal from ELC Note 3 |
| 0 | 1 | 0 | Input signal of timer input pin (TI01) |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | Low-speed on-chip oscillator clock ($f_{IL}$) |
| 1 | 0 | 1 | Sub/low-speed on-chip oscillator select clock ($f_{SL}$) |
| Other than above | | | Setting prohibited |

**Notes 1.** Provided only in products of group E. Write "0" when writing to the timer input select register 0 (TIS0) of the other products.

**2.** Provided only in products of groups D and E. Write "0" when writing to the timer input select register 0 (TIS0) of the other products.

**3.** Provided only in products of groups D and E. Do not set in the other products.

**Cautions 1. When selecting an event input signal from the ELC using timer input select register 0 (TIS0), select $f_{CLK}$ using timer clock select register 0 (TPS0).**

**2. Do not change the select bit of the timer input while inputting data to the TImn pin (m = 0, 1; n = 0 to 7).**

**3. Each of the high-level and low-level widths of the timer input to be selected should be ($1/f_{MCK}$ + 10 ns) or more. So, the TIS02 bit cannot be set to 1 when $f_{SL}$ is selected as $f_{CLK}$ (the CSS bit in the CKC register is set to 1).**

### 6.3.9 Timer input select register 1 (TIS1)

The TIS1 register selects an input source of the timer array unit 0.

The TIS17 and TIS16 bits in the TIS1 register are used in conjunction with the serial array unit to implement the LIN-bus communication operation in channel 7. When the TIS17 and TIS16 bits are set to 1 and 0 respectively, the input signal on the serial data input pin (RxD0) is selected as the timer input.

Set the TIS17 and TIS16 bits at the same time as setting the ISC0 bit in the ISC register (input switch control register).

Set the TIS1 register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 6-19. Format of Timer Input Select Register 1 (TIS1)**

Address: F0075H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TIS1 | TIS17 | TIS16 | 0 | TIS14 | 0 | TIS12 | 0 | TIS10 |

| TIS17 | TIS16 | Selection of timer input used with channel 7 of timer array unit 0 |
|---|---|---|
| 0 | 0 | Input signal of timer input pin (TI07) |
| 0 | 1 | RTC1HZ output signal |
| 1 | 0 | RxD0 pin (detection of the wake-up signal and measurement of the low-level width of the sync break field and the pulse width of the sync field) |
| 1 | 1 | Setting prohibited |

| TIS14 | Selection of timer input used with channel 6 of timer array unit 0 |
|---|---|
| 0 | Input signal of timer input pin (TI06) |
| 1 | RTC1HZ output signal |

| TIS12 | Selection of timer input used with channel 5 of timer array unit 0 |
|---|---|
| 0 | Input signal of timer input pin (TI05) |
| 1 | Input signal of timer input pin (TI03) |

| TIS10 | Selection of timer input used with channel 4 of timer array unit 0 |
|---|---|
| 0 | Input signal of timer input pin (TI04) |
| 1 | Input signal of timer input pin (TI03) |

**Cautions 1. Do not change the select bit of the timer input while inputting data to the TImn pin (m = 0, 1; n = 0 to 7).**

**2. When selecting the RTC1HZ output signal for the clock source of the timer input used in channels 7 and 6 in the TAU, set the TIS17, TIS16, and TIS14 bits to 0, 1, and 1, respectively, and select the RTC1HZ output signal for the timer input of channels 7 and 6.**

**Remark** Set the TIS17 and TIS16 bits to 1 and 0 respectively and select the input signal of the RxD0 pin before using the LIN-bus communication.

### 6.3.10 Timer input select register 2 (TIS2)

The TIS2 register selects an input source of the timer array unit 1.

Set the TIS2 register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

This function is valid only for the Group E products.

**Figure 6-20. Format of Timer Input Select Register 2 (TIS2)**

Address: F007AH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TIS2 | 0 | 0 | 0 | 0 | TIS23 | TIS22 | 0 | 0 |

| TIS22 | Selection of timer input used with channel 6 of timer array unit 1 |
|---|---|
| 0 | Input signal of timer input pin (TI16) |
| 1 | RTC1HZ output signal |

| TIS23 | Selection of timer input used with channel 7 of timer array unit 1 |
|---|---|
| 0 | Input signal of timer input pin (TI17) |
| 1 | RTC1HZ output signal |

**Cautions 1. Do not change the select bit of the timer input while inputting data to the TImn pin (m = 0, 1; n = 0 to 7).**

**2. When selecting the RTC1HZ output signal for the clock source of the timer input used in channels 7 and 6 in the TAU, set the TIS23 and TIS22 bits to 1 and select the RTC1HZ output signal for the timer input of channels 7 and 6.**

### 6.3.11 Timer output enable register m (TOEm)

The TOEm register is used to enable or disable timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TOmn bit of timer output register m (TOm) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TOmn).

The TOEm register can be set by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the TOEm register with a 1-bit or 8-bit memory manipulation instruction with TOEmL.

Reset signal generation clears this register to 0000H.

**Figure 6-21. Format of Timer Output Enable Register m (TOEm)**

Address: F01BAH, F01BBH (TOE0), F01FAH, F01FBH (TOE1) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TOEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TOEm7 | TOEm6 | TOEm5 | TOEm4 | TOEm3 | TOEm2 | TOEm1 | TOEm0 |

| TOEmn | Timer output enable/disable of channel n |
|---|---|
| 0 | Timer output is disabled.<br>Timer operation is not applied to the TOmn bit and the output is fixed.<br>Writing to the TOmn bit is enabled. |
| 1 | Timer output is enabled.<br>Timer operation is applied to the TOmn bit and an output waveform is generated.<br>Writing to the TOmn bit is ignored. |

**Caution Be sure to clear bits 15 to 8 to “0”. (unit 1, 0: Group E products, unit 0: Group A products)**
**Be sure to clear bits 15 to 8 of unit 0 and bits 15 to 4 of unit 1 to "0". (Group B, C, and D products)**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** TOE1n is not provided in the Group A products.
TOE17 to TOE14 are not provided in the Group B, C, and D products.

### 6.3.12 Timer output register m (TOm)

The TOm register is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TOmn) of each channel.

The TOmn bit oh this register can be rewritten by software only when timer output is disabled (TOEmn = 0). When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

Due to pin arrangement, when using the pins shared by TImn and TOmn as port pins, set the corresponding TOmn bit to “0”.

The TOm register can be set by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the TOm register with an 8-bit memory manipulation instruction with TOmL.

Reset signal generation clears this register to 0000H.

**Figure 6-22. Format of Timer Output Register m (TOm)**

Address: F01B8H, F01B9H (TO0), F01F8H, F01F9H (TO1) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TOm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TOm7 | TOm6 | TOm5 | TOm4 | TOm3 | TOm2 | TOm1 | TOm0 |

| TOmn | Timer output of channel n |
|---|---|
| 0 | Timer output value is “0”. |
| 1 | Timer output value is “1”. |

**Caution Be sure to clear bits 15 to 8 to “0”. (unit 1, 0: Group E products, unit 0: Group A products) Be sure to clear bits 15 to 8 of unit 0 and bits 15 to 4 of unit 1 to "0". (Group B, C, and D products)**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** TO1n is not provided in the Group A products.
TO17 to TO14 are not provided in the Group B, C, and D products.

### 6.3.13 Timer output level register m (TOLm)

The TOLm register is a register that controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOEmn = 1) in the Slave channel output mode (TOMmn = 1). In the master channel output mode (TOMmn = 0), this register setting is invalid.

The TOLm register can be set by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the TOLm register with an 8-bit memory manipulation instruction with TOLmL.

Reset signal generation clears this register to 0000H.

**Figure 6-23. Format of Timer Output Level Register m (TOLm)**

Address: F01BCH, F01BDH (TOL0), F01FCH, F01FDH (TOL1) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TOLm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TOLm7 | TOLm6 | TOLm5 | TOLm4 | TOLm3 | TOLm2 | TOLm1 | 0 |

| TOLmn | Control of timer output level of channel n |
|---|---|
| 0 | Non-inverted output (active-high) |
| 1 | Inverted output (active-low) |

**Caution Be sure to clear bits 15 to 8 and 0 to “0”. (unit 1, 0: Group E products, unit 0: Group A products) Be sure to clear bits 15 to 8 and 0 of unit 0 and bits 15 to 4 and 0 of unit 1 to "0". (Group B, C, and D products)**

**Remarks 1.** If the value of this register is rewritten during timer operation, the timer output logic is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**3.** TOL1n is not provided in the Group A products.
TOL17 to TOL14 are not provided in the Group B, C, and D products.

### 6.3.14 Timer output mode register m (TOMm)

The TOMm register is used to control the timer output mode of each channel.

When a channel is used for the independent channel operation function, set the corresponding bit of the channel to be used to 0.

When a channel is used for the simultaneous channel operation function (PWM output, one-shot pulse output, or multiple PWM output), set the corresponding bit of the master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOEmn = 1).

The TOMm register can be set by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the TOMm register with an 8-bit memory manipulation instruction with TOMmL.

Reset signal generation clears this register to 0000H.

**Figure 6-24. Format of Timer Output Mode Register m (TOMm)**

Address: F01BEH, F01BFH (TOM0), F01FEH, F01FFH (TOM1) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TOMm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TOMm7 | TOMm6 | TOMm5 | TOMm4 | TOMm3 | TOMm2 | TOMm1 | 0 |

| TOMmn | Control of timer output mode of channel n |
|---|---|
| 0 | Master channel output mode (to produce toggle output by timer interrupt request signal (INTTMmn)) |
| 1 | Slave channel output mode (output is set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTM0p) of the slave channel) |

**Caution Be sure to clear bits 15 to 8 and 0 to “0”. (unit 1, 0: Group E products, unit 0: Group A products) Be sure to clear bits 15 to 8 and 0 of unit 0 and bits 15 to 4 and 0 of unit 1 to "0". (Group B, C, and D products)**

**Remarks 1.** m: Unit number (m = 0, 1)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7 (For details of the relation between the master channel and slave channel, refer to **6.4.1 Basic rules of simultaneous channel operation function**.)

**2.** TOM1n is not provided in the Group A products.
TOM17 to TOM14 are not provided in the Group B, C, and D products.

### 6.3.15 Noise filter enable registers 1, 2 (NFEN1, NFEN2)

The NFEN1, NFEN2 registers is used to set whether the noise filter can be used for the timer input signal to each channel.

Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal.

When the noise filter is ON, match detection and synchronization of the 2 clocks is performed with the CPU/peripheral hardware clock ($f_{MCK}$). When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock ($f_{MCK}$) **Note**.

Set the NFEN1 and NFEN2 registers by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Note** For details, see **6.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)** and **6.5.2 Start timing of counter**.

**Figure 6-25. Format of Noise Filter Enable Register 1 (NFEN1) (1/2)**

Address: F0071H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| NFEN1 | TNFEN07 | TNFEN06 | TNFEN05 | TNFEN04 | TNFEN03 | TNFEN02 | TNFEN01 | TNFEN00 |

| TNFEN07 | Enable/disable using noise filter of TI07 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN06 | Enable/disable using noise filter of TI06 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN05 | Enable/disable using noise filter of TI05 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN04 | Enable/disable using noise filter of TI04 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN03 | Enable/disable using noise filter of TI03 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN02 | Enable/disable using noise filter of TI02 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN01 | Enable/disable using noise filter of TI01 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN00 | Enable/disable using noise filter of TI00 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

**Caution The pin to be used can be changed by setting the TIS17 and TIS16 bits in the timer input select register 1 (TIS1).**
**When TIS17, TIS16 = 0, 0: The use of the noise filter of the TI07 pin can be enabled or disabled.**
**When TIS17, TIS16 = 1, 0: The use of the noise filter of the RxD0 pin can be enabled or disabled.**

**Remark** The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

**Figure 6-25. Format of Noise Filter Enable Register 2 (NFEN2) (2/2)**

Address: F0072H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| NFEN2 | TNFEN17 | TNFEN16 | TNFEN15 | TNFEN14 | TNFEN13 | TNFEN12 | TNFEN11 | TNFEN10 |

| TNFEN17 | Enable/disable using noise filter of TI17 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN16 | Enable/disable using noise filter of TI16 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN15 | Enable/disable using noise filter of TI15 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN14 | Enable/disable using noise filter of TI14 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN13 | Enable/disable using noise filter of TI13 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN12 | Enable/disable using noise filter of TI12 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN11 | Enable/disable using noise filter of TI11 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

| TNFEN10 | Enable/disable using noise filter of TI00 pin input signal |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |

**Remarks 1.** The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

**2.** NFEN2 is not provided in the Group A products.
Bits 7 to 4 are not provided in the Group B, C, and D products.

### 6.3.16 Port mode registers 1, 3, 4, 7, 12 (PM1, PM3, PM4, PM7, PM12)

These registers set input/output of ports 1, 3, 4, 7, 12 in 1-bit units.

The presence or absence of timer I/O pins depends on the product. When using the timer array unit, set the following port mode registers according to the product used.

Group A and B products: PM1, PM3, PM12
Group C and D products: PM1, PM3, PM4, PM12
Group E products: PM1, PM3, PM4, PM7, PM12

When using the ports (such as P17/TO00/TI00 and P16/TO2/TI02) to be shared with the timer output pin for timer output, set the port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to “0”.

Example: When using P16/TO02/TI02 for timer output
Set the PM16 bit of port mode register 1 to 0.
Set the P16 bit of port register 1 to 0.

When using the ports (such as P17/TO00/TI00 and P16/TO2/TI02) to be shared with the timer output pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P16/TO02/TI02 for timer input
Set the PM16 bit of port mode register 1 to 1.
P16 bit of port register may be 0 or 1.

Set the PM1, PM3, PM4, PM7, and PM12 registers by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

**Figure 6-26. Format of Port Mode Registers 1, 3, 4, 7, 12 (PM1, PM3, PM4, PM7, PM12) (100-pin products)**

Address: FFF21H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM1 | PM17 | PM16 | PM15 | PM14 | PM13 | PM12 | PM11 | PM10 |

Address: FFF23H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM3 | 1 | 1 | 1 | PM34 | PM33 | PM32 | PM31 | PM30 |

Address: FFF24H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM4 | PM47 | PM46 | PM45 | PM44 | PM43 | PM42 | PM41 | PM40 |

Address: FFF27H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM7 | PM77 | PM76 | PM75 | PM74 | PM73 | PM72 | PM71 | PM70 |

Address: FFF2CH After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM12 | PM127 | PM126 | PM125 | 1 | 1 | 1 | 1 | PM120 |

| PMmn | Pmn pin I/O mode selection (m = 1, 3, 4, 7, 12; n = 0 to 7) |
|---|---|
| 0 | Output mode (output buffer on) |
| 1 | Input mode (output buffer off) |

**Remark** The figure shown above presents the format of port mode registers 1, 3, 4, 7, and 12 of the 100-pin products. The format of the port mode register of other products, see **CHAPTER 4 PORT FUNCTIONS**.

### 6.3.17 PWM output delay control register 1 (PWMDLY1)

This register controls output delay of PWM output signal output from the TO0n pin.

Set the PWMDLY1 register by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Address: F022BH After reset: 00H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| PWMDLY1 | TO071 | TO070 | TO061 | TO060 | TO051 | TO050 | TO041 | TO040 |

Address: F022AH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PWMDLY1 | TO031Note | TO030Note | TO021Note | TO020Note | TO011 | TO010 | 0 | 0 |

| TO0n1 | TO0n0 | PWM output delay control of timer array unit 0 TO0n |
|---|---|---|
| 0 | 0 | No delay |
| 0 | 1 | Delayed by one cycle of the CPU/peripheral hardware clock ($f_{CLK}$). |
| 1 | 0 | Delayed by two cycles of the CPU/peripheral hardware clock ($f_{CLK}$). |
| 1 | 1 | Delayed by three cycles of the CPU/peripheral hardware clock ($f_{CLK}$). |

**Note** If this register is set for a delay, this affects PWM output of TO0n, but doesn't affect the operation of the timer output signal to peripheral functions.

**Remark** n: Channel number (n = 1 to 7)

**Cautions**
1. **Set this register before outputting a PWM output signal (do not change the setting during operation).**
2. **Set this register with a 16-bit memory manipulation instruction. Do not set this register with a 1-bit or 8-bit memory manipulation instruction.**
3. **If this register is not used for PWM output, it should be cleared to 0. This is because the timer output is delayed.**
4. **When setting this register after the PWM output is stopped, wait for four cycles of the CPU/peripheral hardware clock ($f_{CLK}$) before the setting.**
5. **Even if this register is set for a delay, this doesn't affect the operation of other pin functions multiplexed on the same pin as the TO0n pin function (n = 1 to 7).**

### 6.3.18 PWM output delay control register 2 (PWMDLY2)

This register controls output delay of PWM output signal output from the TO1n pin.

Set the PWMDLY2 register by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Address: F022DH After reset: 00H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| PWMDLY2 | TO171 | TO170 | TO161 | TO160 | TO151 | TO150 | TO141 | TO140 |

Address: F022CH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PWMDLY2 | TO131 | TO130 | TO121 | TO120 | TO111 | TO110 | 0 | 0 |

| TO1n1 | TO1n0 | PWM output delay control of timer array unit 1 TO1n |
|---|---|---|
| 0 | 0 | No delay |
| 0 | 1 | Delayed by one cycle of the CPU/peripheral hardware clock ($f_{CLK}$). |
| 1 | 0 | Delayed by two cycles of the CPU/peripheral hardware clock ($f_{CLK}$). |
| 1 | 1 | Delayed by three cycles of the CPU/peripheral hardware clock ($f_{CLK}$). |

**Remarks 1.** n: Channel number (n = 1 to 7)

**2.** Bits 15 to 0 are not provided in the Group A products.
Bits 15 to 8 are not provided in the Group B, C, and D products.

**Cautions 1. Set this register before outputting a PWM output signal (do not change the setting during operation).**

**2. Set this register with a 16-bit memory manipulation instruction. Do not set this register with a 1-bit or 8-bit memory manipulation instruction.**

**3. If this register is not used for PWM output, it should be cleared to 0. This is because the timer output is delayed.**

**4. When setting this register after the PWM output is stopped, wait for four cycles of the CPU/peripheral hardware clock ($f_{CLK}$) before the setting.**

**5. Even if this register is set for a delay, this doesn't affect the operation of other pin functions multiplexed on the same pin as the TO1n pin function (n = 1 to 7).**

## 6.4 Basic Rules of Timer Array Unit

### 6.4.1 Basic rules of simultaneous channel operation function

When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

(1) Only an even channel (channel 0, 2, 4, 6) can be set as a master channel.
(2) Any channel, except channel 0, can be set as a slave channel.
(3) The slave channel must be lower than the master channel.
Example: If channel 2 is set as a master channel, channel 3 or those that follow (channels 3, 4, 5, …) can be set as a slave channel.

(4) Two or more slave channels can be set for one master channel.
(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example: If channels 0 and 4 are set as master channels, channels 1 to 3 can be set as the slave channel of master channel 0. Channels 5 to 7 cannot be set as the slave channel of master channel 0.

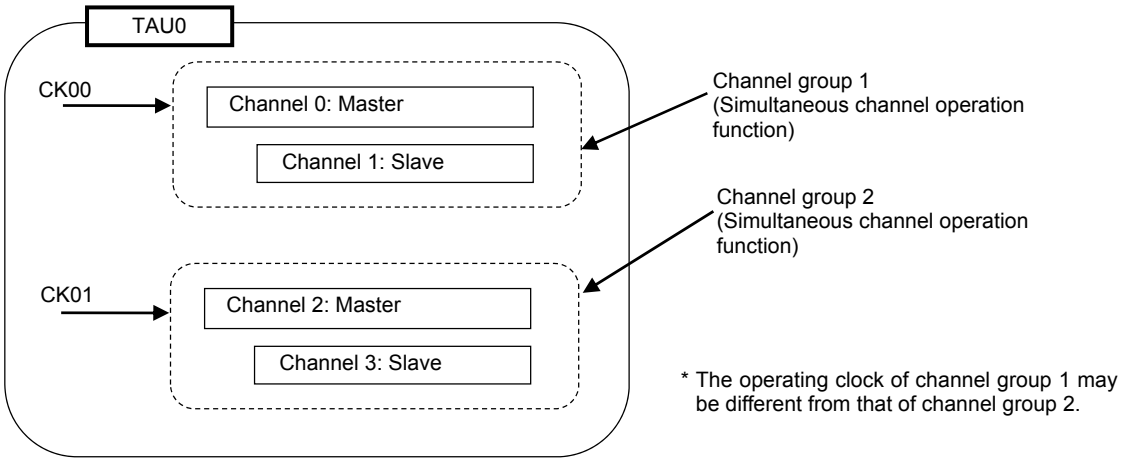
(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKSmn0, CKSmn1 bits (bit 15, 14 of timer mode register mn (TMRmn)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
(7) A master channel can transmit INTTMmn (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use INTTMmn (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or count clock to channels with lower channel numbers.
(9) A master channel cannot use INTTMmn (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.
(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TSmn) of the channels in combination must be set at the same time.
(11) During the counting operation, a TSmn bit of a master channel or TSmn bits of all channels which are operating simultaneously can be set. It cannot be applied to TSmn bits of slave channels alone.
(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TTmn) of the channels in combination must be set at the same time.
(13) Timer mode register m0 (TMRm0) has no master bit (it is fixed as “0”). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).

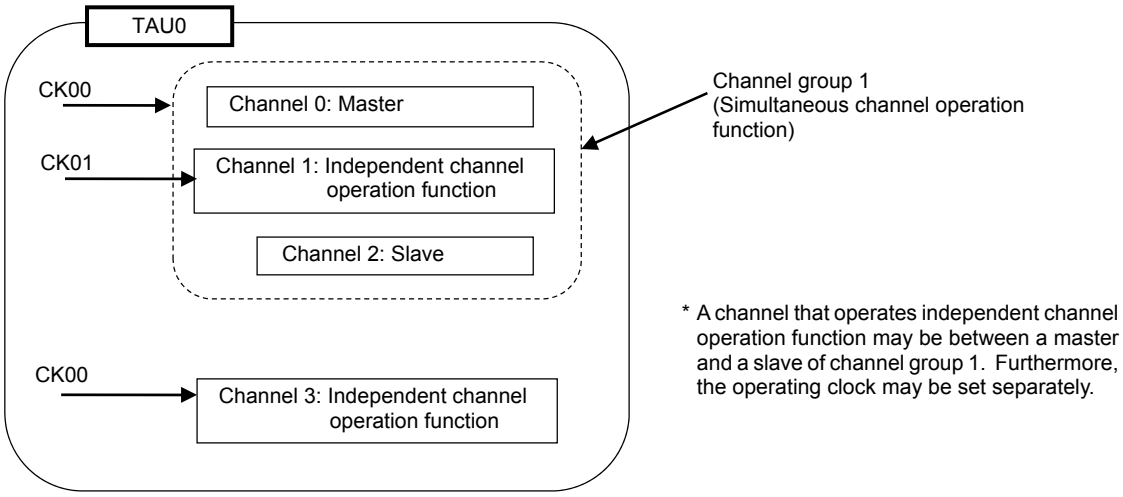
If two or more channel groups that do not operate in combination are specified, the basic rules described above do not apply to the channel groups.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)
**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

Example 1




Example 2



### 6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

(1) The 8-bit timer operation function applies only to channels 1 and 3.
(2) When using 8-bit timers, set the SPLIT bit of timer mode register mn (TMRmn) to 1.
(3) The higher 8 bits can be operated as the interval timer function.
(4) At the start of operation, the higher 8 bits output INTTMm1H/INTTMm3H (an interrupt) (which is the same operation performed when MDmn0 is set to 1).
(5) The operation clock of the higher 8 bits is selected according to the CKSmn1 and CKSmn0 bits of the lower-bit TMRmn register.
(6) For the higher 8 bits, the TSHm1/TSHm3 bit is manipulated to start channel operation and the TTHm1/TTHm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEHm1/TEHm3 bit.
(7) The lower 8 bits operate according to the TMRmn register settings. The following three functions support operation of the lower 8 bits:
- Interval timer function
- External event counter function
- Delay count function

(8) For the lower 8 bits, the TSm1/TSm3 bit is manipulated to start channel operation and the TTm1/TTm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEm1/TEm3 bit.
(9) During 16-bit operation, manipulating the TSHm1, TSHm3, TTHm1, and TTHm3 bits is invalid. The TSm1, TSm3, TTm1, and TTm3 bits are manipulated to operate channels 1 and 3. The TEHm3 and TEHm1 bits are not changed.
(10) For the 8-bit timer function, the simultaneous operation functions (one-shot pulse, PWM, and multiple PWM) cannot be used.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 1, 3)
**2.** Unit 1 is not provided in the Group A products.

## 6.5 Operation Timing of Counter

### 6.5.1 Count clock ($f_{TCLK}$)

The count clock ($f_{TCLK}$) of the timer array unit can be selected between following by CCSmn bit of timer mode register mn (TMRmn).

- Operation clock ($f_{MCK}$) specified by the CKSmn0 and CKSmn1 bits
- Valid edge of input signal input from the TImn pin

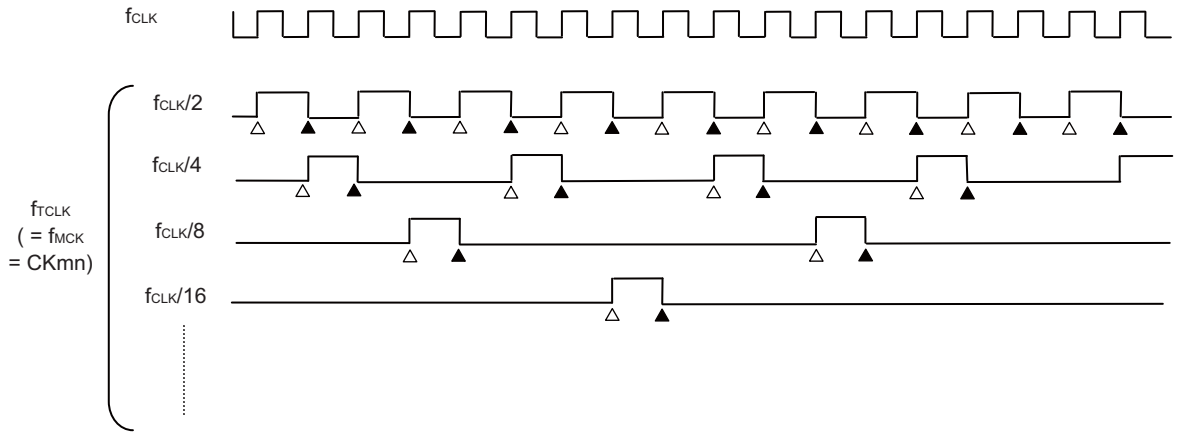
Because the timer array unit is designed to operate in synchronization with $f_{CLK}$, the timings of the count clock ($f_{TCLK}$) are shown below.

**(1) When operation clock ($f_{MCK}$) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 0)**

The count clock ($f_{TCLK}$) is between $f_{CLK}$ to $f_{CLK}/2^{15}$ by setting of timer clock select register m (TPSm). When a divided $f_{CLK}$ is selected, however, the clock is a signal which becomes high level for one period of $f_{CLK}$ from its rising edge. When $f_{CLK}$ is selected, the clock is fixed high.

Counting of timer count register mn (TCRmn) delayed by one period of $f_{CLK}$ from rising edge of the count clock, because of synchronization with $f_{CLK}$. But, this is described as “counting at rising edge of the count clock”, as a matter of convenience.

**Figure 6-27. Timing of $f_{CLK}$ and count clock ($f_{TCLK}$) (When CCSmn = 0)**



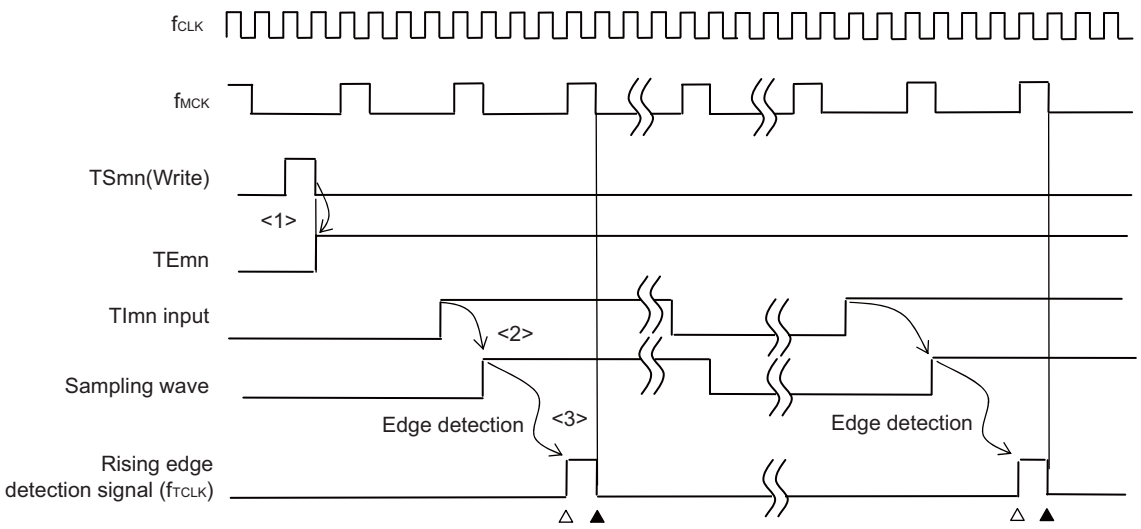

Remarks 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter
2. $f_{CLK}$: CPU/peripheral hardware clock

**(2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)**

The count clock ($f_{TCLK}$) becomes the signal that detects valid edge of input signal via the TImn pin and synchronizes next rising $f_{MCK}$. The count clock ($f_{TCLK}$) is delayed for 1 to 2 period of $f_{MCK}$ from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 to 4 clock).

Counting of timer count register mn (TCRmn) delayed by one period of $f_{CLK}$ from rising edge of the count clock, because of synchronization with $f_{CLK}$. But, this is described as “counting at valid edge of input signal via the TImn pin”, as a matter of convenience.

**Figure 6-28. Timing of count clock ($f_{TCLK}$) (When CCSmn = 1, noise filter unused)**




<1> Setting TSmn bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TImn pin.

<2> The rise of input signal via the TImn pin is sampled by $f_{MCK}$.

<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

**Remarks 1.** △: Rising edge of the count clock
▲: Synchronization, increment/decrement of counter

**2.** $f_{CLK}$: CPU/peripheral hardware clock
$f_{MCK}$: Operation clock of channel n

**3.** The waveform of the input signal via TImn pin of the input pulse interval measurement, the measurement of high/low width of input signal, and the delay counter, and the one-shot pulse output are the same as that shown in Figure 6-28.

### 6.5.2 Start timing of counter

Timer count register mn (TCRmn) becomes enabled to operation by setting of TSmn bit of timer channel start register m (TSm).

Operations from count operation enabled state to timer count Register mn (TCRmn) count start is shown in Table 6-6.

**Table 6-6. Operations from Count Operation Enabled State to Timer count Register mn (TCRmn) Count Start**

| Timer operation mode | Operation when TSmn = 1 is set |
|---|---|
| • Interval timer mode | No operation is carried out from start trigger detection (TSmn=1) until count clock generation.<br>The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3 (1) Start timing in interval timer mode**). |
| • Event counter mode | Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register.<br>The subsequent count clock performs count down operation.<br>The external trigger detection selected by the STSmn2 to STSmn0 bits in the TMRmn register does not start count operation (see **6.5.3 (2) Start timing in event counter mode**). |
| • Capture mode | No operation is carried out from start trigger detection until count clock generation.<br>The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3 (3) Start timing in capture mode**). |
| • One-count mode | The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0).<br>No operation is carried out from start trigger detection until count clock generation.<br>The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3 (4) Start timing in one-count mode**). |
| • Capture & one-count mode | The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0).<br>No operation is carried out from start trigger detection until count clock generation.<br>The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3 (5) Start timing in capture & one-count mode (when high-level width is measured)**). |

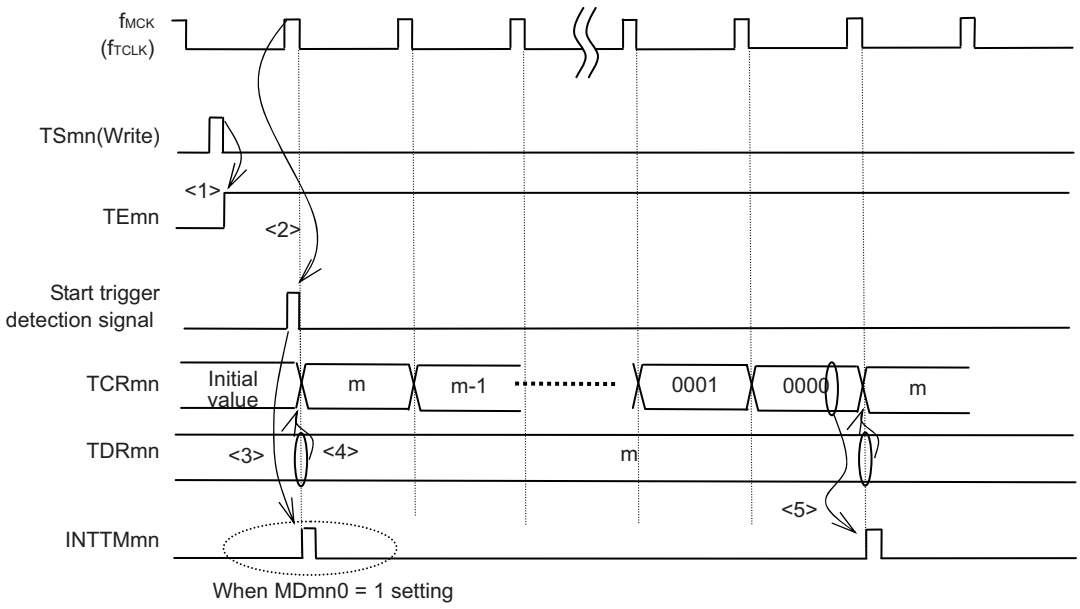
### 6.5.3 Operation of counter

Here, the counter operation in each mode is explained.

#### (1) Start timing in interval timer mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit. Timer count register mn (TCRmn) holds the initial value until count clock generation.
<2> A start trigger is generated at the first count clock after operation is enabled.
<3> When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.
<4> By the first count clock after the operation enable, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting starts in the interval timer mode.
<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting keeps on.

**Figure 6-29. Start Timing (In Interval Timer Mode)**



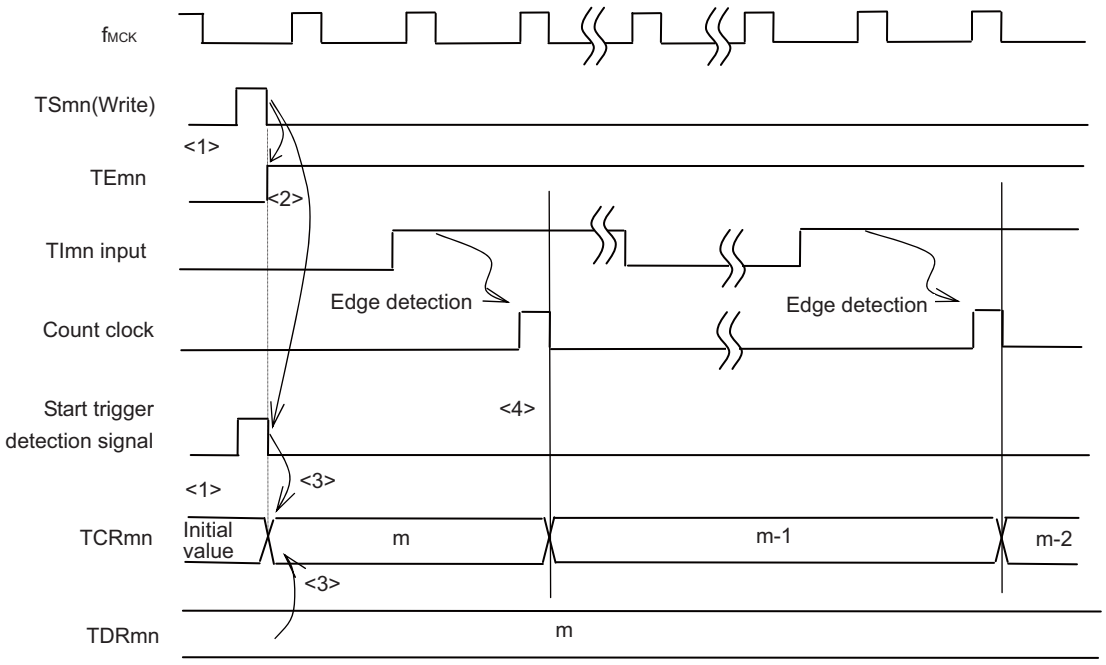

**Caution In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

**Remark** $f_{MCK}$, the start trigger detection signal, and INTTMmn become active between one clock in synchronization with $f_{CLK}$.

**(2) Start timing in event counter mode**

<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).
<2> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<3> As soon as 1 has been written to the TSmn bit and 1 has been set to the TEmn bit, the value of timer data register mn (TDRmn) is loaded to the TCRmn register to start counting.
<4> After that, the TCRmn register value is counted down according to the count clock of the valid edge of the TImn input .

**Figure 6-30. Start Timing (In Event Counter Mode)**

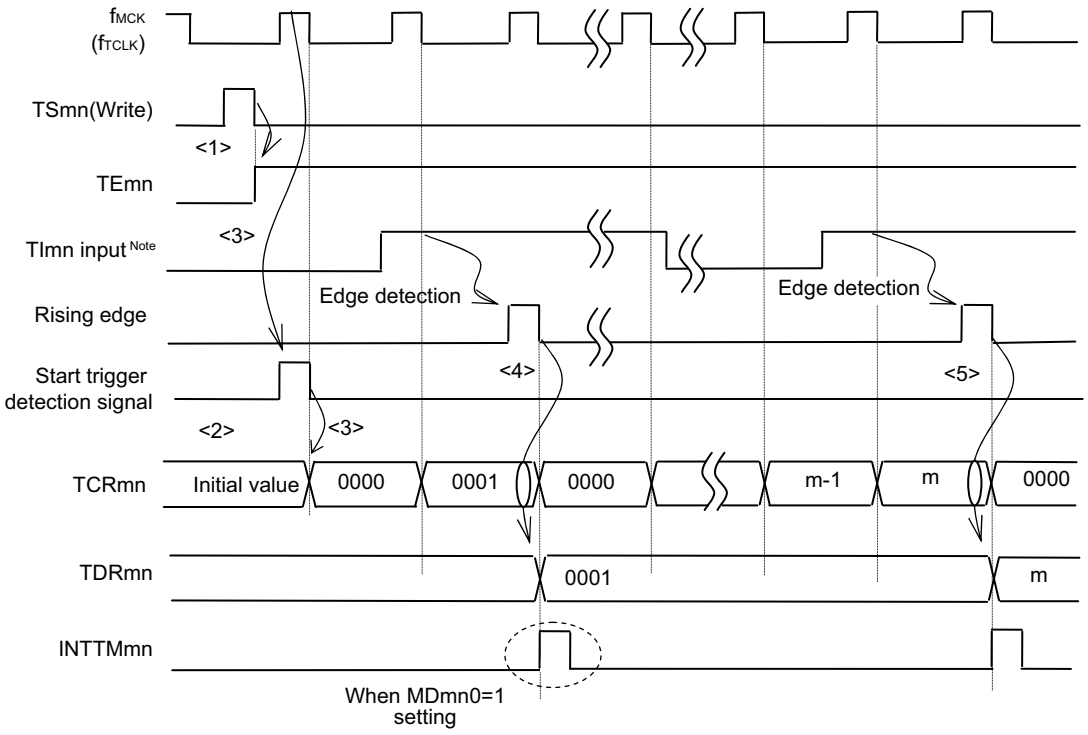



**Remark** The timing is shown in Figure 6-30 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 $f_{MCK}$ cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input.

**(3) Start timing in capture mode**

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<2> Timer count register mn (TCRmn) holds the initial value until count clock generation.
<3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0000H is loaded to the TCRmn register and counting starts in the capture mode. (When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.)
<4> On detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated. However, this capture value is no meaning. The TCRmn register keeps on counting from 0000H.
<5> On next detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

**Figure 6-31. Operation Timing (In Capture Mode : Input Pulse Interval Measurement)**




**Note** If a clock has been input to TImn (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

**Caution In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

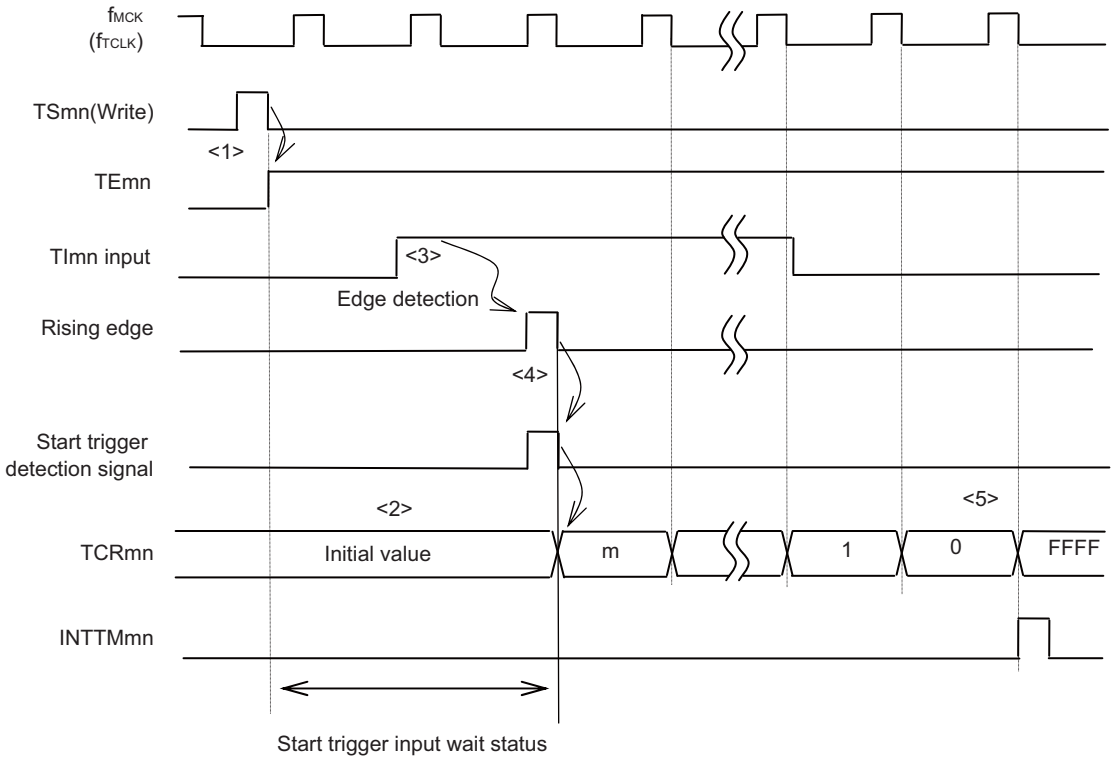
**Remark** The timing is shown in Figure 6-31 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 $f_{MCK}$ cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input.

**(4) Start timing in one-count mode**

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.
<3> Rising edge of the TImn input is detected.
<4> On start trigger detection, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and count starts.
<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of the TCRmn register becomes FFFFH and counting stops

.

**Figure 6-32. Start Timing (In One-count Mode)**

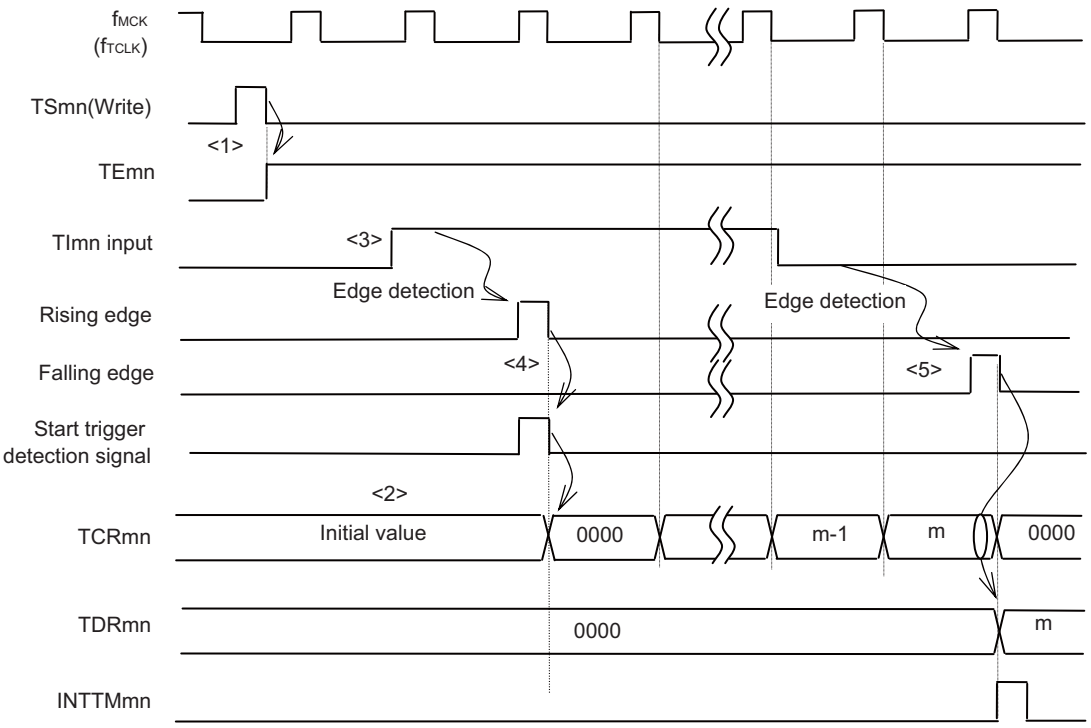



**Remark** The timing is shown in Figure 6-32 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

**(5) Start timing in capture & one-count mode (when high-level width is measured)**

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit of timer channel start register m (TSm).
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.
<3> Rising edge of the TImn input is detected.
<4> On start trigger detection, the value of 0000H is loaded to the TCRmn register and count starts.
<5> On detection of the falling edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

**Figure 6-33. Start Timing (In Capture & One-count Mode)**

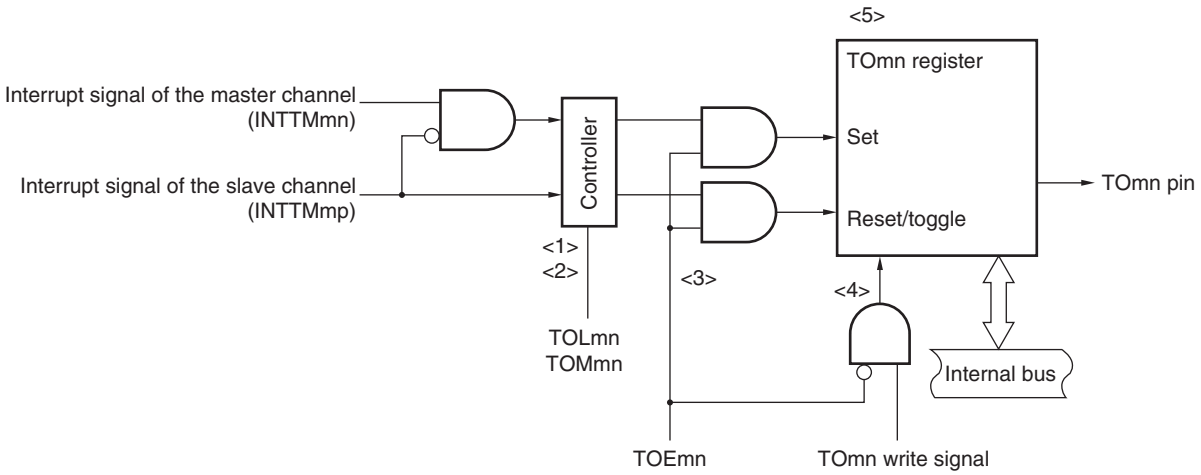



**Remark** The timing is shown in Figure 6-33 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs because of the asynchronous relationship between the period of the TImn input and that of the count clock (fMCK).

## 6.6 Channel Output (TOmn pin) Control

### 6.6.1 TOmn pin output circuit configuration

**Figure 6-34. Output Circuit Configuration**




The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (master channel output mode), the set value of timer output level register m (TOLm) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register m (TOm).

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:

| | |
|---|---|
| When TOLmn = 0: | Forward operation (INTTMmn → set, INTTM0p → reset) |
| When TOLmn = 1: | Reverse operation (INTTMmn → reset, INTTM0p → set) |

When INTTMmn and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register. Writing to the TOm register (TOmn write signal) becomes invalid.
When TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set timer operation is stopped (TOEmn = 0) and to write a value to the TOm register.

<4> While timer output is disabled (TOEmn = 0), writing to the TOmn bit to the target channel (TOmn write signal) becomes valid. When timer output is disabled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TOm register.

<5> The TOm register can always be read, and the TOmn pin output level can be checked.

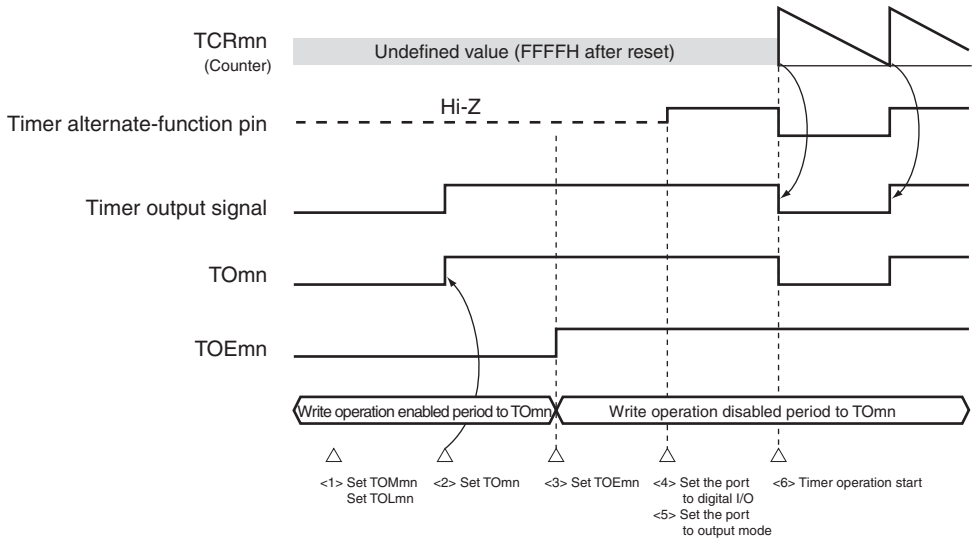
**Remarks 1.** m: Unit number (m = 0, 1)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
$n < p \leq 7$

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

### 6.6.2 TOmn Pin Output Setting

The following figure shows the procedure and status transition of the TOmn output pin from initial setting to timer operation start.

**Figure 6-35. Status Transition from Timer Output Setting to Operation Start**




<1> The operation mode of timer output is set.

- TOMmn bit (0: Master channel output mode, 1: Slave channel output mode)
- TOLmn bit (0: Non-inverted output, 1: Inverted output)

<2> The timer output signal is set to the initial status by setting timer output register m (TOm).

<3> The timer output operation is enabled by writing 1 to the TOEmn bit (writing to the TOm register is disabled).

<4> The port mode control registers (PMCxx) are used to set the use of port pins for digital I/O (see **4.3.6 Port mode control registers 7, 9, 12 (PMC7, PMC9, PMC12)**).

<5> The port I/O setting is set to output (see **6.3.16 Port mode registers 1, 3, 4, 7, 12 (PM1, PM3, PM4, PM7, PM12)**).

<6> The timer operation is enabled (TSmn = 1).

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

### 6.6.3 Cautions on Channel Output Operation

#### (1) Changing values set in the registers TOm, TOEm, TOLm, and TOMm during timer operation

Since the timer operations (operations of timer count register mn (TCRmn) and timer data register mn (TDRmn)) are independent of the TOmn output circuit and changing the values set in timer output register m (TOm), timer output enable register m (TOEm), timer output level register m (TOLm), and timer output mode register m (TOMm) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TOmn pin by timer operation, however, set the TOm, TOEm, TOLm, and TOMm registers to the values stated in the register setting example of each operation shown by 6.7 and 6.8.

When the values set to the TOEm, TOLm, and TOMm registers (but not the TOm register) are changed close to the occurrence of the timer interrupt (INTTMmn) of each channel, the waveform output to the TOmn pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMmn) occurs.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

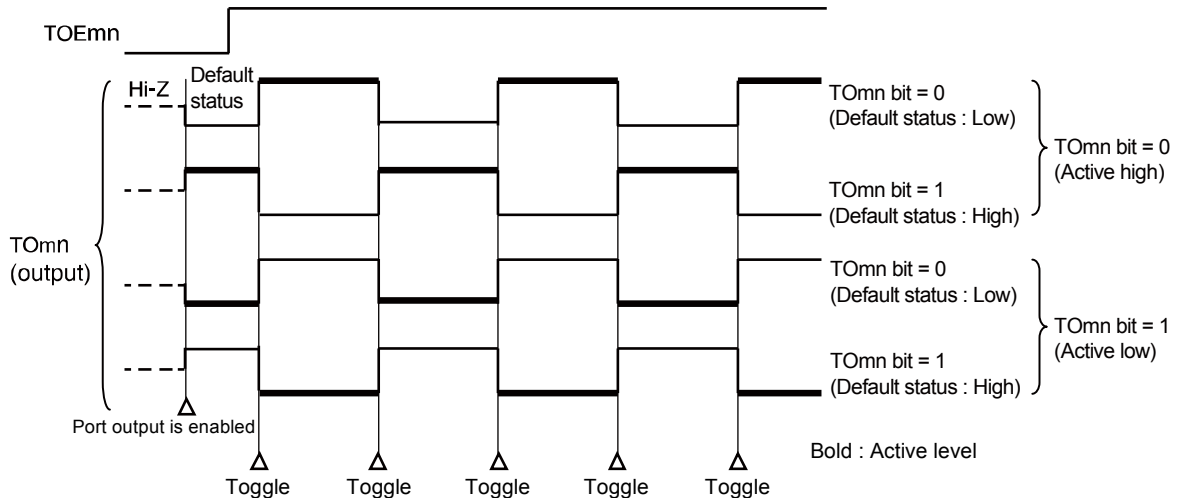
### (2) Default level of TOmn pin and output level after timer operation start

The change in the output level of the TOmn pin when timer output register m (TOm) is written while timer output is disabled (TOEmn = 0), the initial level is changed, and then timer output is enabled (TOEmn = 1) before port output is enabled, is shown below.

#### (a) When operation starts with master channel output mode (TOMmn = 0) setting

The setting of timer output level register m (TOLm) is invalid when master channel output mode (TOMmn = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TOmn pin is reversed.

**Figure 6-36. TOmn Pin Output Status at Toggle Output (TOMmn = 0)**




**Remarks 1.** Toggle: Reverse TOmn pin output status

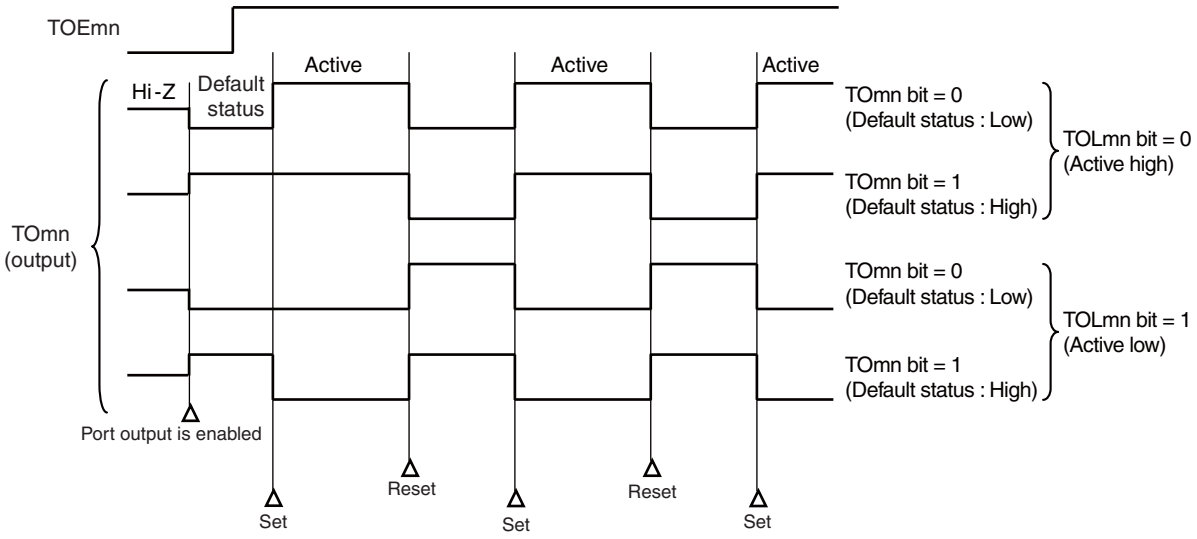
**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**3.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**(b) When operation starts with slave channel output mode (TOMmn = 1) setting (PWM output))**

When slave channel output mode (TOMmn = 1), the active level is determined by timer output level register m (TOLm) setting.

**Figure 6-37. TOmn Pin Output Status at PWM Output (TOMmn = 1)**




**Remarks 1.** Set: The output signal of the TOmn pin changes from inactive level to active level.
Reset: The output signal of the TOmn pin changes from active level to inactive level.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**3.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

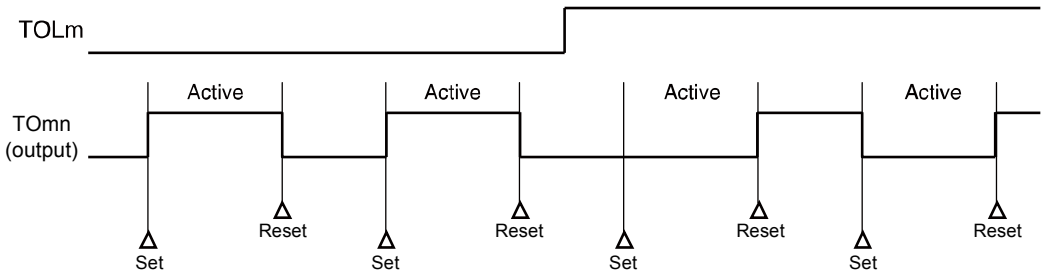
### (3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

#### (a) When timer output level register m (TOLm) setting has been changed during timer operation

When the TOLm register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TOmn pin change condition. Rewriting the TOLm register does not change the output level of the TOmn pin.

The operation when TOMmn is set to 1 and the value of the TOLm register is changed while the timer is operating (TEmn = 1) is shown below.

**Figure 6-38. Operation when TOLm Register Has Been Changed during Timer Operation**




**Remarks 1.** Set: The output signal of the TOmn pin changes from inactive level to active level.
Reset: The output signal of the TOmn pin changes from active level to inactive level.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**3.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

#### (b) Set/reset timing

To realize 0%/100% output at PWM output, the TOmn pin/TOmn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

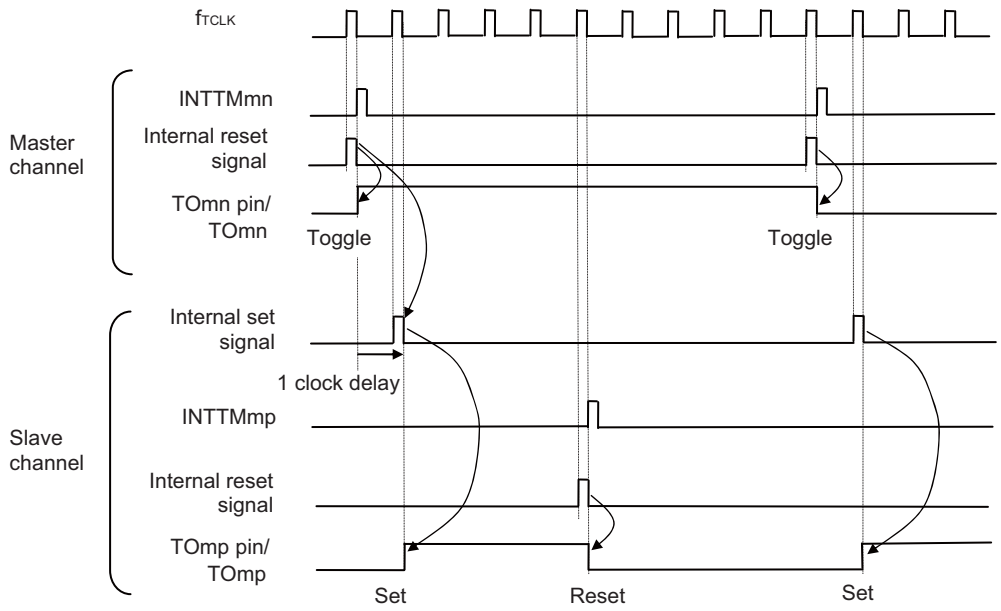
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-39 shows the set/reset operating statuses where the master/slave channels are set as follows.

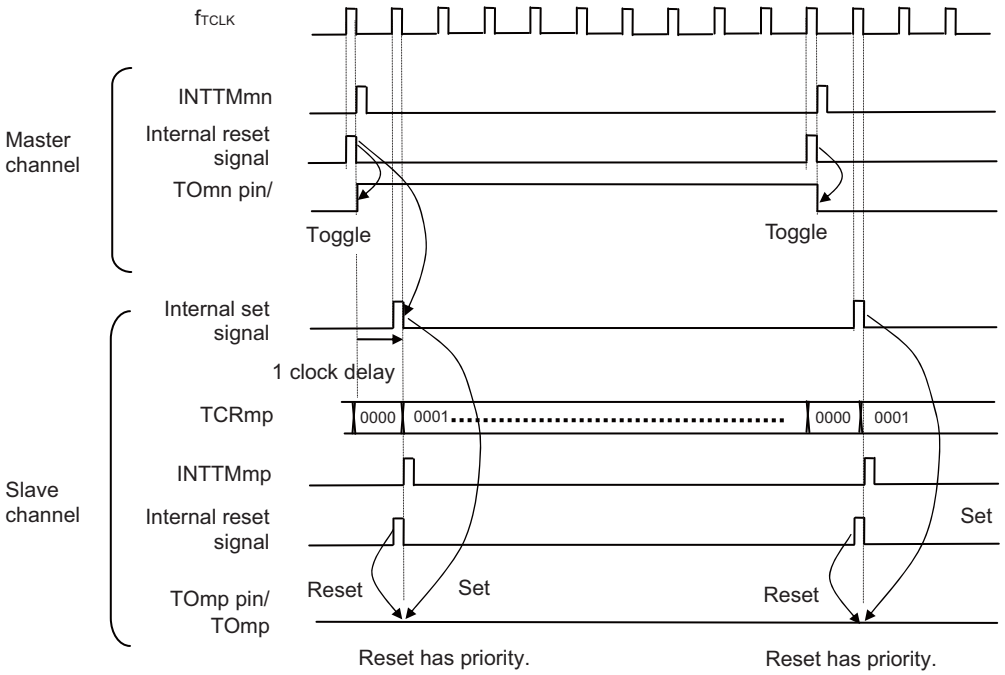
Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

**Figure 6-39. Set/Reset Timing Operating Statuses**

**(1) Basic operation timing**




**(2) Operation timing when 0 % duty**




**Remarks 1.** Internal reset signal: TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

**2.** m: Unit number (m = 0, 1)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

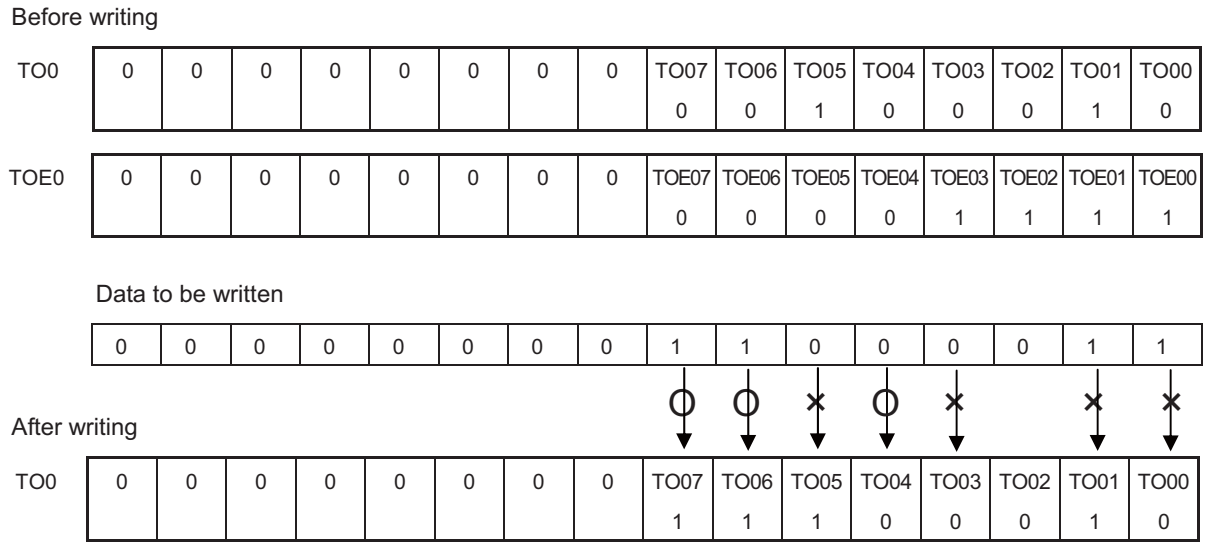
**3.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

### 6.6.4 Collective manipulation of TOmn bit

In timer output register m (TOm), the setting bits for all the channels are located in one register in the same way as timer channel start register m (TSm). Therefore, the TOmn bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TOmn bits (TOEmn = 0) that correspond to the relevant bits of the channel used to perform output (TOmn).

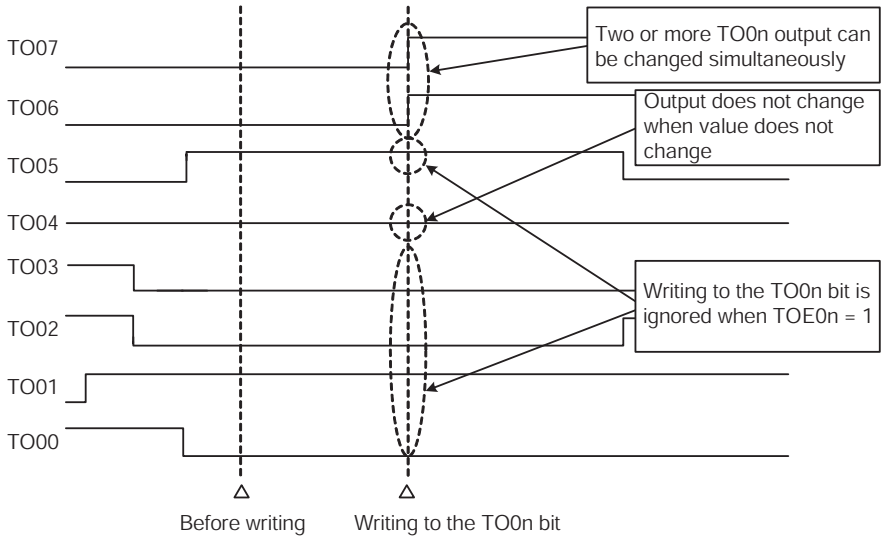
**Figure 6-40. Example of TO0n Bit Collective Manipulation**




Writing is done only to the TOmn bit with TOEmn = 0, and writing to the TOmn bit with TOEmn = 1 is ignored.

TOmn (channel output) to which TOEmn = 1 is set is not affected by the write operation. Even if the write operation is done to the TOmn bit, it is ignored and the output change by timer operation is normally done.

**Figure 6-41. TO0n Pin Statuses by Collective Manipulation of TO0n Bit**




(**Caution** and **Remarks** are given on the next page.)

**Caution While timer output is enabled (TOEmn = 1), even if the output by timer interrupt of each timer (INTTMmn) contends with writing to the TOmn bit, output is normally done to the TOmn pin.**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

### 6.6.5 Timer Interrupt and TOmn Pin Output at Operation Start

In the interval timer mode or capture mode, the MDmn0 bit in timer mode register mn (TMRmn) sets whether or not to generate a timer interrupt at count start.

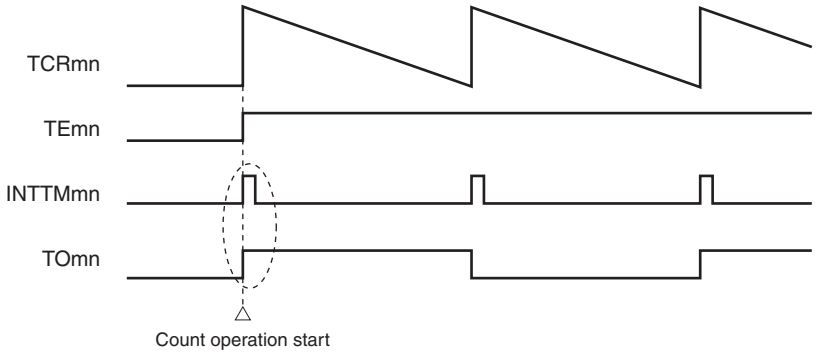
When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

In the other modes, neither timer interrupt at count operation start nor TOmn output is controlled.

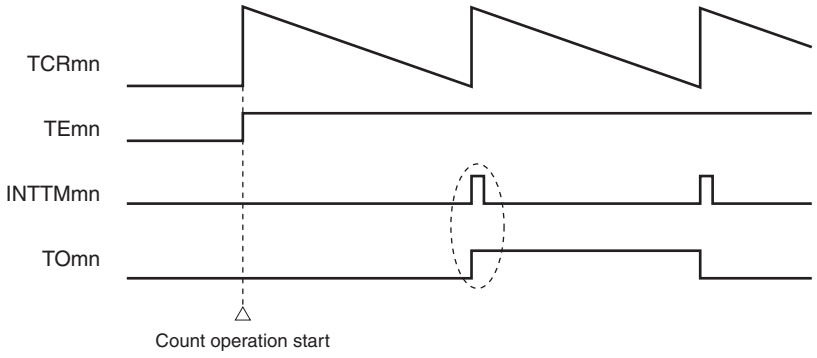
Figures 6-36 show operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

**Figure 6-42. Operation examples of timer interrupt at count operation start and TOmn output**

**(a) When MDmn0 is set to 1**




**(b) When MDmn0 is set to 0**




When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOmn does not change either. After counting one cycle, INTTMmn is output and TOmn performs a toggle operation.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

## 6.7 Independent Channel Operation Function of Timer Array Unit

### 6.7.1 Operation as interval timer/square wave output

**(1) Interval timer**

The timer array unit can be used as a reference timer that generates INTTMmn (timer interrupt) at fixed intervals. The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

**(2) Operation as square wave output**

TOmn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOmn can be calculated by the following expressions.

- Period of square wave output from TOmn = Period of count clock × (Set value of TDRmn + 1) × 2
- Frequency of square wave output from TOmn = Frequency of count clock/{(Set value of TDRmn + 1) × 2}

Timer count register mn (TCRmn) operates as a down counter in the interval timer mode.

The TCRmn register loads the value of timer data register mn (TDRmn) at the first count clock after the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1. If the MDmn0 bit of timer mode register mn (TMRmn) is 0 at this time, INTTMmn is not output and TOmn is not toggled. If the MDmn0 bit of the TMRmn register is 1, INTTMmn is output and TOmn is toggled.

After that, the TCRmn register count down in synchronization with the count clock.

When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, the TCRmn register loads the value of the TDRmn register again. After that, the same operation is repeated.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-43. Block Diagram of Operation as Interval Timer/Square Wave Output**

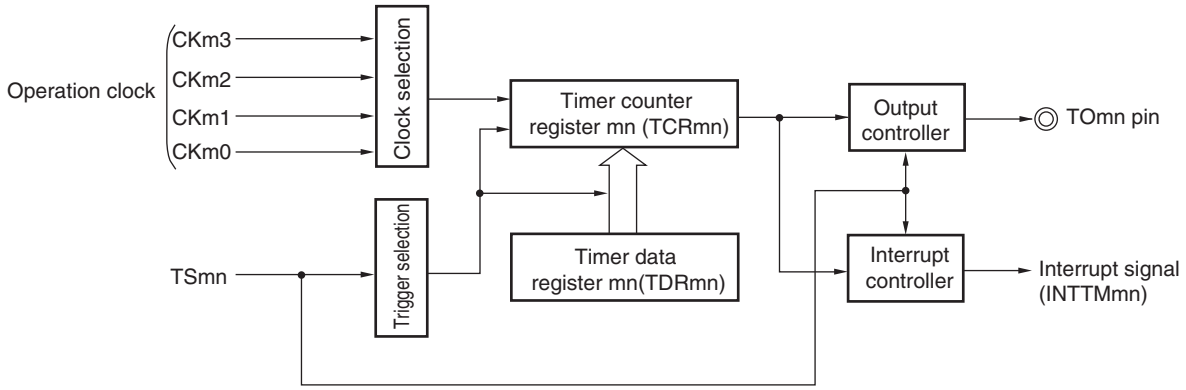
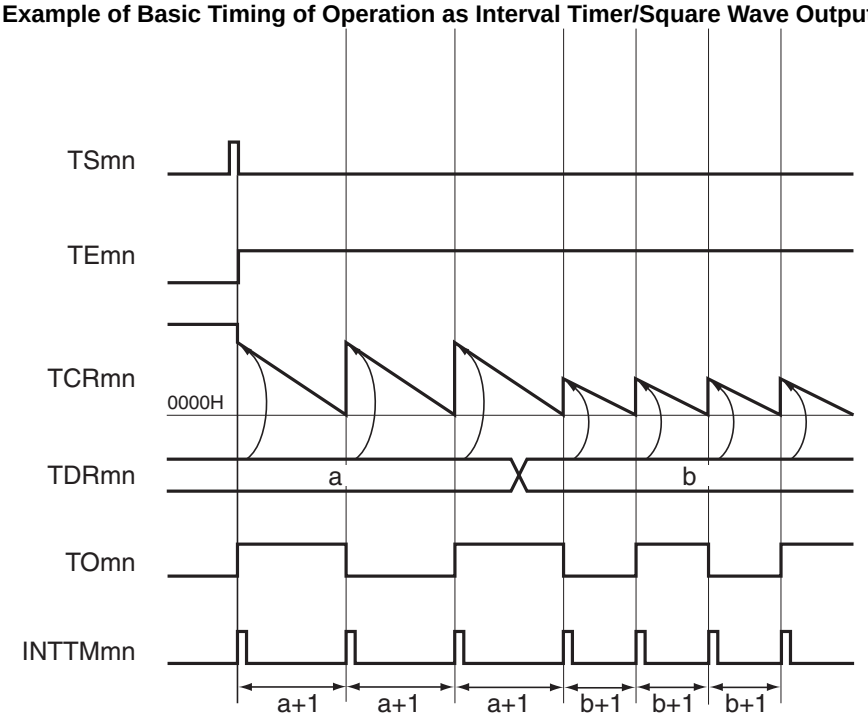



**Figure 6-44. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
TOmn: TOmn pin output signal

**3.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-45. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)**

**(a) Timer mode register mn (TMRmn)**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn | CKSmn1 | CKSmn0 | | CCSmn | M/S Note | STSmn2 | STSmn1 | STSmn0 | CISmn1 | CISmn0 | | | MDmn3 | MDmn2 | MDmn1 | MDmn0 |
| | 1/0 | 1/0 | 0 | 0 | 0/1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1/0 |

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.
1: Generates INTTMmn and inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channels 2, 4, 6)
0: Independent channel operation function.

Setting of SPLITmn bit (channels 1, 3)
1: 8-bit timer mode

Count clock selection
0: Selects operation clock ($f_{MCK}$).

Operation clock ($f_{MCK}$) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channel n.
11B: Selects CKm3 as operation clock of channel n.

**(b) Timer output register m (TOm)**

| | Bit n |
|---|---|
| TOm | TOmn |
| | 1/0 |

0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

**(c) Timer output enable register m (TOEm)**

| | Bit n |
|---|---|
| TOEm | TOEmn |
| | 1/0 |

0: Stops the TOmn output operation by counting operation.
1: Enables the TOmn output operation by counting operation.

**Note** TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-45. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/2)**

**(d) Timer output level register m (TOLm)**

| | Bit n | |
|---|---|---|
| TOLm | TOLmn<br>0 | 0: Cleared to 0 when TOMmn = 0 (master channel output mode) |

**(e) Timer output mode register m (TOMm)**

| | Bit n | |
|---|---|---|
| TOMm | TOMmn<br>0 | 0: Sets master channel output mode. |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-46. Operation Procedure of Interval Timer/Square Wave Output Function (1/2)**

| | Software Operation | Hardware Status |
|---|---|---|
| TAU default setting | | Power-off status<br>(Clock supply is stopped and writing to each register is disabled.) |
| | Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. → | Power-on status. Each channel stops operating.<br>(Clock supply is started and writing to each register is enabled.) |
| | Sets timer clock select register m (TPSm).<br>Determines clock frequencies of CKm0 to CKm3. | |
| Channel default setting | Sets timer mode register mn (TMRmn) (determines operation mode of channel).<br>Sets interval (period) value to timer data register mn (TDRmn). | Channel stops operating.<br>(Clock is supplied and some power is consumed.) |
| | To use the TOmn output<br>Clears the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode).<br>Clears the TOLmn bit to 0.<br>Sets the TOmn bit and determines default level of the TOmn output. → | The TOmn pin goes into Hi-Z output state.<br>The TOmn default setting level is output when the port mode register is in the output mode and the port register is 0. |
| | Sets the TOEmn bit to 1 and enables operation of TOmn. → | TOmn does not change because channel stops operating. |
| | Clears the port register and port mode register to 0. → | The TOmn pin outputs the TOmn set level. |
| Operation start | (Sets the TOEmn bit to 1 only if using TOmn output and resuming operation.).<br>Sets the TSmn (TSHm1, TSHm3) bit to 1. →<br>The TSmn (TSHm1, TSHm3) bit automatically returns to 0 because it is a trigger bit. | TEmn (TEHm1, TEHm3) = 1, and count operation starts.<br>Value of the TDRmn register is loaded to timer count register mn (TCRmn) at the count clock input. INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1. |
| During operation | Set values of the TMRmn register, TOMmn, and TOLmn bits cannot be changed.<br>Set value of the TDRmn register can be changed.<br>The TCRmn register can always be read.<br>The TSRmn register is not used.<br>Set values of the TOm and TOEm registers can be changed. | Counter (TCRmn) counts down. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again and the count operation is continued.<br>By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation.<br>After that, the above operation is repeated. |
| Operation stop | The TTmn (TTHm1, TTHm3) bit is set to 1. →<br>The TTmn (TTHm1, TTHm3) bit automatically returns to 0 because it is a trigger bit. | TEmn (TEHm1, TEHm3), and count operation stops.<br>The TCRmn register holds count value and stops.<br>The TOmn output is not initialized but holds current status. |
| | The TOEmn bit is cleared to 0 and value is set to the TOmn bit. → | The TOmn pin outputs the TOmn bit set level. |

Operation is resumed.

(**Remarks** are listed on the next page.)

**Figure 6-46. Operation Procedure of Interval Timer/Square Wave Output Function (2/2)**

| | Software Operation | Hardware Status |
|---|---|---|
| TAU stop | To hold the TOmn pin output level<br>Clears the TOmn bit to 0 after the value to be held is set to the port register. → | The TOmn pin output level is held by port function. |
| | When holding the TOmn pin output level is not necessary<br>Setting not required. | |
| | The TAUmEN bit of the PER0 register is cleared to 0. → | Power-off status<br>All circuits are initialized and SFR of each channel is also initialized.<br>(The TOmn bit is cleared to 0 and the TOmn pin is set to port mode.) |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

### 6.7.2 Operation as external event counter

The timer array unit can be used as an external event counter that counts the number of times the valid input edge (external event) is detected in the TImn pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

| Specified number of counts = Set value of TDRmn + 1 |
|---|

Timer count register mn (TCRmn) operates as a down counter in the event counter mode.

The TCRmn register loads the value of timer data register mn (TDRmn) by setting any channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) to 1.

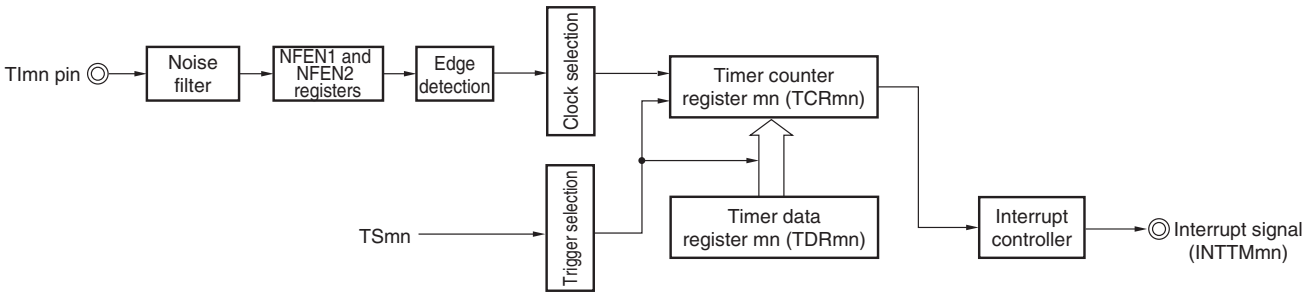
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TOmn pin. Stop the output by setting the TOEmn bit of timer output enable register m (TOEm) to 0.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

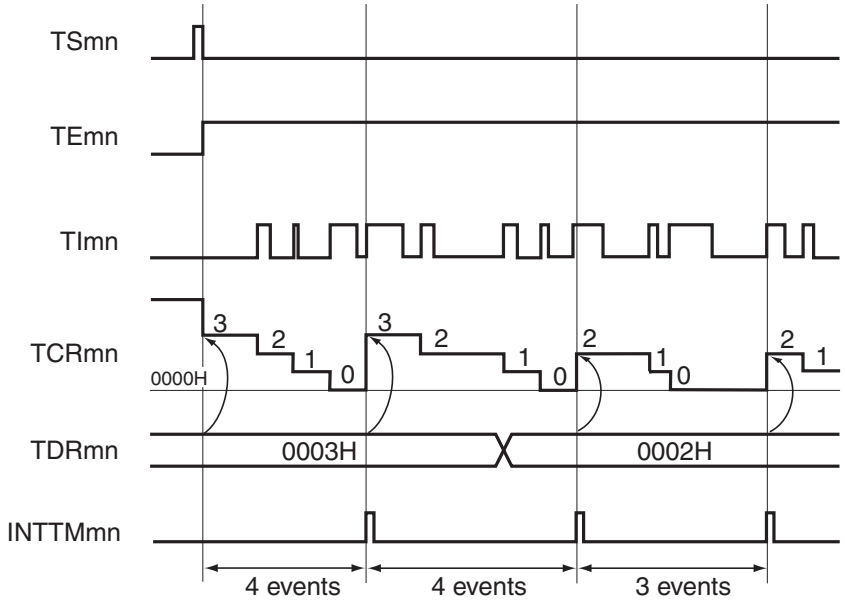
**Figure 6-47. Block Diagram of Operation as External Event Counter**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-48. Example of Basic Timing of Operation as External Event Counter**



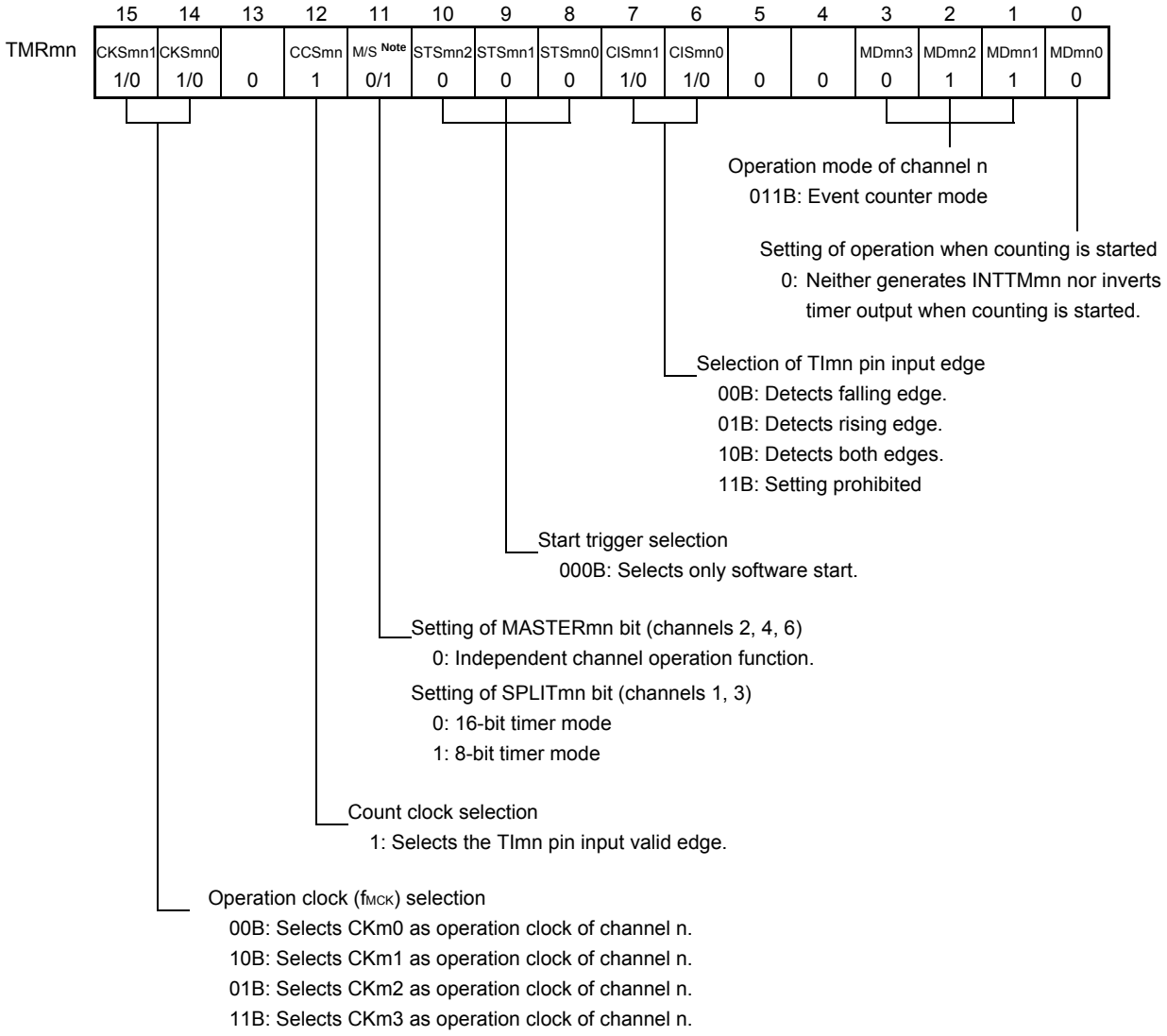

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

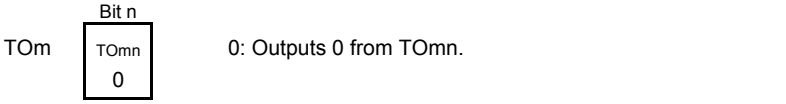
**3.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-49. Example of Set Contents of Registers in External Event Counter Mode (1/2)**

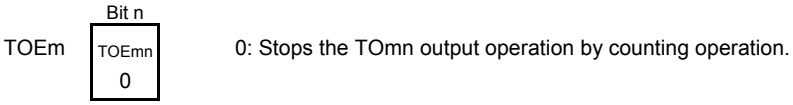
**(a) Timer mode register mn (TMRmn)**




**(b) Timer output register m (TOm)**




**(c) Timer output enable register m (TOEm)**




**Note** TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-49. Example of Set Contents of Registers in External Event Counter Mode (2/2)**

**(d) Timer output level register m (TOLm)**

| | Bit n | |
|---|---|---|
| TOLm | TOLmn | 0: Cleared to 0 when TOMmn = 0 (master channel output mode). |
| | 0 | |

**(e) Timer output mode register m (TOMm)**

| | Bit n | |
|---|---|---|
| TOMm | TOMmn | 0: Sets master channel output mode. |
| | 0 | |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-50. Operation Procedure When External Event Counter Function Is Used**

| | Software Operation | Hardware Status |
|---|---|---|
| TAU default setting | | Power-off status<br>(Clock supply is stopped and writing to each register is disabled.) |
| | Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. → | Power-on status. Each channel stops operating.<br>(Clock supply is started and writing to each register is enabled.) |
| | Sets timer clock select register m (TPSm).<br>Determines clock frequencies of CKm0 to CKm3. | |
| Channel default setting | Sets timer mode register mn (TMRmn) (determines operation mode of channel).<br>Sets number of counts to timer data register mn (TDRmn).<br>Clears the TOEmn bit of timer output enable register m (TOEm) to 0. | Channel stops operating.<br>(Clock is supplied and some power is consumed.) |
| Operation start | Sets the TSmn bit to 1. →<br>The TSmn bit automatically returns to 0 because it is a trigger bit. | TEmn = 1, and count operation starts.<br>Value of the TDRmn register is loaded to timer count register mn (TCRmn) and detection of the TImn pin input edge is awaited. |
| During operation | Set value of the TDRmn register can be changed.<br>Sets corresponding bit of noise filter enable registers 1 and 2 (NFEN1 and NFEN2) to 1.<br>The TCRmn register can always be read.<br>The TSRmn register is not used.<br>Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed. | Counter (TCRmn) counts down each time input edge of the TImn pin has been detected. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again, and the count operation is continued. By detecting TCRmn = 0000H, the INTTMmn output is generated.<br>After that, the above operation is repeated. |
| Operation stop | The TTmn bit is set to 1. →<br>The TTmn bit automatically returns to 0 because it is a trigger bit. | TEmn = 0, and count operation stops.<br>The TCRmn register holds count value and stops. |
| TAU stop | The TAUmEN bit of the PER0 register is cleared to 0. → | Power-off status<br>All circuits are initialized and SFR of each channel is also initialized. |

Operation is resumed.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

### 6.7.3 Operation as frequency divider

The timer array unit can be used as a frequency divider that divides a clock input to the TImn pin and outputs the result from the TOmn pin.

Set the TImn and TOmn pins so that they are different from each other by the peripheral I/O redirection registers 0, 1, 2, and 3 (PIOR0, PIOR1, PIOPR2, and PIOR3).

The divided clock frequency output from TOmn can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDR00 + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDR00 + 1)

Timer count register m (TCRm) operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS00) of timer channel start register 0 (TS0) is set to 1, the TCR00 register loads the value of timer data register 00 (TDR00) when the TI00 valid edge is detected.

If the MNmn0 bit of timer mode register 00 (TMR00) is 0 at this time, INTTM00 is not output and TO00 is not toggled. If the MNmn0 bit of timer mode register 00 (TMR00) is 1, INTTM00 is output and TO00 is toggled.

After that, the TCR00 register counts down at the valid edge of the TI00 pin. When TCR00 = 0000H, it toggles TO00. At the same time, the TCR00 register loads the value of the TDR00 register again, and continues counting.

If detection of both the edges of the TI00 pin is selected, the duty factor error of the input clock affects the divided clock period of the TO00 output.

The period of the TO00 output clock includes a sampling error of one period of the operation clock.

Clock period of TO00 output = Ideal TO00 output clock period ± Operation clock period (error)

The TDR00 register can be rewritten at any time. The new value of the TDR00 register becomes valid during the next count period.

**Figure 6-51. Block Diagram of Operation as Frequency Divider**

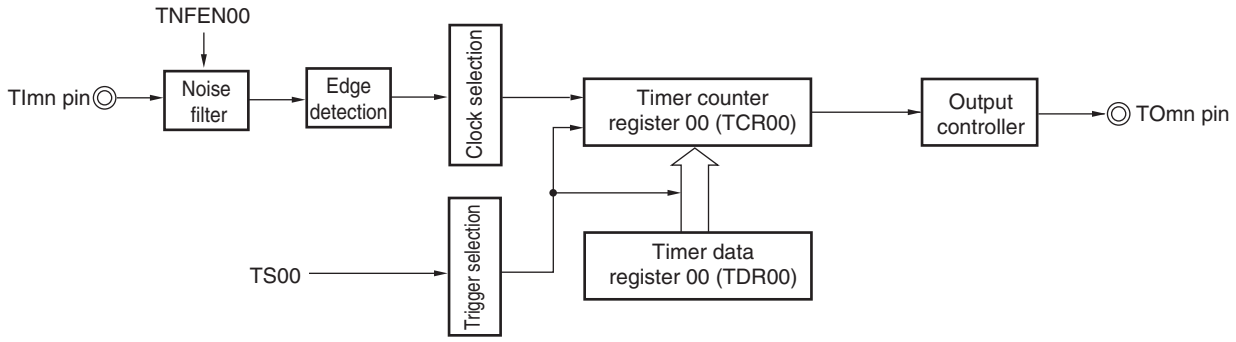
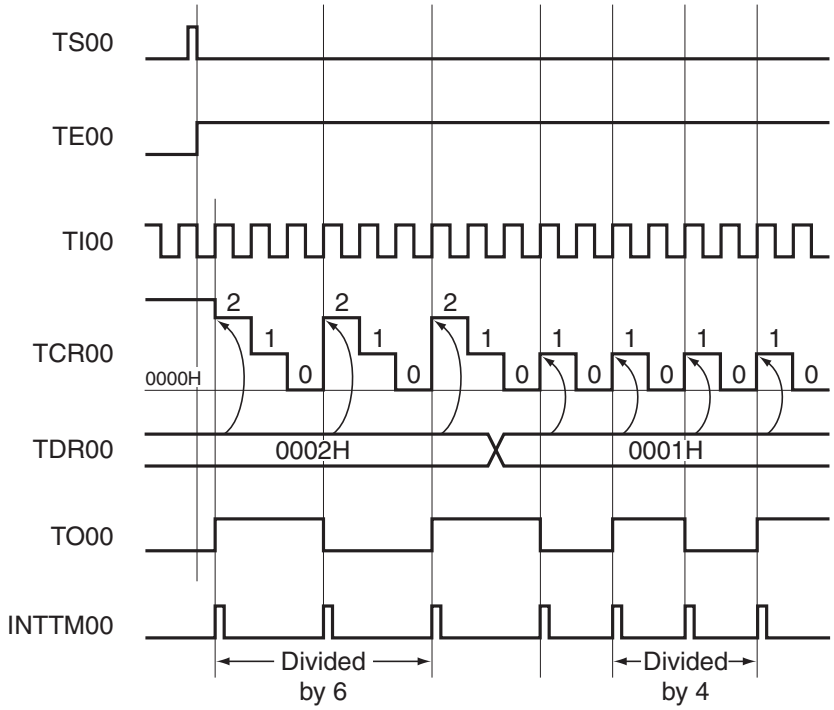


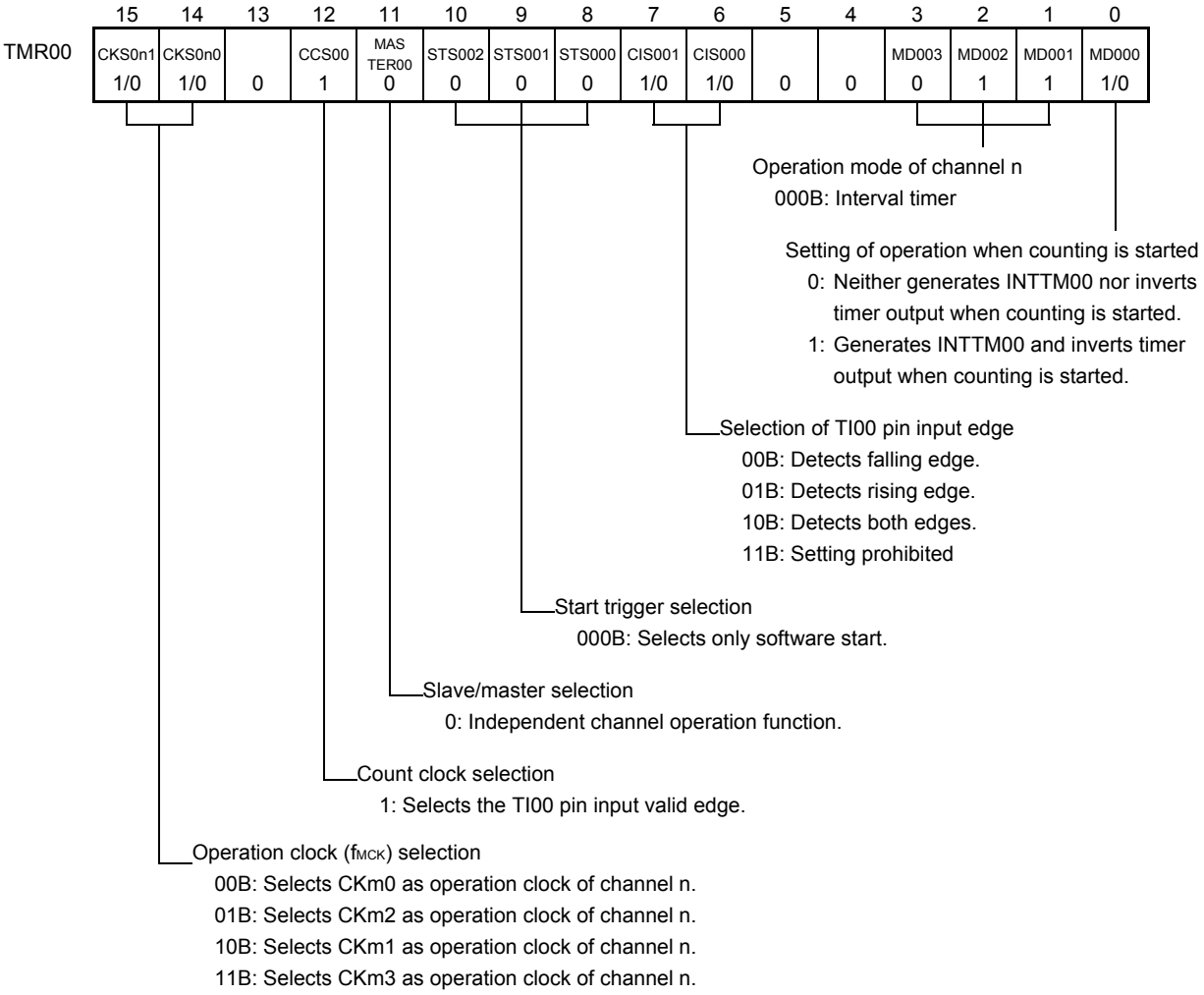
**Figure 6-52. Example of Basic Timing of Operation as Frequency Divider (MDmn0 = 1)**




**Remark** TS00: Bit n of timer channel start register m (TSm)
TE00: Bit n of timer channel enable status register m (TEm)
TI00: TI00 pin input signal
TCR00: Timer count register 00 (TCR00)
TDR00: Timer data register 00 (TDR00)
TO00: TO00 pin output signal

**Figure 6-53. Example of Set Contents of Registers During Operation as Frequency Divider**

**(a) Timer mode register 00 (TMR00)**




**(b) Timer output register m (TOm)**

| | Bit n | |
|---|---|---|
| TOm | TO00<br>1/0 | 0: Outputs 0 from TO00.<br>1: Outputs 1 from TO00. |

**(c) Timer output enable register m (TOEm)**

| | Bit n | |
|---|---|---|
| TOEm | TOE00<br>1/0 | 0: Stops the TO00 output operation by counting operation.<br>1: Enables the TO00 output operation by counting operation. |

**(d) Timer output level register m (TOLm)**

| | Bit n | |
|---|---|---|
| TOLm | TOL00<br>0 | 0: Cleared to 0 when TOM00 = 0 (master channel output mode) |

**(e) Timer output mode register m (TOMm)**

| | Bit n | |
|---|---|---|
| TOMm | TOM00<br>0 | 0: Sets master channel output mode. |

**Figure 6-54. Operation Procedure When Frequency Divider Function Is Used**

| | Software Operation | Hardware Status |
|---|---|---|
| TAU default setting | | Power-off status<br>(Clock supply is stopped and writing to each register is disabled.) |
| | Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. → | Power-on status. Each channel stops operating.<br>(Clock supply is started and writing to each register is enabled.) |
| | Sets timer clock select register m (TPSm).<br>Determines clock frequencies of CKm0 to CKm3. | |
| Channel default setting | Sets timer mode register 0n (TMR0n) (determines operation mode of channel and selects the detection edge).<br>Sets interval (period) value to timer data register 00 (TDR00). | Channel stops operating.<br>(Clock is supplied and some power is consumed.) |
| | Clears the TOM00 bit of timer output mode register m (TOMm) to 0 (master channel output mode).<br>Clears the TOL00 bit to 0.<br>Sets the TO00 bit and determines default level of the TO00 output. → | The TO00 pin goes into Hi-Z output state.<br>The TO00 default setting level is output when the port mode register is in output mode and the port register is 0. |
| | Sets the TOE00 bit to 1 and enables operation of TO00. → | TO00 does not change because channel stops operating. |
| | Clears the port register and port mode register to 0. → | The TO00 pin outputs the TO00 set level. |
| Operation start | Sets the TOE00 bit to 1 (only when operation is resumed). →<br>Sets the TS00 bit to 1.<br>The TS00 bit automatically returns to 0 because it is a trigger bit. | TE00 = 1, and count operation starts.<br>Value of the TDR00 register is loaded to timer count register 00 (TCR00) at the count clock input. INTTM00 is generated and TO00 performs toggle operation if the MD000 bit of the TMR00 register is 1. |
| During operation | Set value of the TDR00 register can be changed.<br>Sets corresponding bit of noise filter enable registers 1 and 2 (NFEN1 and NFEN2) to 1.<br>The TCR00 register can always be read.<br>The TSR00 register is not used.<br>Set values of the TOm and TOEm registers can be changed.<br>Set values of the TMR00 register, TOM00, and TOL00 bits cannot be changed. | Counter (TCR00) counts down. When count value reaches 0000H, the value of the TDR00 register is loaded to the TCR00 register again, and the count operation is continued.<br>By detecting TCR00 = 0000H, INTTM00 is generated and TO00 performs toggle operation.<br>After that, the above operation is repeated. |
| Operation stop | The TT00 bit is set to 1. →<br>The TT00 bit automatically returns to 0 because it is a trigger bit. | TE00 = 0, and count operation stops.<br>The TCR00 register holds count value and stops.<br>The TO00 output is not initialized but holds current status. |
| | The TOE00 bit is cleared to 0 and value is set to the TO00 bit. → | The TO00 pin outputs the TO00 set level. |
| TAU stop | To hold the TO00 pin output level<br>Clears the TO00 bit to 0 after the value to be held is set to the port register. →<br>When holding the TO00 pin output level is not necessary<br>Setting not required. | The TO00 pin output level is held by port function. |
| | The TAUmEN bit of the PER0 register is cleared to 0. → | Power-off status<br>All circuits are initialized and SFR of each channel is also initialized.<br>(The TO00 bit is cleared to 0 and the TO00 pin is set to port mode). |

Operation is resumed.

### 6.7.4 Operation as input pulse interval measurement

The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured.

The pulse interval can be calculated by the following expression.

| TImn input pulse interval = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1)) |
|---|

**Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error of up to one operating clock cycle occurs.**

Timer count register mn (TCRmn) operates as an up counter in the capture mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TCRmn register counts up from 0000H in synchronization with the count clock.

When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

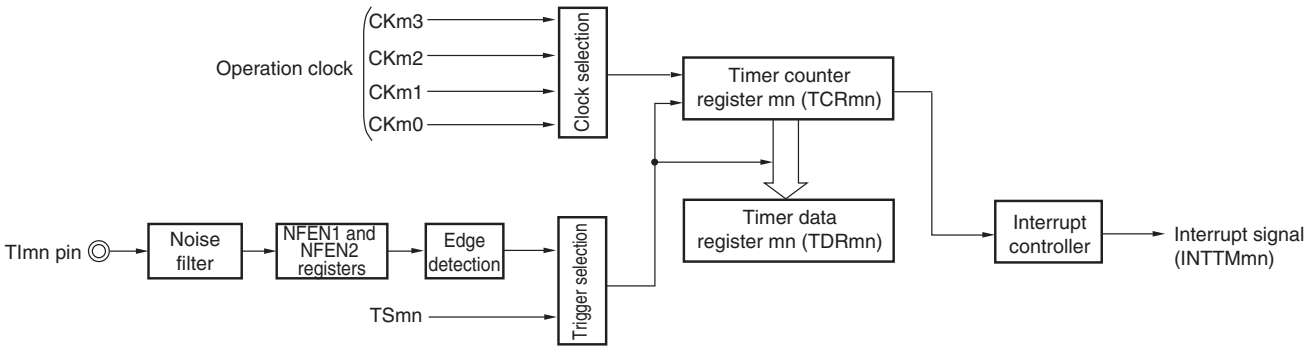
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Set the STSmn2 to STSmn0 bits of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

When TEmn = 1, a software operation (TSmn = 1) can be used as a capture trigger, instead of using the TImn pin input.

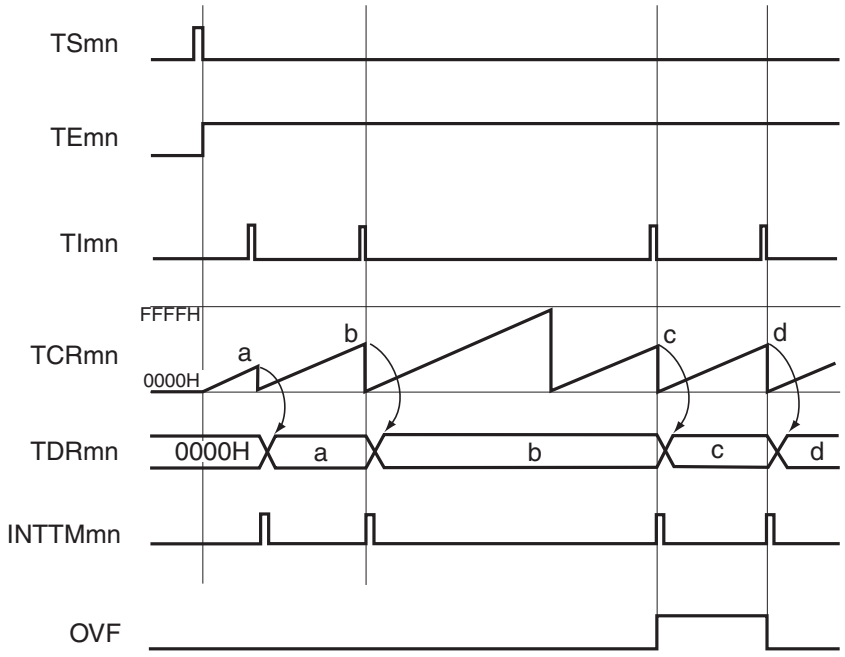
**Figure 6-55. Block Diagram of Operation as Input Pulse Interval Measurement**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-56. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)**



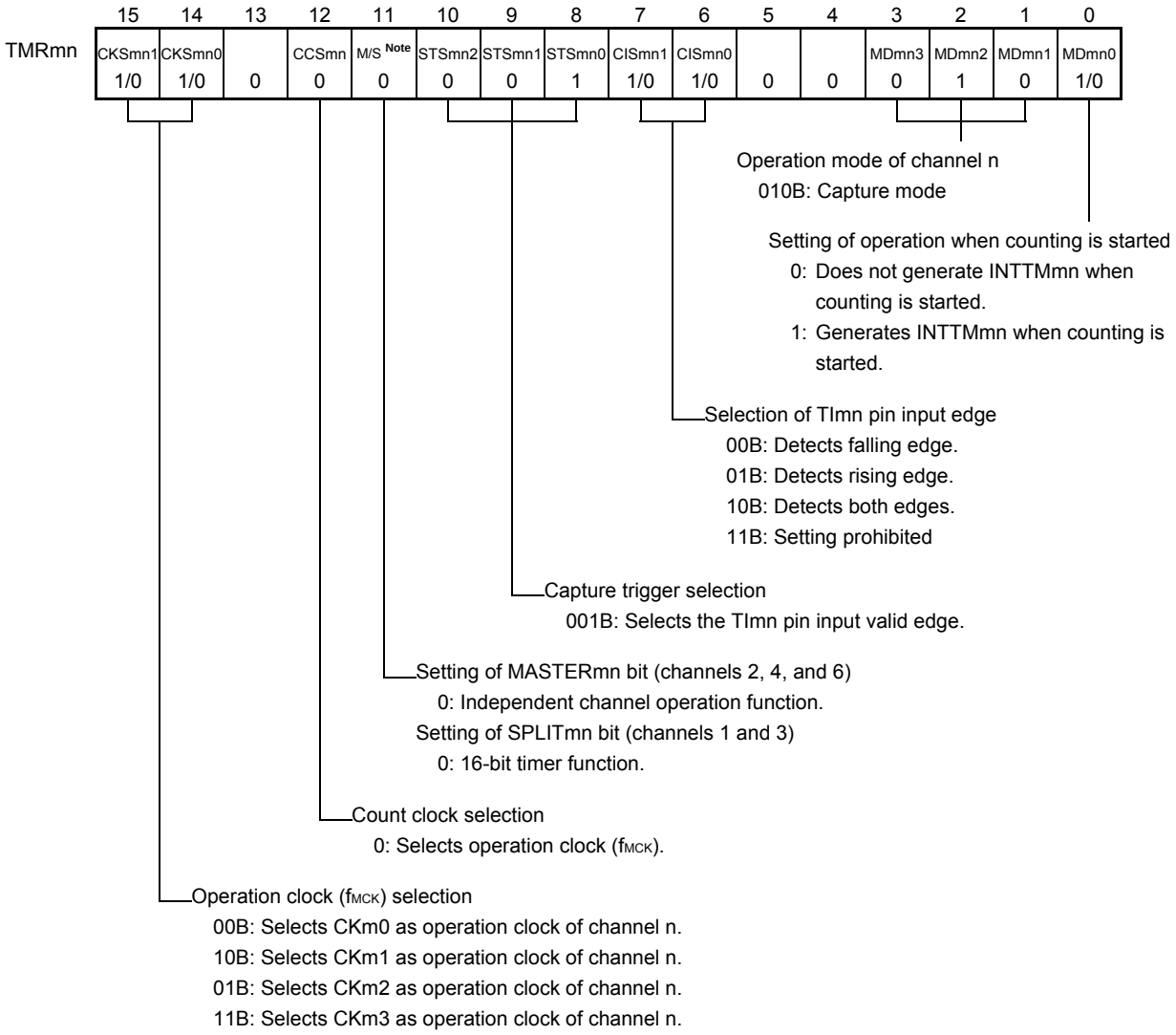

**Remarks 1.** m: Unit number (m = 0, 1)n: Channel number (n = 0 to 7)

**2.** TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

**3.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-57. Example of Set Contents of Registers to Measure Input Pulse Interval**

**(a) Timer mode register mn (TMRmn)**




**(b) Timer output register m (TOm)**

| | Bit n | |
|---|---|---|
| TOm | TOmn<br>0 | 0: Outputs 0 from TOmn. |

**(c) Timer output enable register m (TOEm)**

| | Bit n | |
|---|---|---|
| TOEm | TOEmn<br>0 | 0: Stops TOmn output operation by counting operation. |

**(d) Timer output level register m (TOLm)**

| | Bit n | |
|---|---|---|
| TOLm | TOLmn<br>0 | 0: Cleared to 0 when TOMmn = 0 (master channel output mode). |

**(e) Timer output mode register m (TOMm)**

| | Bit n | |
|---|---|---|
| TOMm | TOMmn<br>0 | 0: Sets master channel output mode. |

**Note** TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit

TMRm0, TMRm5, TMRm7: Fixed to 0

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-58. Operation Procedure When Input Pulse Interval Measurement Function Is Used**

| | Software Operation | Hardware Status |
|---|---|---|
| TAU default setting | | Power-off status<br>(Clock supply is stopped and writing to each register is disabled.) |
| | Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. → | Power-on status. Each channel stops operating.<br>(Clock supply is started and writing to each register is enabled.) |
| | Sets timer clock select register m (TPSm).<br>Determines clock frequencies of CKm0 to CKm3. | |
| Channel default setting | Sets timer mode register mn (TMRmn) (determines operation mode of channel). | Channel stops operating.<br>(Clock is supplied and some power is consumed.) |
| Operation start | Sets TSmn bit to 1. →<br>The TSmn bit automatically returns to 0 because it is a trigger bit. | TEmn = 1, and count operation starts.<br>Timer count register mn (TCRmn) is cleared to 0000H at the count clock input.<br>When the MDmn0 bit of the TMRmn register is 1, INTTMmn is generated. |
| During operation | Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed.<br>Sets corresponding bit of noise filter enable registers 1 and 2 (NFEN1 and NFEN2) to 1.<br>The TDRmn register can always be read.<br>The TCRmn register can always be read.<br>The TSRmn register can always be read.<br>Set values of the TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed. | Counter (TCRmn) counts up from 0000H. When the TImn pin input valid edge is detected, the count value is transferred (captured) to timer data register mn (TDRmn). At the same time, the TCRmn register is cleared to 0000H, and the INTTMmn signal is generated.<br>If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared.<br>After that, the above operation is repeated. |
| Operation stop | The TTmn bit is set to 1. →<br>The TTmn bit automatically returns to 0 because it is a trigger bit. | TEmn = 0, and count operation stops.<br>The TCRmn register holds count value and stops.<br>The OVF bit of the TSRmn register is also held. |
| TAU stop | The TAUmEN bit of the PER0 register is cleared to 0. → | Power-off status<br>All circuits are initialized and SFR of each channel is also initialized. |

Operation is resumed.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

### 6.7.5 Operation as input signal high-/low-level width measurement

**Caution When using a channel to implement the LIN-bus, set bit 7 (TIS17) to 1 and bit 6 (TIS16) to 0 of the timer input select register 1 (TIS1). In the following descriptions, read TImn as RxD0.**

By starting counting at one edge of the TImn pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TImn can be measured. The signal width of TImn can be calculated by the following expression.

| Signal width of TImn input = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1)) |
|---|

**Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error equivalent to one operation clock occurs.**

Timer count register mn (TCRmn) operates as an up counter in the capture & one-count mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TEmn bit is set to 1 and the TImn pin start edge detection wait status is set.

When the TImn pin input start edge (rising edge of the TImn pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TImn pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register mn (TDRmn) and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops at the value “value transferred to the TDRmn register + 1”, and the TImn pin start edge detection wait status is set. After that, the above operation is repeated.

As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

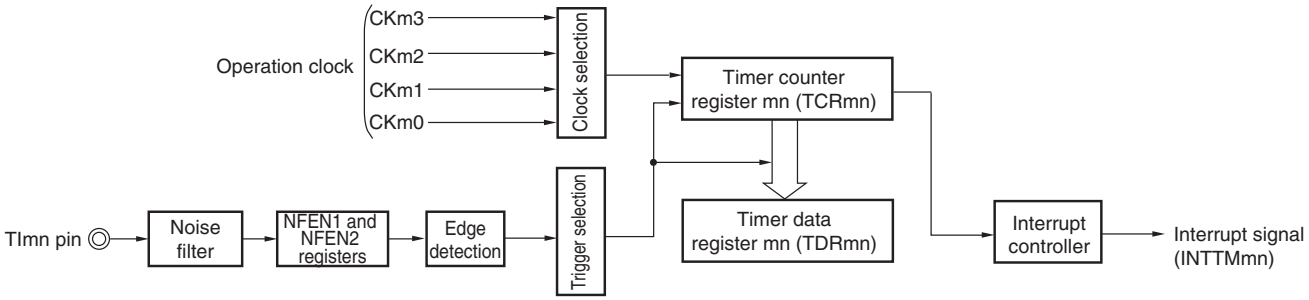
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.

Because this function is used to measure the signal width of the TImn pin input, the TSmn bit cannot be set to 1 while the TEmn bit is 1.

CISmn1, CISmn0 of TMRmn register = 10B: Low-level width is measured.

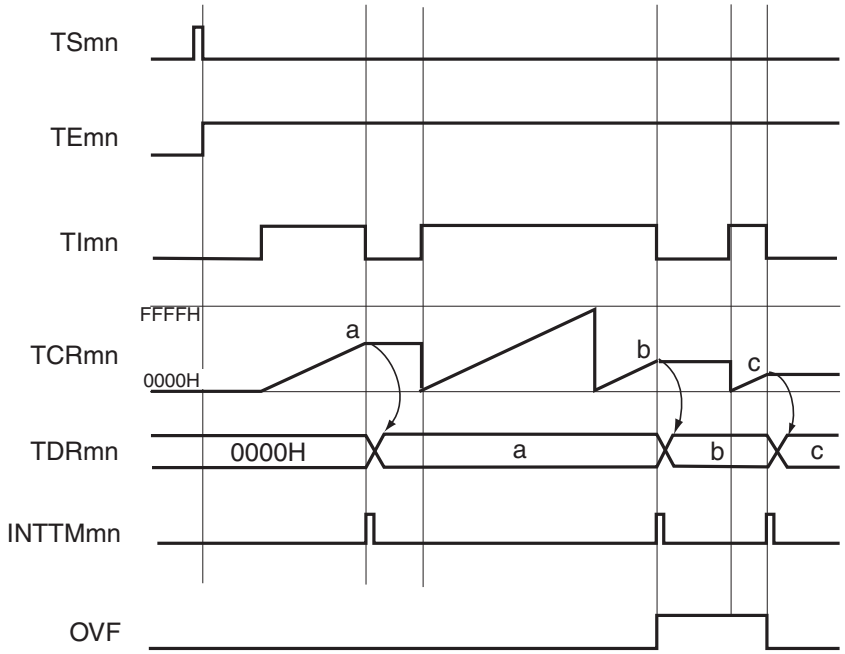
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

**Figure 6-59. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**Figure 6-60. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement**



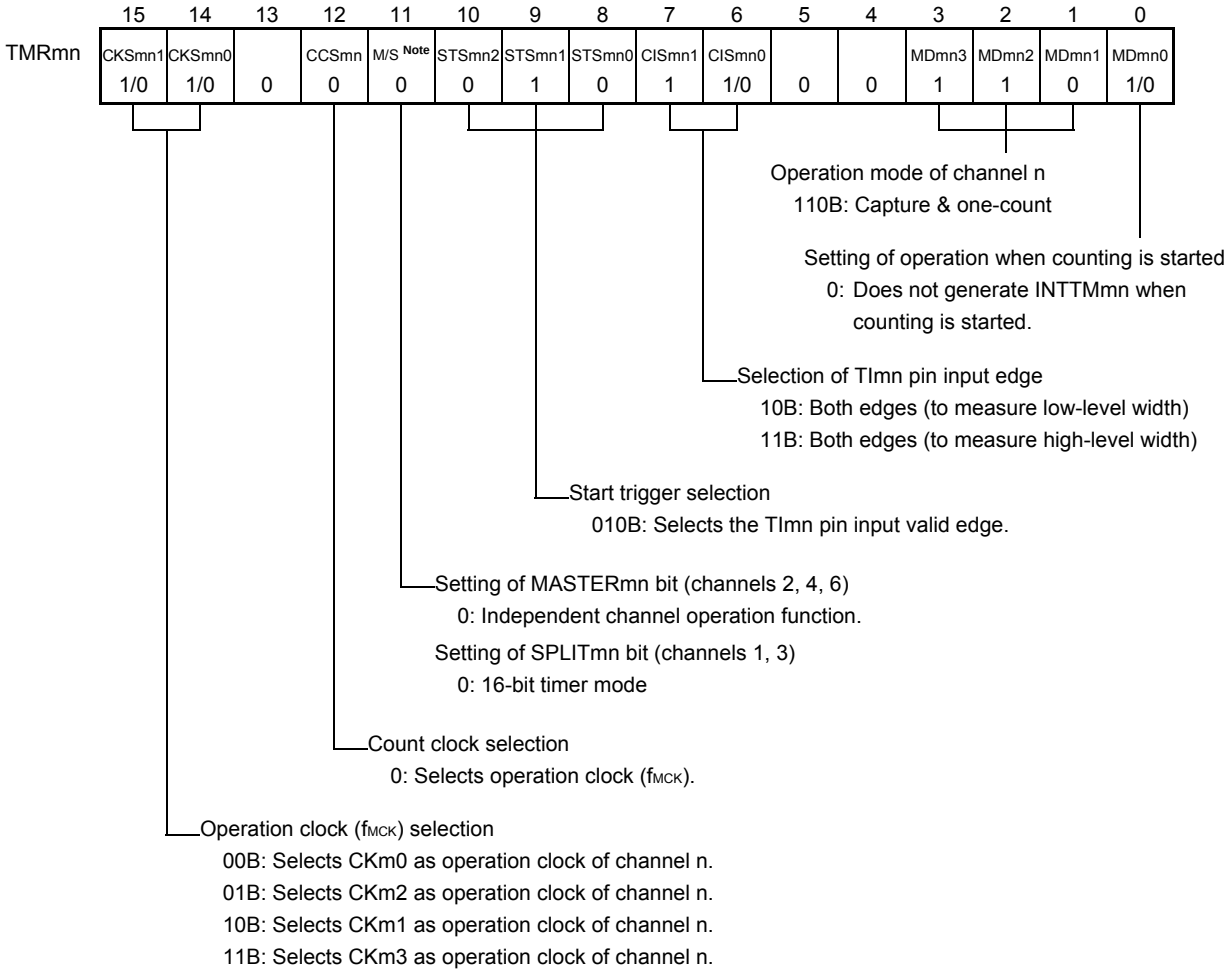

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

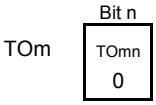
**3.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-61. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width**

**(a) Timer mode register mn (TMRmn)**

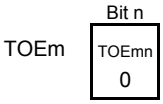



**(b) Timer output register m (TOm)**



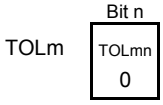

0: Outputs 0 from TOmn.

**(c) Timer output enable register m (TOEm)**



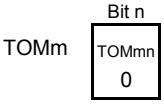

0: Stops the TOmn output operation by counting operation.

**(d) Timer output level register m (TOLm)**




0: Cleared to 0 when TOMmn = 0 (master channel output mode).

**(e) Timer output mode register m (TOMm)**




0: Sets master channel output mode.

**Note** TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.

Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-62. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used**

| | Software Operation | Hardware Status |
|---|---|---|
| TAU default setting | | Power-off status<br>(Clock supply is stopped and writing to each register is disabled.) |
| | Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. → | Power-on status. Each channel stops operating.<br>(Clock supply is started and writing to each register is enabled.) |
| | Sets timer clock select register m (TPSm).<br>Determines clock frequencies of CKm0 to CKm3. | |
| Channel default setting | Sets timer mode register mn (TMRmn) (determines operation mode of channel).<br>Clears the TOEmn bit to 0 and stops operation of TOmn. | Channel stops operating.<br>(Clock is supplied and some power is consumed.) |
| Operation start | Sets the TSmn bit to 1. →<br>The TSmn bit automatically returns to 0 because it is a trigger bit. | TEmn = 1, and the TImn pin start edge detection wait status is set. |
| | Detects the TImn pin input count start valid edge. → | Clears timer count register mn (TCRmn) to 0000H and starts counting up. |
| During operation | Set value of the TDRmn register can be changed.<br>Sets corresponding bit of noise filter enable registers 1 and 2 (NFEN1 and NFEN2) to 1.<br>The TCRmn register can always be read.<br>The TSRmn register is not used.<br>Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed. | When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to timer data register mn (TDRmn) and INTTMmn is generated.<br>If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. The TCRmn register stops the count operation until the next TImn pin start edge is detected. |
| Operation stop | The TTmn bit is set to 1. →<br>The TTmn bit automatically returns to 0 because it is a trigger bit. | TEmn = 0, and count operation stops.<br>The TCRmn register holds count value and stops.<br>The OVF bit of the TSRmn register is also held. |
| TAU stop | The TAUmEN bit of the PER0 register is cleared to 0. → | Power-off status<br>All circuits are initialized and SFR of each channel is also initialized. |

Operation is resumed.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

### 6.7.6 Operation as delay counter

It is possible to start counting down when the valid edge of the TImn pin input is detected (an external event), and then generate INTTMmn (a timer interrupt) after any specified interval.

It can also generate INTTMmn (timer interrupt) at any interval by making a software set TSmn = 1 and the count down start during the period of TEmn = 1.

The interrupt generation period can be calculated by the following expression.

| Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1) |
|---|

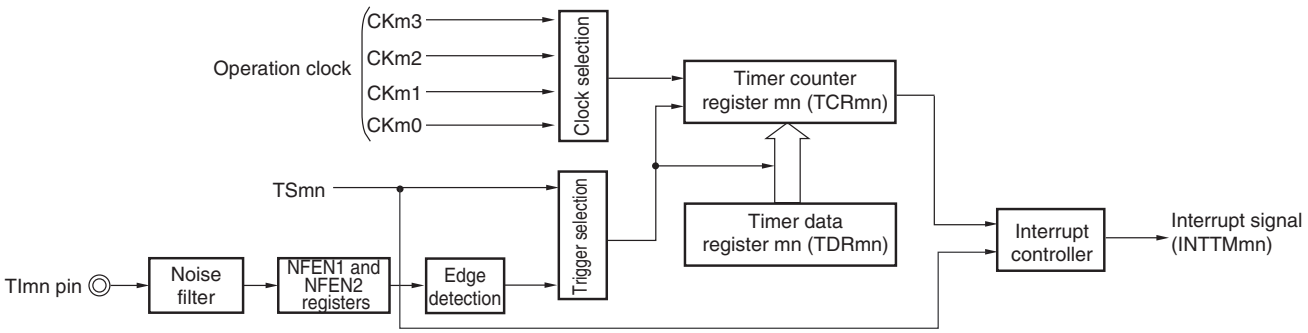
Timer count register mn (TCRmn) operates as a down counter in the one-count mode.

When the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1, the TEmn, TEHm1, TEHm3 bits are set to 1 and the TImn pin input valid edge detection wait status is set.

Timer count register mn (TCRmn) starts operating upon TImn pin input valid edge detection and loads the value of timer data register mn (TDRmn). The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next TImn pin input valid edge is detected.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

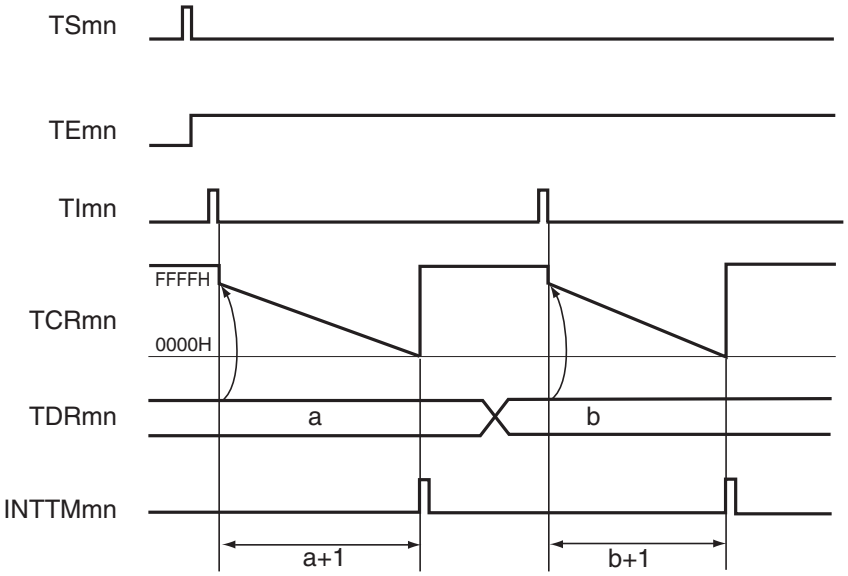
**Figure 6-63. Block Diagram of Operation as Delay Counter**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-64. Example of Basic Timing of Operation as Delay Counter**



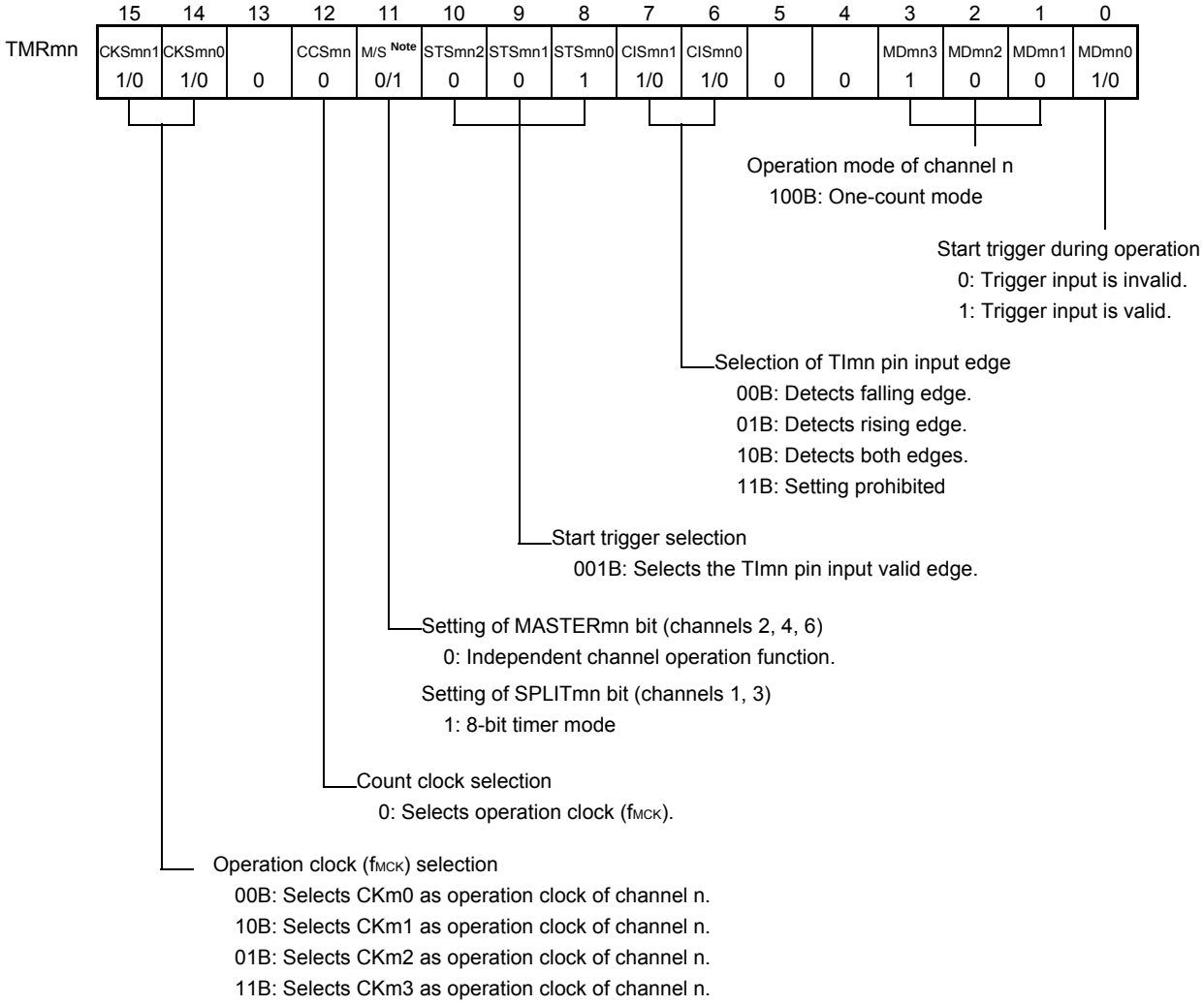

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

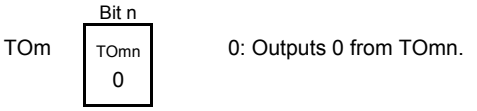
**3.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-65. Example of Set Contents of Registers to Delay Counter (1/2)**

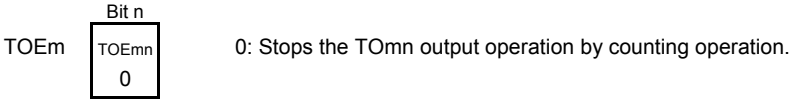
**(a) Timer mode register mn (TMRmn)**




**(b) Timer output register m (TOm)**




**(c) Timer output enable register m (TOEm)**




**Note** TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-65. Example of Set Contents of Registers to Delay Counter (2/2)**

**(d) Timer output level register m (TOLm)**

| | Bit n | |
|---|---|---|
| TOLm | TOLmn<br>0 | 0: Cleared to 0 when TOMmn = 0 (master channel output mode). |

**(e) Timer output mode register m (TOMm)**

| | Bit n | |
|---|---|---|
| TOMm | TOMmn<br>0 | 0: Sets master channel output mode. |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-66. Operation Procedure When Delay Counter Function Is Used**

| | Software Operation | Hardware Status |
|---|---|---|
| TAU default setting | | Power-off status<br>(Clock supply is stopped and writing to each register is disabled.) |
| | Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. → | Power-on status. Each channel stops operating.<br>(Clock supply is started and writing to each register is enabled.) |
| | Sets timer clock select register m (TPSm).<br>Determines clock frequencies of CKm0 to CKm3. | |
| Channel default setting | Sets timer mode register mn (TMRmn) (determines operation mode of channel).<br>INTTMmn output delay is set to timer data register mn (TDRmn).<br>Clears the TOEmn bit to 0 and stops operation of TOmn. | Channel stops operating.<br>(Clock is supplied and some power is consumed.) |
| Operation start | Sets the TSmn bit to 1. →<br>The TSmn bit automatically returns to 0 because it is a trigger bit. | TEmn = 1, and the TImn pin input valid edge detection wait status is set. |
| | Detects the TImn pin input valid edge. → | Value of the TDRmn register is loaded to the timer count register mn (TCRmn). |
| During operation | Set value of the TDRmn register can be changed.<br>Sets corresponding bit of noise filter enable registers 1 and 2 (NFEN1 and NFEN2) to 1.<br>The TCRmn register can always be read.<br>The TSRmn register is not used. | The counter (TCRmn) counts down. When TCRmn counts down to 0000H, INTTMmn is output, and counting stops (which leaves TCRmn at 0000H) until the next TImn pin input. |
| Operation stop | The TTmn bit is set to 1. →<br>The TTmn bit automatically returns to 0 because it is a trigger bit. | TEmn = 0, and count operation stops.<br>The TCRmn register holds count value and stops. |
| TAU stop | The TAUmEN bit of the PER0 register is cleared to 0. → | Power-off status<br>All circuits are initialized and SFR of each channel is also initialized. |

Operation is resumed.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

## 6.8 Simultaneous Channel Operation Function of Timer Array Unit

### 6.8.1 Operation as one-shot pulse output function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the signal input to the TImn pin.

The delay time and pulse width can be calculated by the following expressions.

| |
|---|
| Delay time = {Set value of TDRmn (master) + 2} × Count clock period |
| Pulse width = {Set value of TDRmp (slave)} × Count clock period |

The master channel operates in the one-count mode and counts the delays. Timer count register mn (TCRmn) of the master channel starts operating upon start trigger detection and loads the value of timer data register mn (TDRmn).

The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next start trigger is detected.

The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

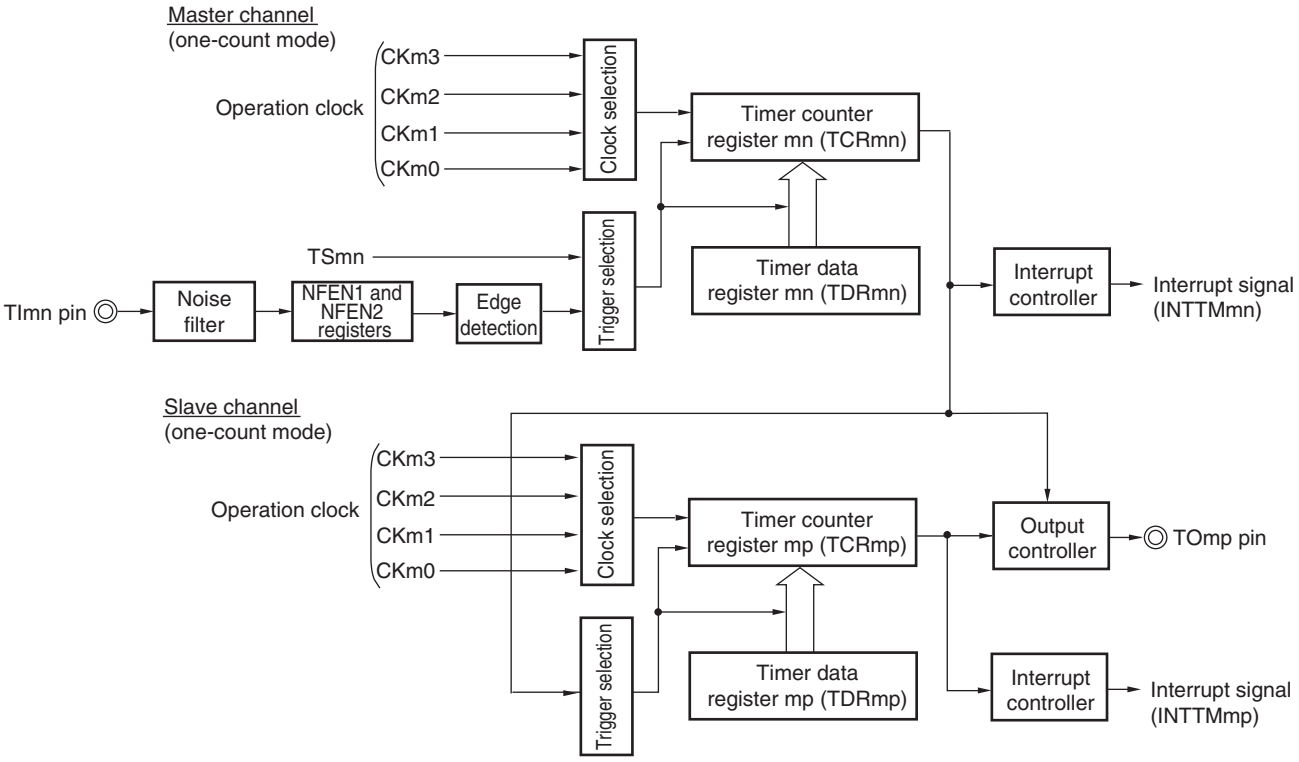
Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

**Caution The timing of loading of timer data register mn (TDRmn) of the master channel is different from that of the TDRmp register of the slave channel. If the TDRmn and TDRmp registers are rewritten during operation, therefore, an illegal waveform is output. Rewrite the TDRmn register after INTTMmn is generated and the TDRmp register after INTTMmp is generated.**

**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

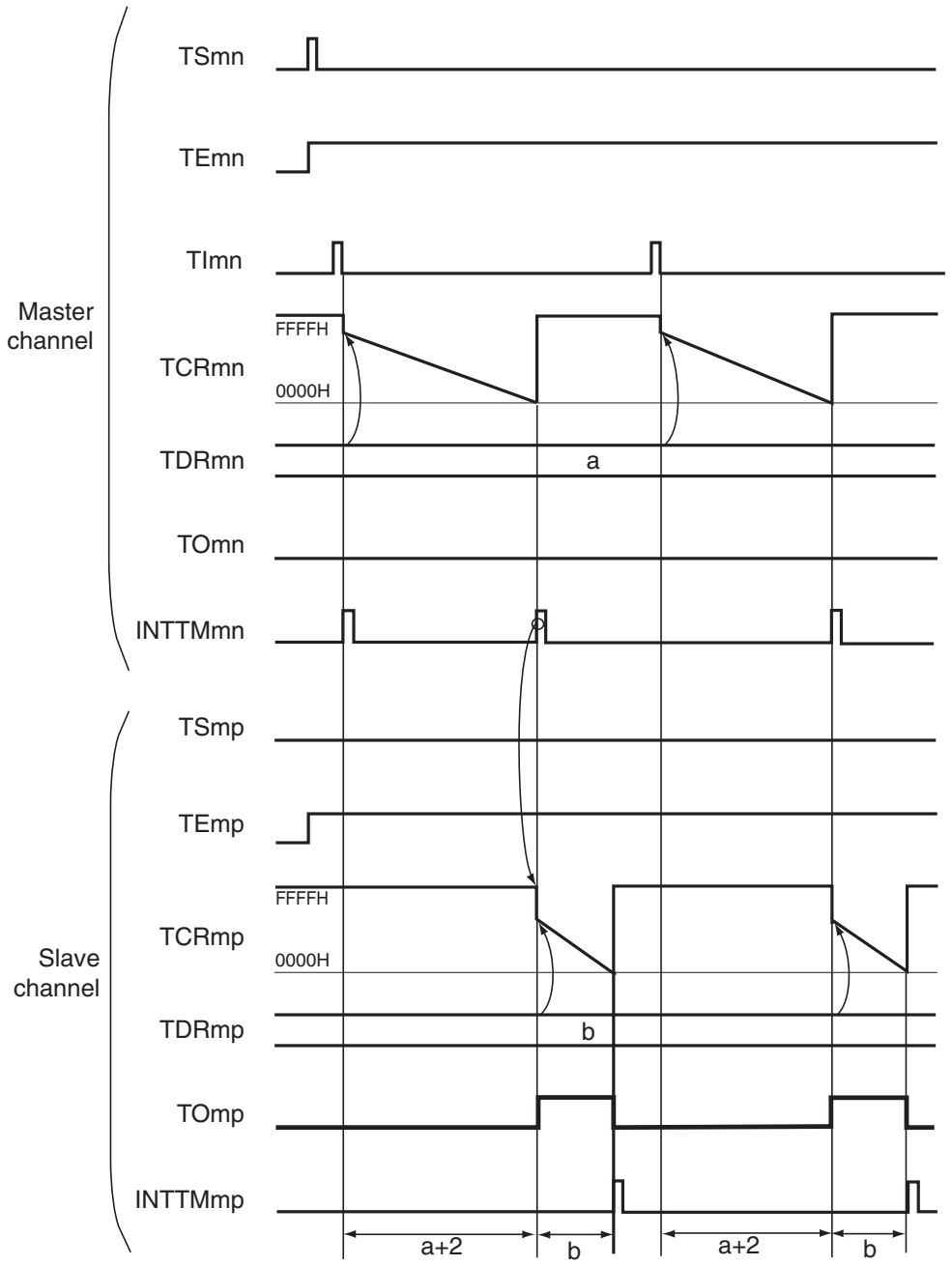
**Figure 6-67. Block Diagram of Operation as One-Shot Pulse Output Function**




**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number ($n < p \leq 7$)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-68. Example of Basic Timing of Operation as One-Shot Pulse Output Function**



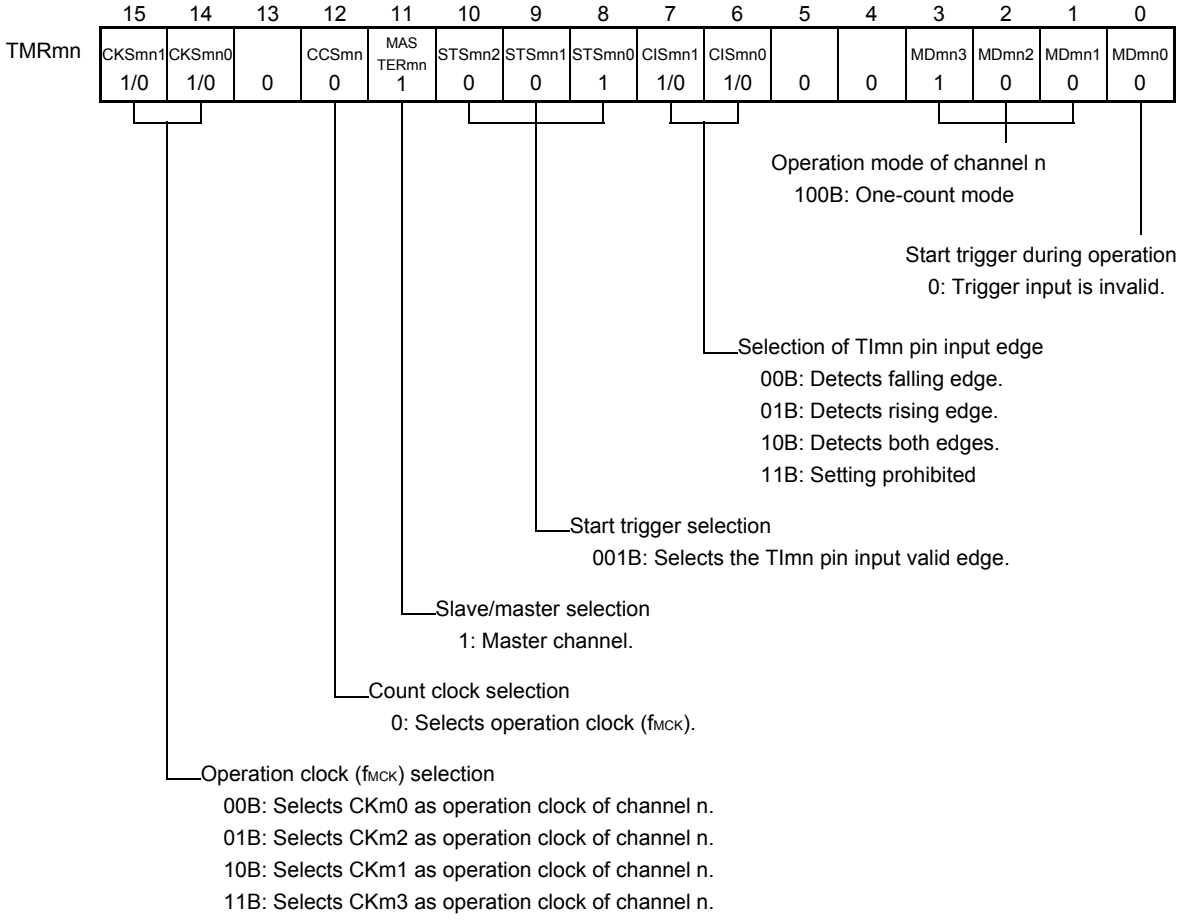

**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

**2.** TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TImn, TImp: TImn and TImp pins input signal
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

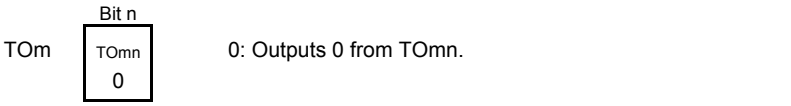
**3.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-69. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)**

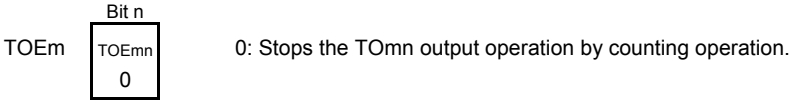
**(a) Timer mode register mn (TMRmn)**



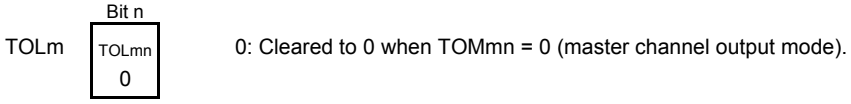

**(b) Timer output register m (TOm)**



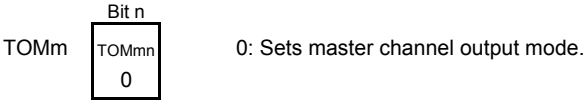

**(c) Timer output enable register m (TOEm)**




**(d) Timer output level register m (TOLm)**




**(e) Timer output mode register m (TOMm)**

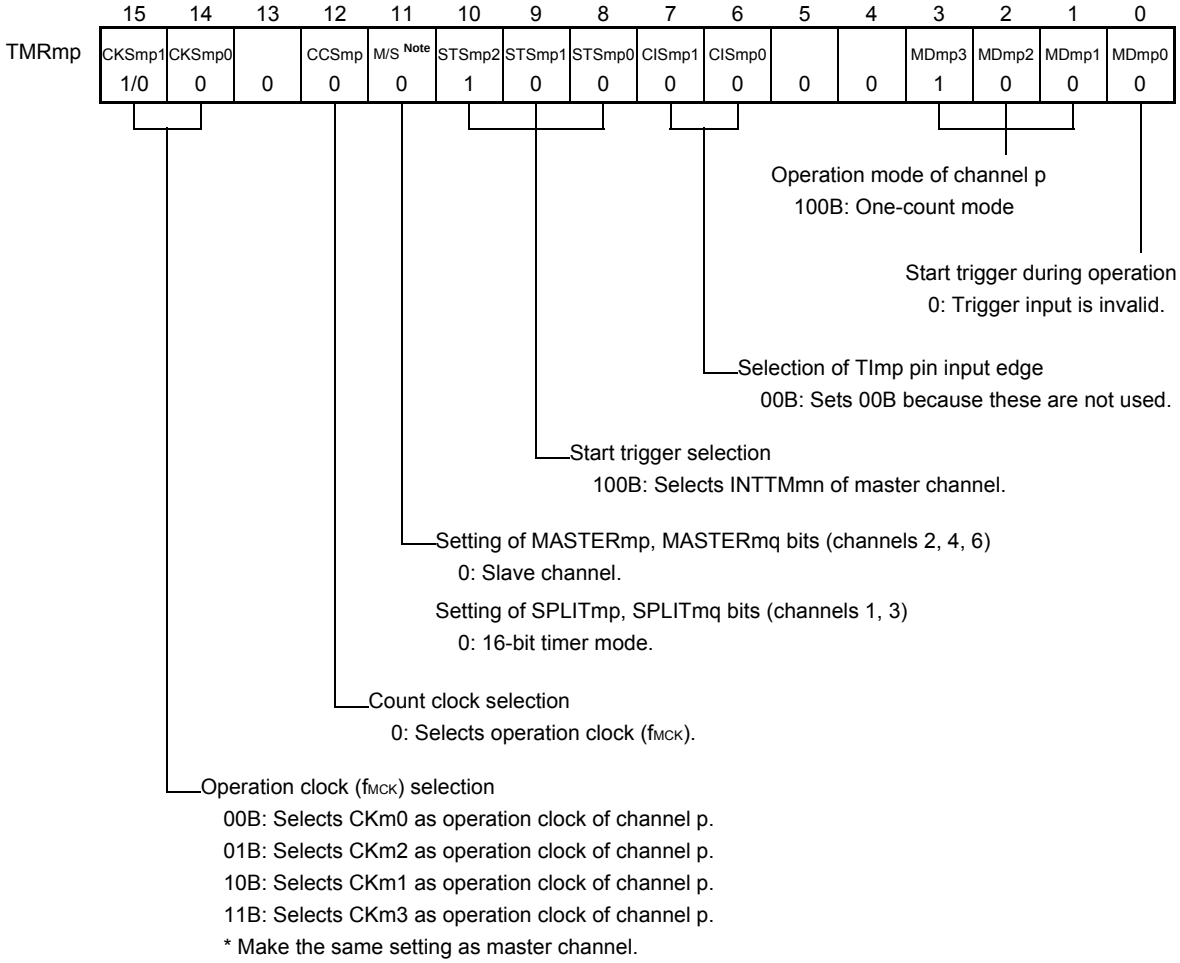



**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-70. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)**

**(a) Timer mode register mp (TMRmp)**




**(b) Timer output register m (TOm)**

| | Bit p | |
|---|---|---|
| TOm | TOmp<br>1/0 | 0: Outputs 0 from TOmp.<br>1: Outputs 1 from TOmp. |

**(c) Timer output enable register m (TOEm)**

| | Bit p | |
|---|---|---|
| TOEm | TOEmp<br>1/0 | 0: Stops the TOmp output operation by counting operation.<br>1: Enables the TOmp output operation by counting operation. |

**(d) Timer output level register m (TOLm)**

| | Bit p | |
|---|---|---|
| TOLm | TOLmp<br>1/0 | 0: Positive logic output (active-high)<br>1: Negative logic output (active-low) |

**(e) Timer output mode register m (TOMm)**

| | Bit p | |
|---|---|---|
| TOMm | TOMmp<br>1 | 1: Sets the slave channel output mode. |

**Note** TMRm2, TMRm4, TMRm6: MASTERmp bit
TMRm1, TMRm3: SPLITmp bit
TMRm5, TMRm7: Fixed to 0

**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-71. Operation Procedure of One-Shot Pulse Output Function (1/2)**

| | Software Operation | Hardware Status |
|---|---|---|
| TAU default setting | | Power-off status<br>(Clock supply is stopped and writing to each register is disabled.) |
| | Sets the TAUmEN bit of peripheral enable registers 0 (PER0) to 1. → | Power-on status. Each channel stops operating.<br>(Clock supply is started and writing to each register is enabled.) |
| | Sets timer clock select register m (TPSm).<br>Determines clock frequencies of CKm0 to CKm3. | |
| Channel default setting | Sets timer mode register mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels).<br>An output delay is set to timer data register mn (TDRmn) of the master channel, and a pulse width is set to the TDRmp register of the slave channel. | Channel stops operating.<br>(Clock is supplied and some power is consumed.) |
| | Sets slave channel.<br>The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode).<br>Sets the TOLmp bit.<br>Sets the TOmp bit and determines default level of the TOmp output. → | The TOmp pin goes into Hi-Z output state.<br>The TOmp default setting level is output when the port mode register is in output mode and the port register is 0. |
| | Sets the TOEmp bit to 1 and enables operation of TOmp. → | TOmp does not change because channel stops operating. |
| | Clears the port register and port mode register to 0. → | The TOmp pin outputs the TOmp set level. |

(**Remarks** are listed on the next page.)

**Figure 6-71. Operation Procedure of One-Shot Pulse Output Function (2/2)**

| | Software Operation | Hardware Status |
|---|---|---|
| Operation start | Sets the TOEmp bit (slave) to 1 (only when operation is resumed).<br>The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → | The TEmn and TEmp bits are set to 1 and the master channel enters the TImn input edge detection wait status.<br>Counter stops operating. |
| | The TSmn and TSmp bits automatically return to 0 because they are trigger bits.<br>Detects the TImn pin input valid edge of master channel. → | Master channel starts counting. |
| During operation | Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed.<br>Sets corresponding bit of noise filter enable registers 1 and 2 (NFEN1 and NFEN2) to 1.<br>Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed.<br>The TCRmn and TCRmp registers can always be read.<br>The TSRmn and TSRmp registers are not used.<br>Set values of the TOm and TOEm registers of slave channel can be changed. | Master channel loads the value of the TDRmn register to timer count register mn (TCRmn) when the TImn pin valid input edge is detected, and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and the counter stops until the next valid edge is input to the TImn pin.<br>The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down.<br>The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped.<br>After that, the above operation is repeated. |
| Operation stop | The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → | TEmn, TEmp = 0, and count operation stops.<br>The TCRmn and TCRmp registers hold count value and stop.<br>The TOmp output is not initialized but holds current status. |
| | The TTmn and TTmp bits automatically return to 0 because they are trigger bits.<br>The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. → | The TOmp pin outputs the TOmp set level. |
| TAU stop | To hold the TOmp pin output level<br>Clears the TOmp bit to 0 after the value to be held is set to the port register. → | The TOmp pin output level is held by port function. |
| | When holding the TOmp pin output level is not necessary<br>Setting not required.<br>The TAUmEN bit of the PER0 register is cleared to 0. → | Power-off status<br>All circuits are initialized and SFR of each channel is also initialized.<br>(The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.) |

Operation is resumed.

**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number ($n < p \leq 7$)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

### 6.8.2 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

| |
|---|
| Pulse period = {Set value of TDRmn (master) + 1} × Count clock period |
| Duty factor [%] = {Set value of TDRmp (slave)}/{Set value of TDRmn (master) + 1} × 100 |
| 0% output: Set value of TDRmp (slave) = 0000H |
| 100% output: Set value of TDRmp (slave) ≥ {Set value of TDRmn (master) + 1} |

**Remark** The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, an interrupt (INTTMmn) is output, the value set to timer data register mn (TDRmn) is loaded to timer count register mn (TCRmn), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTMmn is output, the value of the TDRmn register is loaded again to the TCRmn register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TTmn) of timer channel stop register m (TTm) is set to 1.

If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TOmp) cycle.

The slave channel operates in one-count mode. By using INTTMmn from the master channel as a start trigger, the TCRmp register loads the value of the TDRmp register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTMmp and waits until the next start trigger (INTTMmn from the master channel) is generated.

If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.

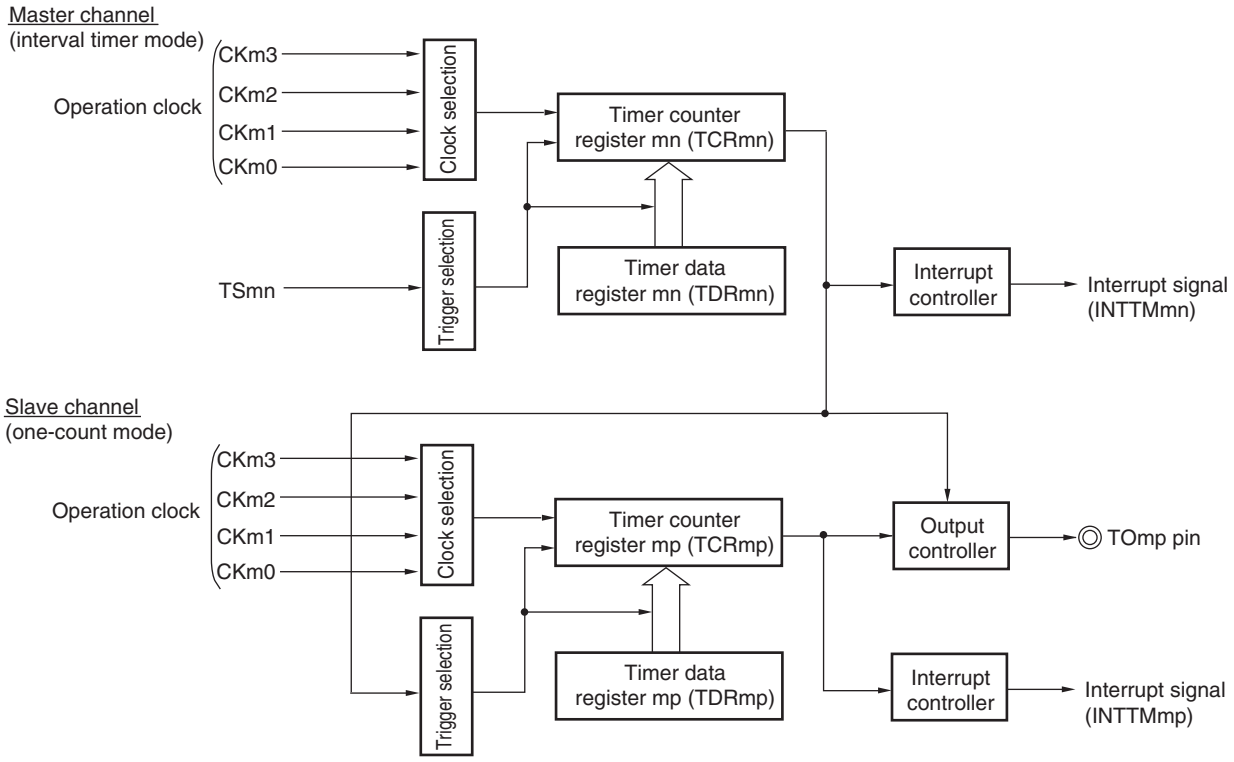
PWM output (TOmp) goes to the active level one clock after the master channel generates INTTMmn and goes to the inactive level when the TCRmp register of the slave channel becomes 0000H.

**Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel, a write access is necessary two times. The timing at which the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.**

**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

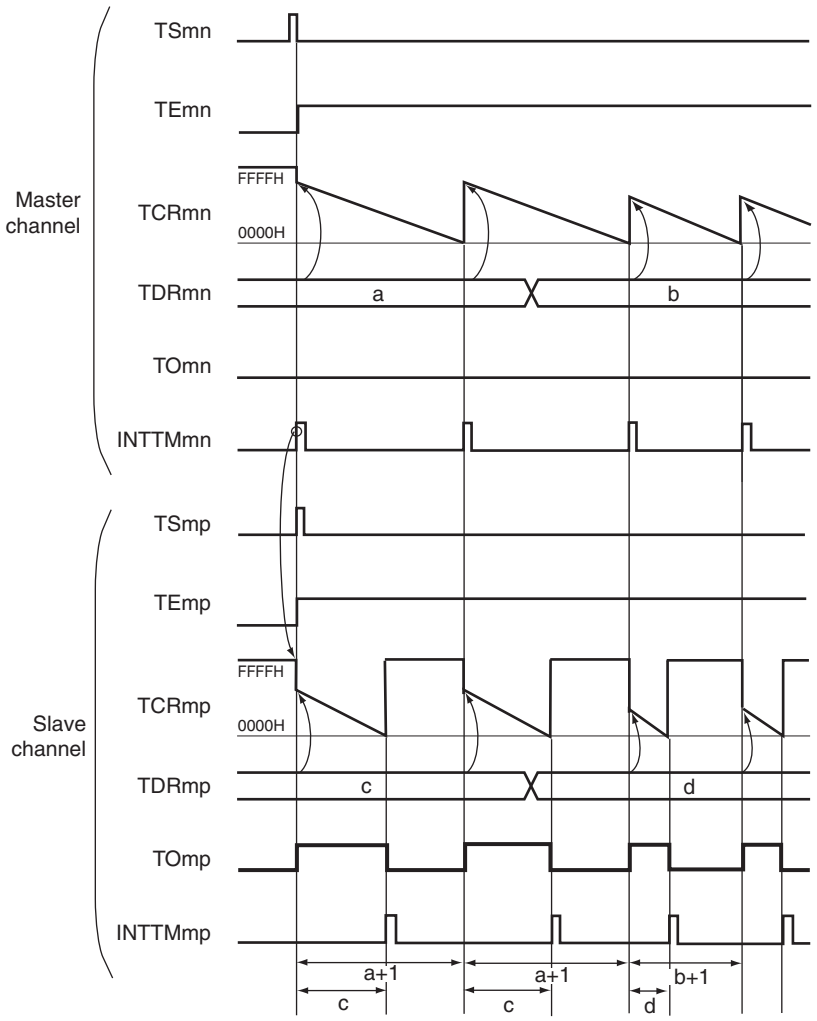
**Figure 6-72. Block Diagram of Operation as PWM Function**




**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number ($n < p \leq 7$)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-73. Example of Basic Timing of Operation as PWM Function**



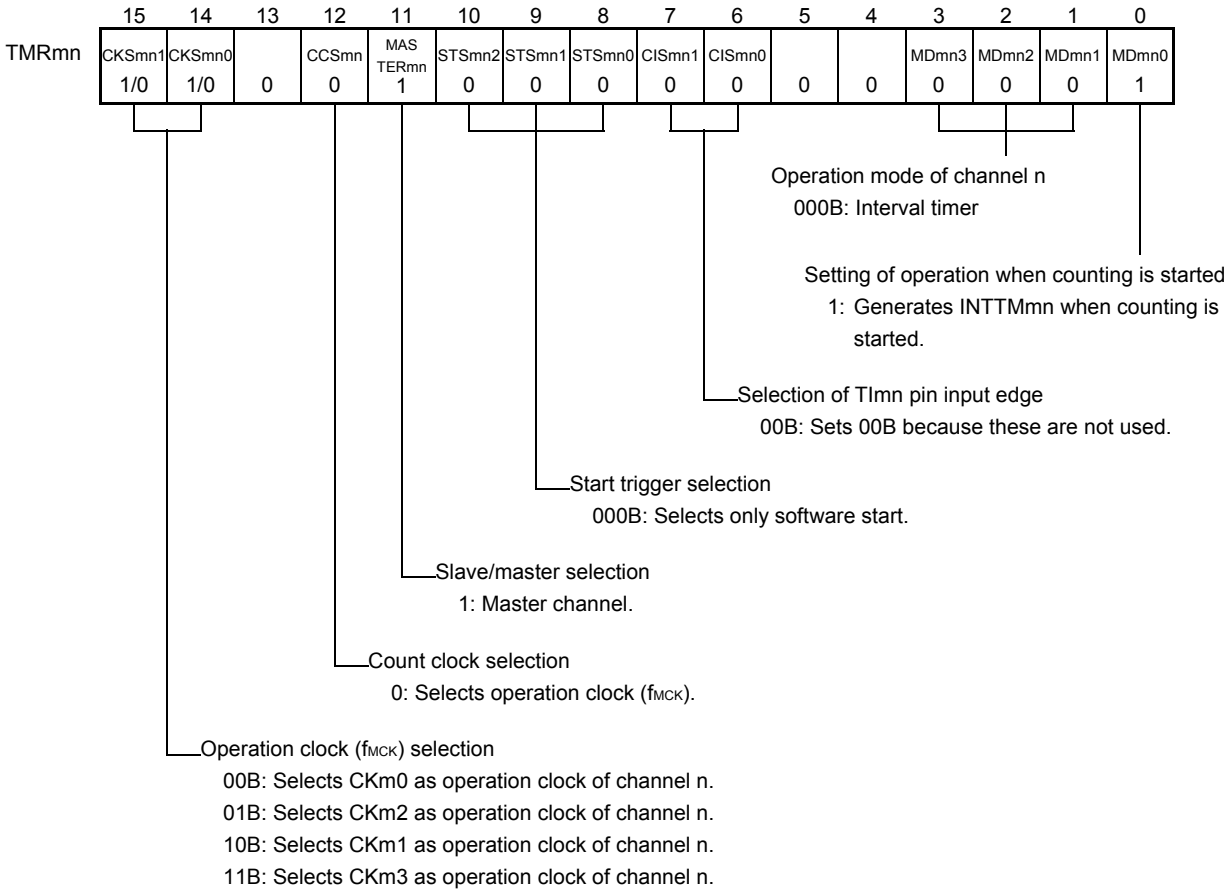

**Remarks1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4 6)
p: Slave channel number (n < p ≤ 7)

**2.** TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

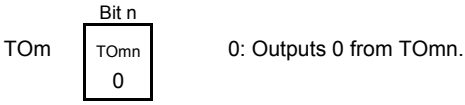
**3.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-74. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used**

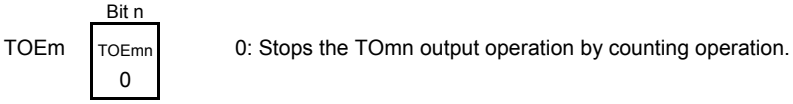
**(a) Timer mode register mn (TMRmn)**



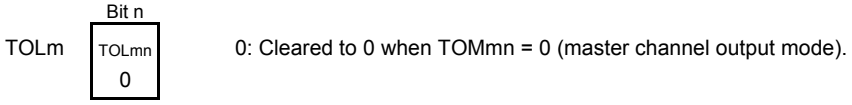

**(b) Timer output register m (TOm)**



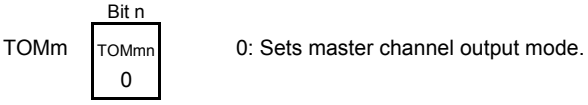

**(c) Timer output enable register m (TOEm)**




**(d) Timer output level register m (TOLm)**




**(e) Timer output mode register m (TOMm)**

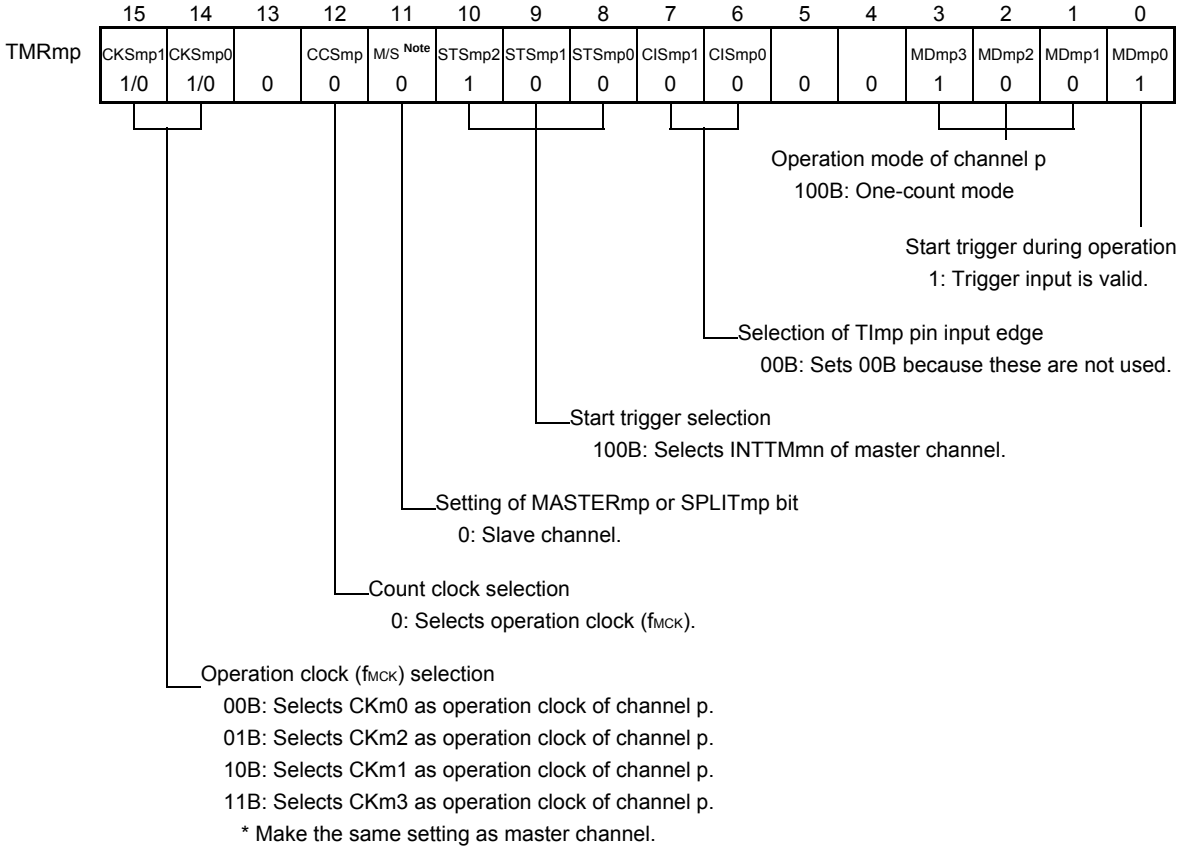



**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)

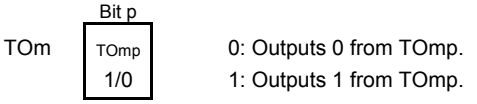
**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-75. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used**

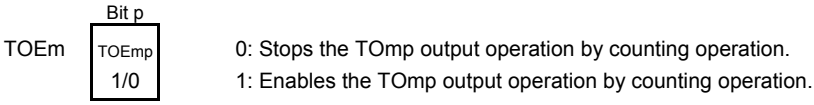
**(a) Timer mode register mp (TMRmp)**



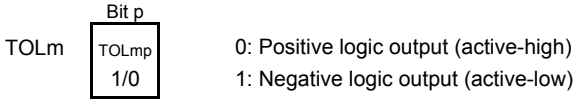

**(b) Timer output register m (TOm)**



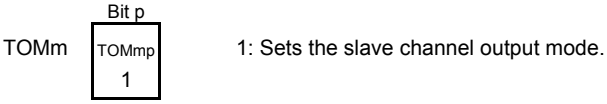

**(c) Timer output enable register m (TOEm)**




**(d) Timer output level register m (TOLm)**




**(e) Timer output mode register m (TOMm)**




**Note** TMRm5, TMRm7: Fixed to 0
TMRm1, TMRm3: SPLITmp bit

**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-76. Operation Procedure When PWM Function Is Used (1/2)**

| | Software Operation | Hardware Status |
|---|---|---|
| TAU default setting | | Power-off status<br>(Clock supply is stopped and writing to each register is disabled.) |
| | Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. → | Power-on status. Each channel stops operating.<br>(Clock supply is started and writing to each register is enabled.) |
| | Sets timer clock select register m (TPSm).<br>Determines clock frequencies of CKm0 to CKm3. | |
| Channel default setting | Sets timer mode registers mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels).<br>An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp register of the slave channel. | Channel stops operating.<br>(Clock is supplied and some power is consumed.) |
| | Sets slave channel.<br>The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode).<br>Sets the TOLmp bit.<br>Sets the TOmp bit and determines default level of the TOmp output. → | The TOmp pin goes into Hi-Z output state.<br>The TOmp default setting level is output when the port mode register is in output mode and the port register is 0. |
| | Sets the TOEmp bit to 1 and enables operation of TOmp. → | TOmp does not change because channel stops operating. |
| | Clears the port register and port mode register to 0. → | The TOmp pin outputs the TOmp set level. |

(**Remarks** are listed on the next page.)

**Figure 6-76. Operation Procedure When PWM Function Is Used (2/2)**

| | Software Operation | Hardware Status |
|---|---|---|
| Operation start | Sets the TOEmp bit (slave) to 1 (only when operation is resumed).<br>The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → | TEmn = 1, TEmp = 1<br>→ When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting. |
| | The TSmn and TSmp bits automatically return to 0 because they are trigger bits. | |
| During operation | Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed.<br>Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated.<br>The TCRmn and TCRmp registers can always be read.<br>The TSRmn and TSRmp registers are not used. | The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn), and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again.<br>At the slave channel, the value of the TDRmp register is loaded to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down.<br>The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped.<br>After that, the above operation is repeated. |
| Operation stop | The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → | TEmn, TEmp = 0, and count operation stops.<br>The TCRmn and TCRmp registers hold count value and stop.<br>The TOmp output is not initialized but holds current status. |
| | The TTmn and TTmp bits automatically return to 0 because they are trigger bits. | |
| | The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. → | The TOmp pin outputs the TOmp set level. |
| TAU stop | To hold the TOmp pin output level<br>Clears the TOmp bit to 0 after the value to be held is set to the port register. → | The TOmp pin output level is held by port function. |
| | When holding the TOmp pin output level is not necessary<br>Setting not required. | |
| | The TAUmEN bit of the PER0 register is cleared to 0. → | Power-off status<br>All circuits are initialized and SFR of each channel is also initialized.<br>(The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.) |

Operation is resumed.

**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

### 6.8.3 Operation as multiple PWM output function

By extending the PWM function and using multiple slave channels, many PWM waveforms with different duty values can be output.

For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDRmp (slave 1)}/{Set value of TDRmn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRmq (slave 2)}/{Set value of TDRmn (master) + 1} × 100

**Remark** Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

Timer count register mn (TCRmn) of the master channel operates in the interval timer mode and counts the periods.

The TCRmp register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. The TCRmp register loads the value of timer data register mp (TDRmp), using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmp = 0000H, TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = 0000H, the TCRmq register outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.

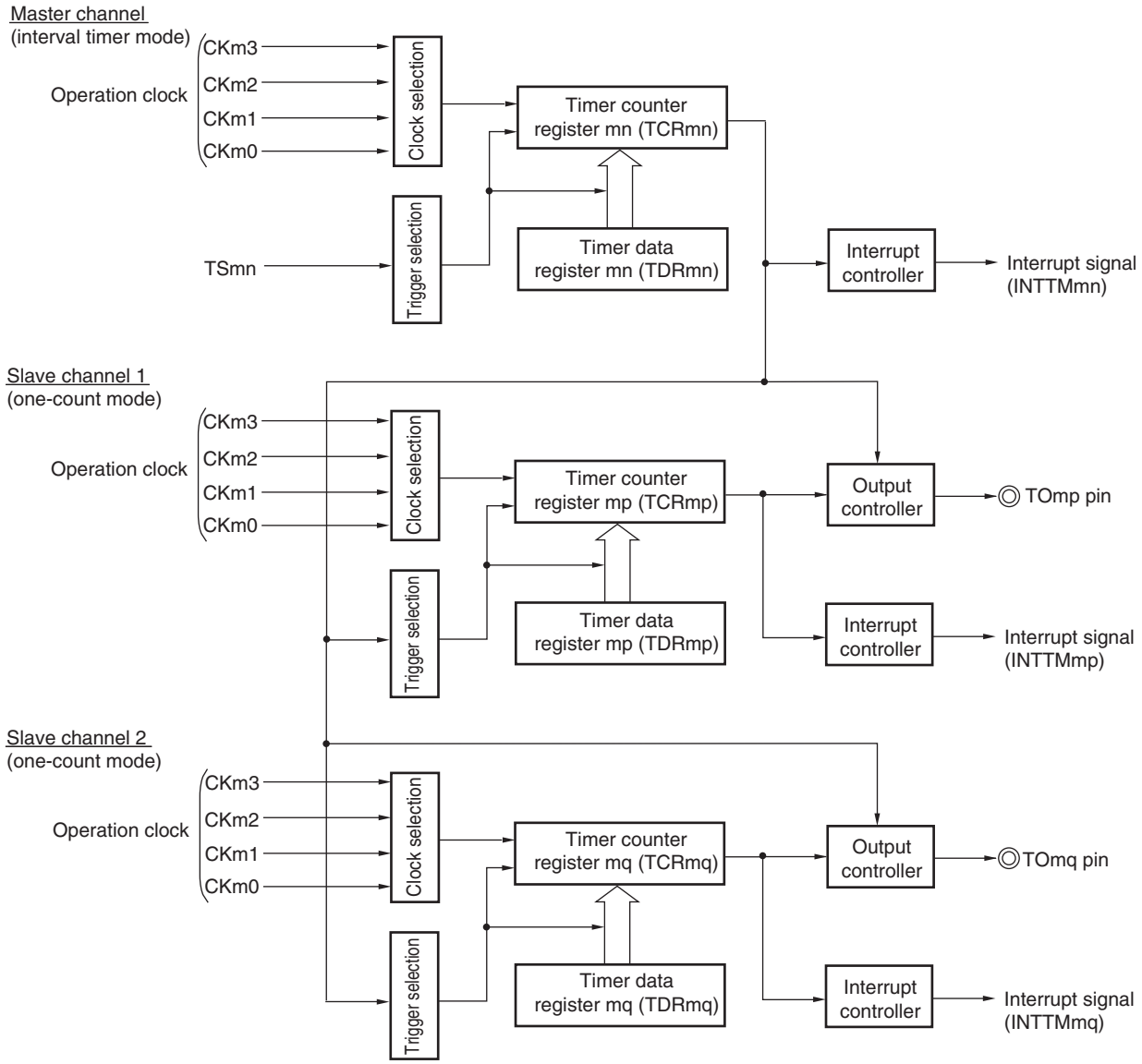
When channel 0 is used as the master channel as above, up to three types of PWM signals can be output at the same time.

**Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1, write access is necessary at least twice. Since the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to the TDRmq register of the slave channel 2).**

**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

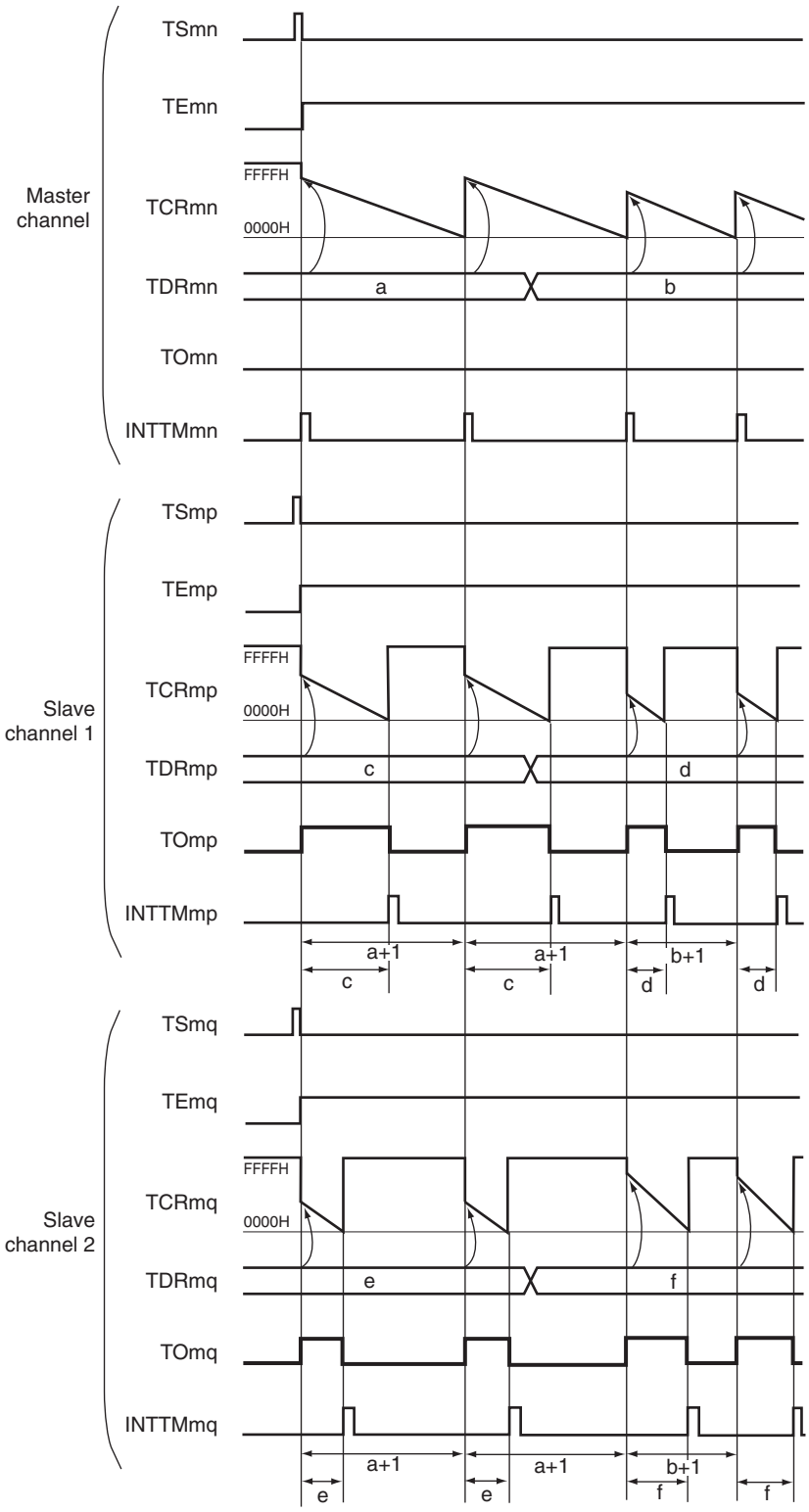
**Figure 6-77. Block Diagram of Operation as Multiple PWM Output Function (output two types of PWMs)**




**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \leq 7$ (Where p and q are integers greater than n)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-78. Example of Basic Timing of Operation as Multiple PWM Output Function (Output two types of PWMs)**




(**Remarks** are listed on the next page.)

**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

**2.** TSmn, TSmp, TSmq: Bit n, p, q of timer channel start register m (TSm)
TEmn, TEmp, TEmq: Bit n, p, q of timer channel enable status register m (TEm)
TCRmn, TCRmp, TCRmq: Timer count registers mn, mp, mq (TCRmn, TCRmp, TCRmq)
TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)
TOmn, TOmp, TOmq: TOmn, TOmp, and TOmq pins output signal

**3.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-79. Example of Set Contents of Registers**
**When Multiple PWM Output Function (Master Channel) Is Used**

**(a) Timer mode register mn (TMRmn)**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmn | CKSmn1 | CKSmn0 | | CCSmn | MAS TERmn | STSmn2 | STSmn1 | STSmn0 | CISmn1 | CISmn0 | | | MDmn3 | MDmn2 | MDmn1 | MDmn0 |
| | 1/0 | 1/0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Master channel.

Count clock selection
0: Selects operation clock ($f_{MCK}$).

Operation clock ($f_{MCK}$) selection
00B: Selects CKm0 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
11B: Selects CKm3 as operation clock of channel n.

**(b) Timer output register m (TOm)**

| | Bit n |
|---|---|
| TOm | TOmn |
| | 0 |

0: Outputs 0 from TOmn.

**(c) Timer output enable register m (TOEm)**

| | Bit n |
|---|---|
| TOEm | TOEmn |
| | 0 |

0: Stops the TOmn output operation by counting operation.

**(d) Timer output level register m (TOLm)**

| | Bit n |
|---|---|
| TOLm | TOLmn |
| | 0 |

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

**(e) Timer output mode register m (TOMm)**

| | Bit n |
|---|---|
| TOMm | TOMmn |
| | 0 |

0: Sets master channel output mode.

**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-80. Example of Set Contents of Registers**
**When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)**

**(a) Timer mode register mp, mq (TMRmp, TMRmq)**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMRmp | CKSmp1 | CKSmp0 | | CCSmp | M/S Note | STSmp2 | STSmp1 | STSmp0 | CISmp1 | CISmp0 | | | MDmp3 | MDmp2 | MDmp1 | MDmp0 |
| | 1/0 | 1/0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| TMRmq | CKSmq1 | CKSmq0 | | CCSmq | M/S Note | STSmq2 | STSmq1 | STSmq0 | CISmq1 | CISmq0 | | | MDmq3 | MDmq2 | MDmq1 | MDmq0 |
| | 1/0 | 1/0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

Operation mode of channel p, q
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp and TImq pins input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmp, MASTERmq bits (channels 2, 4, 6)
0: Slave channel.

Setting of SPLITmp, SPLITmq bits (channels 1, 3)
0: 16-bit timer mode.

Count clock selection
0: Selects operation clock ($f_{MCK}$).

Operation clock ($f_{MCK}$) selection
00B: Selects CKm0 as operation clock of channel p, q.
01B: Selects CKm2 as operation clock of channel p, q.
10B: Selects CKm1 as operation clock of channel p, q.
11B: Selects CKm3 as operation clock of channel p, q.
* Make the same setting as master channel.

**(b) Timer output register m (TOm)**

| | Bit q | Bit p |
|---|---|---|
| TOm | TOmq | TOmp |
| | 1/0 | 1/0 |

0: Outputs 0 from TOmp or TOmq.
1: Outputs 1 from TOmp or TOmq.

**(c) Timer output enable register m (TOEm)**

| | Bit q | Bit p |
|---|---|---|
| TOEm | TOEmq | TOEmp |
| | 1/0 | 1/0 |

0: Stops the TOmp or TOmq output operation by counting operation.
1: Enables the TOmp or TOmq output operation by counting operation.

**(d) Timer output level register m (TOLm)**

| | Bit q | Bit p |
|---|---|---|
| TOLm | TOLmq | TOLmp |
| | 1/0 | 1/0 |

0: Positive logic output (active-high)
1: Negative logic output (active-low)

**(e) Timer output mode register m (TOMm)**

| | Bit q | Bit p |
|---|---|---|
| TOMm | TOMmq | TOMmp |
| | 1 | 1 |

1: Sets the slave channel output mode.

**Note** TMRm5, TMRm7: Fixed to 0
TMRm1, TMRm3: SPLITmp, SPLITmq bit

**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \leq 7$ (Where p and q are integers greater than n)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

**Figure 6-81. Operation Procedure When Multiple PWM Output Function Is Used (output two types of PWMs) (1/2)**

| | Software Operation | Hardware Status |
|---|---|---|
| TAU default setting | | Power-off status<br>(Clock supply is stopped and writing to each register is disabled.) |
| | Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. → | Power-on status. Each channel stops operating.<br>(Clock supply is started and writing to each register is enabled.) |
| | Sets timer clock select register m (TPSm).<br>Determines clock frequencies of CKm0 to CKm3. | |
| Channel default setting | Sets timer mode registers mn, mp, mq (TMRmn, TMRmp, TMRmq) of each channel to be used (determines operation mode of channels).<br>An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp and TDRmq registers of the slave channels. | Channel stops operating.<br>(Clock is supplied and some power is consumed.) |
| | Sets slave channels.<br>The TOMmp and TOMmq bits of timer output mode register m (TOMm) are set to 1 (slave channel output mode).<br>Clears the TOLmp and TOLmq bits to 0.<br>Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. → | The TOmp and TOmq pins go into Hi-Z output state.<br>The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0. |
| | Sets the TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOmq. → | TOmp and TOmq do not change because channels stop operating. |
| | Clears the port register and port mode register to 0. → | The TOmp and TOmq pins output the TOmp and TOmq set levels. |

(**Remarks** are listed on the next page.)

**Figure 6-81. Operation Procedure When Multiple PWM Output Function Is Used (output two types of PWMs) (2/2)**

| | Software Operation | Hardware Status |
|---|---|---|
| Operation start | (Sets the TOEmp and TOEmq (slave) bits to 1 only when resuming operation.)<br>The TSmn bit (master), and TSmp and TSmq (slave) bits of timer channel start register m (TSm) are set to 1 at the same time.<br>The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits. | → TEmn = 1, TEmp, TEmq = 1<br>When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting. |
| During operation | Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed.<br>Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated.<br>The TCRmn, TCRmp, and TCRmq registers can always be read.<br>The TSRmn, TSRmp, and TSRmq registers are not used. | The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn) and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again.<br>At the slave channel 1, the values of the TDRmp register are transferred to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped.<br>At the slave channel 2, the values of the TDRmq register are transferred to TCRmq register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped.<br>After that, the above operation is repeated. |
| Operation stop | The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time.<br>The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits. | → TEmn, TEmp, TEmq = 0, and count operation stops.<br>The TCRmn, TCRmp, and TCRmq registers hold count value and stop.<br>The TOmp and TOmq output are not initialized but hold current status. |
| | The TOEmp and TOEmq bits of slave channels are cleared to 0 and value is set to the TOmp and TOmq bits. | → The TOmp and TOmq pins output the TOmp and TOmq set levels. |
| TAU stop | To hold the TOmp and TOmq pin output levels<br>Clears the TOmp and TOmq bits to 0 after the value to be held is set to the port register.<br>When holding the TOmp and TOmq pin output levels are not necessary<br>Setting not required | → The TOmp and TOmq pin output levels are held by port function. |
| | The TAUmEN bit of the PER0 register is cleared to 0. | → Power-off status<br>All circuits are initialized and SFR of each channel is also initialized.<br>(The TOmp and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.) |

Operation is resumed.

**Remarks 1.** m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are a consecutive integer greater than n)

**2.** Unit 1 is not provided in the Group A products.
Channels 7 to 4 of unit 1 are not provided in the Group B, C, and D products.

## 6.9 Cautions When Using Timer Array Unit

### 6.9.1 Cautions When Using Timer output

(1) When the PCLK (not divided) is selected as the operating clock for the timer array unit and TDRnm (n = 0, 1; m = 0 to 7) are set to 0000H, an interrupt signal from the timer array unit is fixed to high, and an interrupt request cannot be detected.
To use this setting, the interrupt function should be masked.

(2) Do not change the input source for the timer set by the TIS0, TIS1, and TIS2 registers while the timer operates.

# CHAPTER 7 TIMER RJ

Timer RJ is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events.

## 7.1 Overview

This 16-bit timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and they can be accessed by accessing the TRJ0 register.

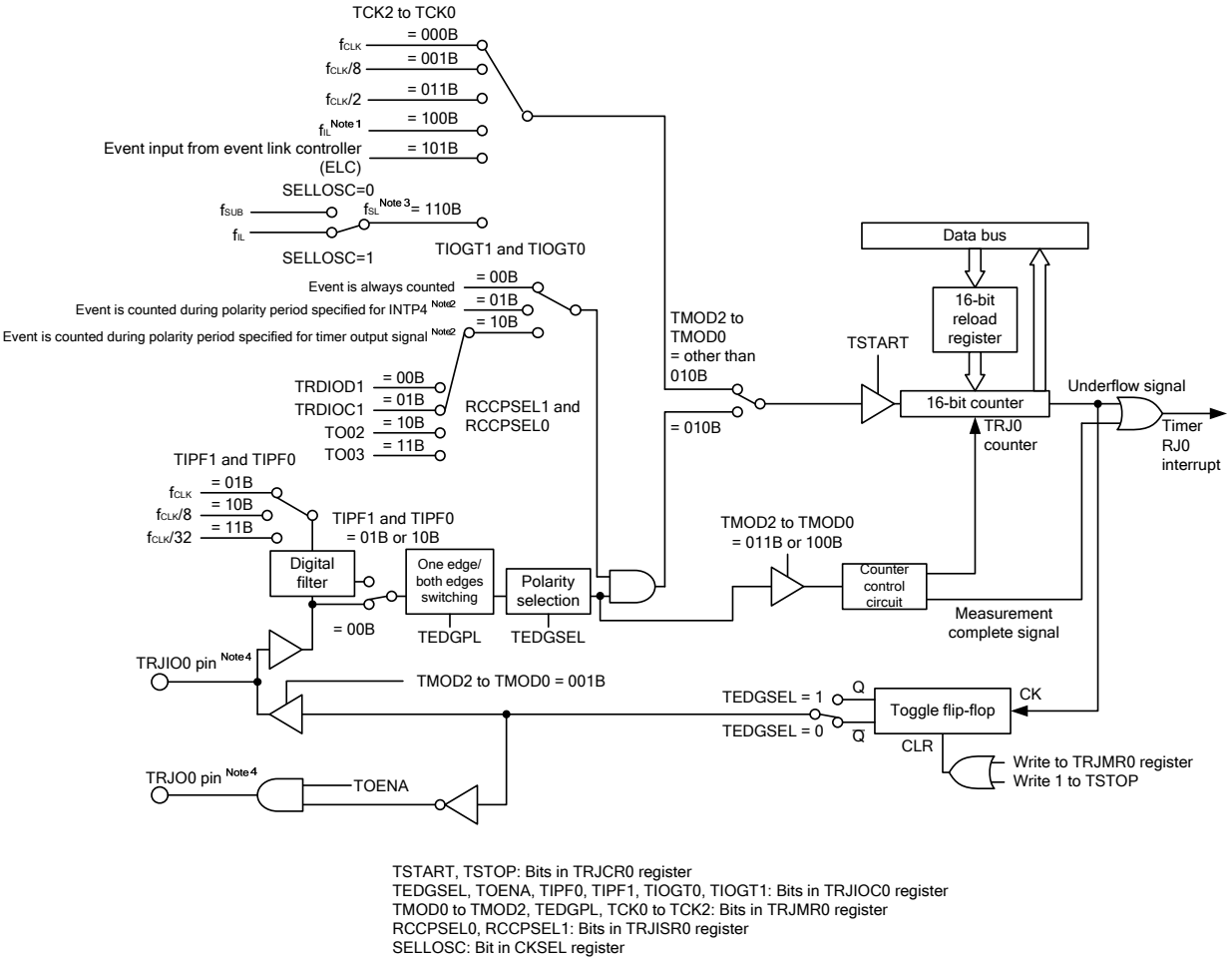
Table 7-1 lists the Timer RJ Specifications. Figure 7-1 shows the Timer RJ Block Diagram.

**Table 7-1. Timer RJ Specifications**

| Item | | Description |
|---|---|---|
| Operating modes | Timer mode | The count source is counted. |
| | Pulse output mode Note | The count source is counted and the output is inverted at each underflow of the timer. |
| | Event counter mode Note | An external event is counted.<br>Operation is possible in STOP mode. |
| | Pulse width measurement mode Note | An external pulse width is measured. |
| | Pulse period measurement mode Note | An external pulse period is measured. |
| Count source (Operating clock) | | $f_{CLK}$, $f_{CLK}/2$, $f_{CLK}/8$, $f_{IL}$, $f_{SL}$, or event input from the event link controller (ELC) selectable |
| Interrupt | | • When the counter underflows.<br>• When the measurement of the active width of the external input (TRJIO0) is completed in pulse width measurement mode.<br>• When the set edge of the external input (TRJIO0) is input in pulse period measurement mode. |
| Selectable functions | | • Coordination with the event link controller (ELC).<br>Event input from the ELC is selectable as a count source. |

**Note** The 20-pin products do not have TRJIO0 and TRJO0 pins. Thus, the pulse output mode, event counter mode, pulse width measurement mode, and pulse period measurement mode cannot be used.

**Figure 7-1. Timer RJ Block Diagram**




TSTART, TSTOP: Bits in TRJCR0 register
TEDGSEL, TOENA, TIPF0, TIPF1, TIOGT0, TIOGT1: Bits in TRJIOC0 register
TMOD0 to TMOD2, TEDGPL, TCK0 to TCK2: Bits in TRJMR0 register
RCCPSEL0, RCCPSEL1: Bits in TRJISR0 register
SELLOSC: Bit in CKSEL register

**Notes 1.** When selecting $f_{IL}$ as the count source, set the WUTMMCK0 bit in the operation speed mode control register (OSMC) to 1.

**2.** The polarity can be selected by the RCCPSEL2 bit in the TRJISR0 register.

**3.** $f_{SUB}$ cannot be selected as the count source for timer RJ when $f_{SL}$ ($f_{IL}$) is selected as the count source for timer RD or the output clock for clock output/buzzer output.

**4.** The 20-pin products do not have TRJIO0 and TRJO0 pins.

## 7.2 I/O Pins

Table 7-2 lists the Timer RJ Pin Configuration.

**Table 7-2. Timer RJ Pin Configuration**

| Pin Name | I/O | Function |
|---|---|---|
| INTP4 | Input | External input for timer RJ |
| TRJIO0 **Note** | Input/output | External event input and pulse output for timer RJ |
| TRJO0 **Note** | Output | Pulse output for timer RJ |

**Note** The 20-pin products do not have TRJIO0 and TRJO0 pins.

## 7.3 Registers

Table 7-3 lists the Timer RJ Register Configuration.

**Table 7-3. Timer RJ Register Configuration**

| Register Name | Symbol | After Reset | Address | Access Size |
|---|---|---|---|---|
| Peripheral Enable Register 1 | PER1 | 00H | F02C0H | 1, 8 |
| Operation speed mode control register | OSMC | 00H | F00F3H | 8 |
| Clock select register | CKSEL | 00H | F02C4H | 1, 8 |
| Timer RJ Counter Register 0 Note 1 | TRJ0 | FFFFH | F06F0H | 16 |
| Timer RJ Control Register 0 | TRJCR0 | 00H | F0240H | 8 |
| Timer RJ I/O Control Register 0 Note 2 | TRJIOC0 | 00H | F0241H | 1, 8 |
| Timer RJ Mode Register 0 | TRJMR0 | 00H | F0242H | 8 |
| Timer RJ Event Pin Select Register 0 Note 2 | TRJISR0 | 00H | F0243H | 8 |
| Port Register 1 | P1 | 00H | FFF01H | 8 |
| Port Register 4 | P4 | 00H | FFF04H | 8 |
| Port Mode Register 1 | PM1 | FFH | FFF21H | 8 |
| Port Mode Register 4 | PM4 | FFH | FFF24H | 8 |

**Notes 1.** When the TRJ0 register is accessed, the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to the TRJ0 register is one clock for both writing and reading.

**2.** The 20-pin products do not have the timer RJ I/O control register 0 (TRJIOC0) and timer RJ event pin select register 0 (TRJISR0).

### 7.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use Timer RJ, be sure to set bit 0 (TRJ0EN) to 1.

Set the PER1 register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 7-2. Format of Peripheral Enable Register 1 (PER1)**

Address: F02C0H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PER1 | DACEN[Note] | 0 | CMPEN[Note] | TRD0EN | DTCEN | 0 | 0 | TRJ0EN |

| TRJ0EN | Control of timer RJ0 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by timer RJ0 cannot be written.<br>• Timer RJ0 is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by timer RJ0 can be read and written. |

**Note** Only on the RL78/F14.

**Cautions 1. When setting timer RJ, be sure to set the TRJ0EN bit to 1 first. If TRJ0EN = 0, writing to a control register of timer RJ is ignored, and all read values are default values (except for port mode registers 1, 4 (PM1, PM4) and port registers 1, 4 (P1, P4)).**

**2. Be sure to set the following bits to 0:**
**RL78/F13: bits 1, 2, 5, 6, and 7**
**RL78/F14: bits 1, 2, and 6**

### 7.3.2 Operation speed mode control register (OSMC)

The low-speed on-chip oscillator can be operated by setting the WUTMMCK0 bit in the OSMC register.

To select the low-speed on-chip oscillator as the count source of the timer RJ, set the bits TCK2 to TCK0 in the timer RJ mode register 0 (TRJMR0).

The RTCLPC bit is used to reduce power consumption by stopping unnecessary clock functions.

For the setting of the RTCLPC bit, see **CHAPTER 5 CLOCK GENERATOR**.

Set the OSMC register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 7-3. Format of Operation Speed Mode Control Register (OSMC)**

Address: F00F3H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSMC | RTCLPC | 0 | 0 | WUTMMCK0 | 0 | 0 | 0 | 0 |

| WUTMMCK0 | Low-speed on-chip oscillator operation control |
|---|---|
| 0 | Low-speed on-chip oscillator operating |
| 1 | Low-speed on-chip oscillator stopped |

### 7.3.3 Clock Select Register (CKSEL)

This register is used to select the CPU clock ($f_{SUB}$/$f_{IL}$) the clocks for the timer RJ, timer RD, and clock output/buzzer output. Together with the CMC register, the SELLOSC bit is used to set the operation mode of the subsystem clock. For details, see **Figure 5-3 Format of Clock Operation Mode Control Register (CMC)**.

Set the CKSEL register by a 1-bit or 8-bit memory manipulation instruction.

Writing to the CKSEL register is disabled when the GCSC bit of the IAWCTL register is set to 1.

**Figure 7-4. Format of Clock Select Register (CKSEL)**

Address: F02C4H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| CKSEL | 0 | 0 | 0 | 0 | 0 | TRD_CKSEL | 0 | SELLOSC Notes 3, 4 |

| SELLOSC Notes 3, 4 | Control of sub/low-speed on-chip oscillator selection clock ($f_{SL}$) selection |
|---|---|
| 0 | Selects $f_{SUB}$ Note 1 |
| 1 | Selects $f_{IL}$ Note 2 |

**Notes 1.** When setting $f_{SUB}$ as the CPU/peripheral hardware clock, first set the SELLOSC bit to 0 and then set the CSS bit in the CKC register to 1.

**2.** When setting $f_{IL}$ as the CPU/peripheral hardware clock, first set the SELLOSC bit to 1 and then set the CSS bit in the CKC register to 1.

**3.** When the SELLOSC bit is set to 1, the low-speed on-chip oscillator operates.

**4.** When setting the CKSEL register in the 20-, 30-, or 32-pin products, set the SELLOSC bit to 1.

### 7.3.4 Timer RJ Counter Register 0 (TRJ0), Timer RJ Reload Register

TRJ0 is a 16-bit register. The write value is written to the reload register and the read value is read from the counter.

The states of the reload register and the counter are changed depending on the TSTART bit in the TRJCR0 register.

For details, see **7. 4. 1 Reload Register and Counter Rewrite Operation**.

**Figure 7-5. Format of Timer RJ Counter Register 0 (TRJ0), Timer RJ Reload Register**

Address : F06F0H After Reset: FFFFH

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRJ0 | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – | – |

| — | Function | Setting Range | R/W |
|---|---|---|---|
| Bits 15 to 0 | 16-bit counter and reload register Notes 1, 2, 3 | 0000H to FFFFH | R/W |

**Notes 1.** When 1 is written to the TSTOP bit in the TRJCR0 register, the 16-bit counter is forcibly stopped and set to FFFFH.

**2.** The TRJ0 register must be accessed in 16-bit units. Do not access this register in 8-bit units.

**3.** When the setting of bits TCK2 to TCK0 in the TRJMR0 register is other than 001B ($f_{CLK}$/8) or 011B ($f_{CLK}$/2), if the TRJ0 register is set to 0000H, a request signal to the data transfer controller (DTC) and the event link controller (ELC) is generated only once immediately after the count starts. However, the TRJO0 and TRJIO0 output is toggled.

When the TRJ0 register is set to 0000H in event counter mode, regardless of the value of bits TCK2 to TCK0, a request signal to the DTC, ELC, and interrupt functions is generated only once immediately after the count starts. In addition, the TRJO0 output is toggled even during a period other than the specified count period.

When the TRJ0 register is set to 0000H or a higher value, a request signal is generated each time TRJ underflows.

**Caution When the TRJ0 register is accessed, the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to the TRJ0 register is one clock for both writing and reading.**

### 7.3.5 Timer RJ Control Register 0 (TRJCR0)

**Figure 7-6. Format of Timer RJ Control Register 0 (TRJCR0)**

Address : F0240H After Reset: 00H

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRJCR0 | — | — | TUNDF | TEDGF | — | TSTOP | TCSTF | TSTART |

| Bits 7 to 6 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| TUNDF | Timer RJ underflow flag Note 1 | R/W |
|---|---|---|
| 0 | No underflow | R/W |
| 1 | Underflow | |
| [Condition for setting to 0]<br>• When 0 is written to this bit by a program.<br>[Condition for setting to 1]<br>• When the counter underflows. | | |

| TEDGF | Active edge judgement flag Notes 1, 2 | R/W |
|---|---|---|
| 0 | No active edge received | R/W |
| 1 | Active edge received | |
| [Condition for setting to 0]<br>• When 0 is written to this bit by a program.<br>[Conditions for setting to 1]<br>• When the measurement of the active width of the external input (TRJIO0) is completed in pulse width measurement mode.<br>• The set edge of the external input (TRJIO0) is input in pulse period measurement mode. | | |

| Bit 3 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| TSTOP | Timer RJ count forced stop Note 3 | R/W |
|---|---|---|
| When 1 is written to this bit, the count is forcibly stopped. The read value is 0. | | W |

| TCSTF | Timer RJ count status flag Note 4 | R/W |
|---|---|---|
| 0 | Count stops | R |
| 1 | Count in progress | |
| [Conditions for setting to 0]<br>• When 0 is written to the TSTART bit (the TCSTF bit is set to 0 in synchronization with the count source).<br>• When 1 is written to the TSTOP bit.<br>[Condition for setting to 1]<br>• When 1 is written to the TSTART bit (the TCSTF bit is set to 1 in synchronization with the count source). | | |

(Notes are listed on the next page.)

| TSTART | Timer RJ count start Note 4 | R/W |
|---|---|---|
| 0 | Count stops | R/W |
| 1 | Count starts | |
| Count operation is started by writing 1 to the TSTART bit and stopped by writing 0. When the TSTART bit is set to 1 (count starts), the TCSTF bit is set to 1 (count in progress) in synchronization with the count source. Also, after 0 is written to the TSTART bit, the TCSTF bit is set to 0 (count stops) in synchronization with the count source. For details, see **7. 5. 1 Count Operation Start and Stop Control.** | | |

**Notes 1.** Set the TRJCR0 register by an 8-bit memory manipulation instruction.

**2.** The 20-pin products do not have the active edge judgement flag.

**3.** When 1 (count is forcibly stopped) is written to the TSTOP bit, bits TSTART and TCSTF are initialized at the same time. The pulse output level is also initialized.

**4.** For notes on using bits TSTART and TCSTF, see **7. 5. 1 Count Operation Start and Stop Control.**

### 7.3.6 Timer RJ I/O Control Register 0 (TRJIOC0)

**Figure 7-7. Format of Timer RJ I/O Control Register 0 (TRJIOC0)**

Address : F0241H After Reset: 00H

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRJIOC0 | TIOGT1 | TIOGT0 | TIPF1 | TIPF0 | — | TOENA | — | TEDGSEL |

| TIOGT1 | TIOGT0 | TRJIO0 count control Notes 1, 2 | R/W |
|---|---|---|---|
| 0 | 0 | Event is always counted | R/W |
| 0 | 1 | Event is counted during polarity period specified for INTP4 | |
| 1 | 0 | Event is counted during polarity period specified for timer output signal | |
| 1 | 1 | Do not set. | |

**Notes 1.** When INTP4 or the timer output signal is used, the polarity to count an event can be selected by the RCCPSEL2 bit in the TRJISR0 register.

**2.** Bits TIOGT0 and TIOGT1 are enabled only in event counter mode.

| TIPF1 | TIPF0 | TRJIO0 input filter select | R/W |
|---|---|---|---|
| 0 | 0 | No filter | R/W |
| 0 | 1 | Filter sampled at $f_{CLK}$ | |
| 1 | 0 | Filter sampled at $f_{CLK}/8$ | |
| 1 | 1 | Filter sampled at $f_{CLK}/32$ | |
| These bits are used to specify the sampling frequency of the filter for the TRJIO0 input. If the input to the TRJIO0 pin is sampled and the value matches three successive times, that value is taken as the input value. | | | |

| Bit 3 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| TOENA | TRJO0 output enable | R/W |
|---|---|---|
| 0 | TRJO0 output disabled (port) | R/W |
| 1 | TRJO0 output enabled | |

| Bit 1 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| TEDGSEL | I/O polarity switch | R/W |
|---|---|---|
| Function varies depending on the operating mode (see **Table 7-4** and **Table 7-5**). The TEDGSEL bit is used to switch the TRJO0 output polarity and the TRJIO0 I/O edge and polarity. In pulse output mode, only the inversion/non-inversion of toggle flip-flop is controlled. The toggle flip-flop is initialized when the TRJMR0 register is written or 1 is written to the TSTOP bit in the TRJCR0 register. | | R/W |

**Caution The 20-pin products do not have the timer RJ I/O control register 0 (TRJIOC0).**

**Table 7-4. TRJIO0 I/O Edge and Polarity Switching**

| Operating Mode | Function |
|---|---|
| Timer mode | Not used (I/O port) |
| Pulse output mode | 0: Output is started at high (Initialization level: High)<br>1: Output is started at low (Initialization level: Low) |
| Event counter mode | 0: Count at rising edge<br>1: Count at falling edge |
| Pulse width measurement mode | 0: Low-level width is measured<br>1: High-level width is measured |
| Pulse period measurement mode | 0: Measure from one rising edge to the next rising edge<br>1: Measure from one falling edge to the next falling edge |

**Table 7-5. TRJO0 Output Polarity Switching**

| Operating Mode | Function |
|---|---|
| All modes | 0: Output is started at low (Initialization level: Low)<br>1: Output is started at high (Initialization level: High) |

### 7.3.7 Timer RJ Mode Register 0 (TRJMR0)

**Figure 7-8. Format of Timer RJ Mode Register 0 (TRJMR0)**

Address : F0242H After Reset: 00H

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRJMR0 | — | TCK2 | TCK1 | TCK0 | TEDGPL | TMOD2 | TMOD1 | TMOD0 |

| Bit 7 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| TCK2 | TCK1 | TCK0 | Timer RJ count source select Notes 1, 2 | R/W |
|---|---|---|---|---|
| 0 | 0 | 0 | $f_{CLK}$ | R/W |
| 0 | 0 | 1 | $f_{CLK}/8$ | |
| 0 | 1 | 1 | $f_{CLK}/2$ | |
| 1 | 0 | 0 | $f_{IL}$ Note 4 | |
| 1 | 0 | 1 | Event input from event link controller (ELC) Note 5 | |
| 1 | 1 | 0 | $f_{SL}$ | |
| Other than above | | | Setting prohibited | |

| TEDGPL | TRJIO0 edge polarity select Note 6 | R/W |
|---|---|---|
| 0 | One edge | R/W |
| 1 | Both edges | |

| TMOD2 | TMOD1 | TMOD0 | Timer RJ operating mode select Note 3 | R/W |
|---|---|---|---|---|
| 0 | 0 | 0 | Timer mode | R/W |
| 0 | 0 | 1 | Pulse output mode Note 7 | |
| 0 | 1 | 0 | Event counter mode Note 7 | |
| 0 | 1 | 1 | Pulse width measurement mode Note 7 | |
| 1 | 0 | 0 | Pulse period measurement mode Note 7 | |
| Other than above | | | Setting prohibited | |

**Notes 1.** When event counter mode is selected, the external input (TRJIO0) is selected as the count source regardless of the setting of bits TCK0 to TCK2.

**2.** Do not switch count sources during count operation. When switching count sources, set the TSTART and TCSTF bits in the TRJCR0 register to 0 (count stops).

**3.** The operating mode can be changed only when the count is stopped while both the bits TSTART and TCSTF in the TRJCR0 register are set to 0 (count stops). Do not change the operating mode during count operation.

**4.** When selecting $f_{IL}$ as the count source, set the WUTMMCK0 bit in the operation speed mode register (OSMC) to 1.
However, $f_{SUB}$ cannot be selected as the count source for timer RJ when the SELLOSC bit in the CKSEL register is set to 1.

**5.** Only available in the RL78/F14. Do not make the setting in other products.

**6.** The TEDGPL bit is enabled only in event counter mode.

**7.** The 20-pin products do not have TRJIO0 and TRJO0 pins. Thus, the pulse output mode, event counter mode, pulse width measurement mode, and pulse period measurement mode cannot be used.

**Caution Write access to the TRJMR0 register initializes the output from pins TRJO0 and TRJIO0 of timer RJ. For details on the output level at initialization, refer to the description shown below Figure 7-7 Format of Timer RJ I/O Control Register 0 (TRJIOC0).**

### 7.3.8 Timer RJ Event Pin Select Register 0 (TRJISR0)

**Figure 7-9. Format of Timer RJ Event Pin Select Register 0 (TRJISR0)**

Address : F0243H After Reset: 00H

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRJISR0 | — | — | — | — | — | RCCPSEL2 **Note** | RCCPSEL1 **Note** | RCCPSEL0 **Note** |

| Bit 7 to 3 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| RCCPSEL2 | Timer output signal and INTP4 polarity selection | R/W |
|---|---|---|
| 0 | An event is counted during the low-level period | R/W |
| 1 | An event is counted during the high-level period | |

| RCCPSEL1 | RCCPSEL0 | Timer output signal selection | R/W |
|---|---|---|---|
| 0 | 0 | TRDIOD1 | R/W |
| 0 | 1 | TRDIOC1 | |
| 1 | 0 | TO02 | |
| 1 | 1 | TO03 | |

**Note** Bits RCCPSEL0 to RCCPSEL2 are enabled only in event counter mode.

**Caution The 20-pin products do not have the timer RJ event pin select register 0 (TRJISR0).**

### 7.3.9 Port mode registers 1, 4 (PM1, PM4)

These registers set input/output of ports 1 and 4 in 1-bit units.

When using the ports (such as P41/TRJIO0 and P10/TRJO0) to be shared with the timer output pin for timer output, set the port mode register (PMxx) bit and the port register (Pxx) bit corresponding to each port to 0.

Example: When using P41/TRJIO0 for timer output
Set the PM41 bit of port mode register 4 to 0.
Set the P41 bit of port register 4 to 0.

When using the ports (such as P41/TRJIO0) to be shared with the timer input pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P41/TRJIO0 for timer input
Set the PM41 bit of port mode register 4 to 1.
Set the P41 bit of port register 4 to 1.

The PM1 and PM4 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

**Figure 7-10. Format of Port Mode Registers 1, 4 (PM1, PM4) (100-pin products)**

Address: FFF21H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM1 | PM17 | PM16 | PM15 | PM14 | PM13 | PM12 | PM11 | PM10 |

Address: FFF24H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM4 | PM47 | PM46 | PM45 | PM44 | PM43 | PM42 | PM41 | PM40 |

| PMmn | Pmn pin I/O mode selection (m = 1, 4 ; n = 0 to 7) |
|---|---|
| 0 | Output mode (output buffer on) |
| 1 | Input mode (output buffer off) |

**Remark** The figure shown above presents the format of port mode registers 1 and 4 of the 100-pin products. The format of the port mode register of other products, see **4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function.**

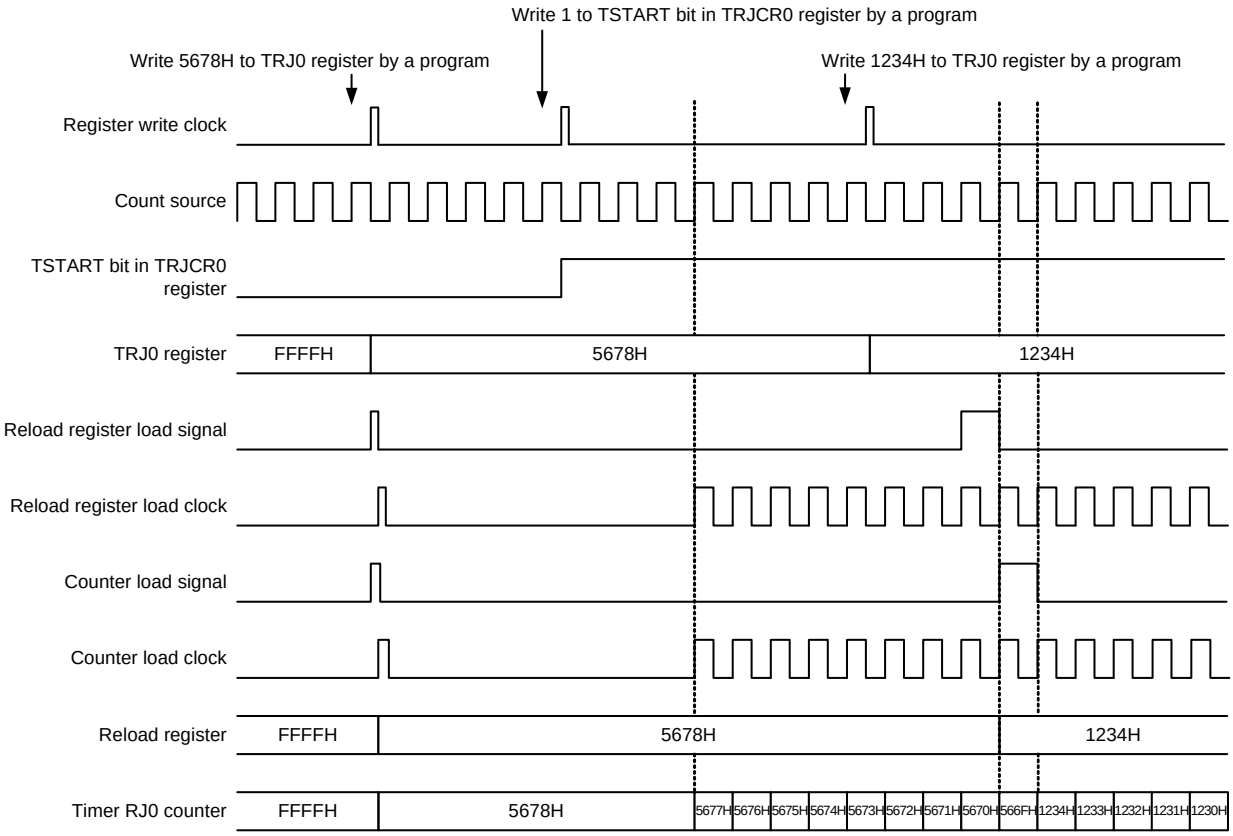
## 7.4 Operation

### 7.4.1 Reload Register and Counter Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter differs depending on the value in the TSTART bit in the TRJCR0 register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (count starts), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source.

Figure 7-11 shows the Timing of Rewrite Operation with TSTART Bit Value.

**Figure 7-11. Timing of Rewrite Operation with TSTART Bit Value**



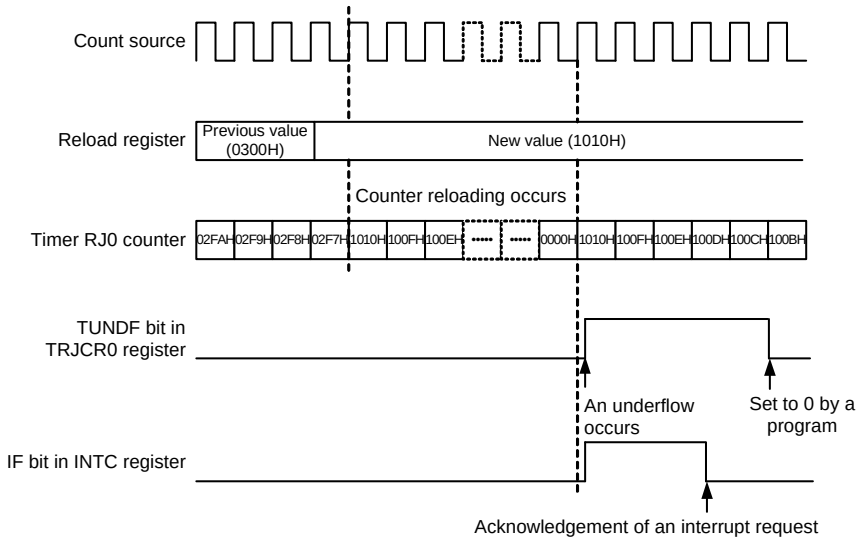
### 7.4.2 Timer Mode

In this mode, the counter is decremented by the count source selected by bits TCK0 to TCK2 in the TRJMR0 register.

In timer mode, the count value is decremented by 1 each time the count source is input. When the count value reaches 0000H and the next count source is input, an underflow occurs and an interrupt request is generated.

Figure 7-12 shows the Operation Example in Timer Mode.

**Figure 7-12. Operation Example in Timer Mode**



### 7.4.3 Pulse Output Mode

In this mode, the counter is decremented by the count source selected by bits TCK0 to TCK2 in the TRJMR0 register, and the output level of pins TRJIO0 and TRJO0 pin is inverted each time an underflow occurs.

In pulse output mode, the count value is decremented by 1 each time the count source is input. When the count value reaches 0000H and the next count source is input, an underflow occurs and an interrupt request is generated.

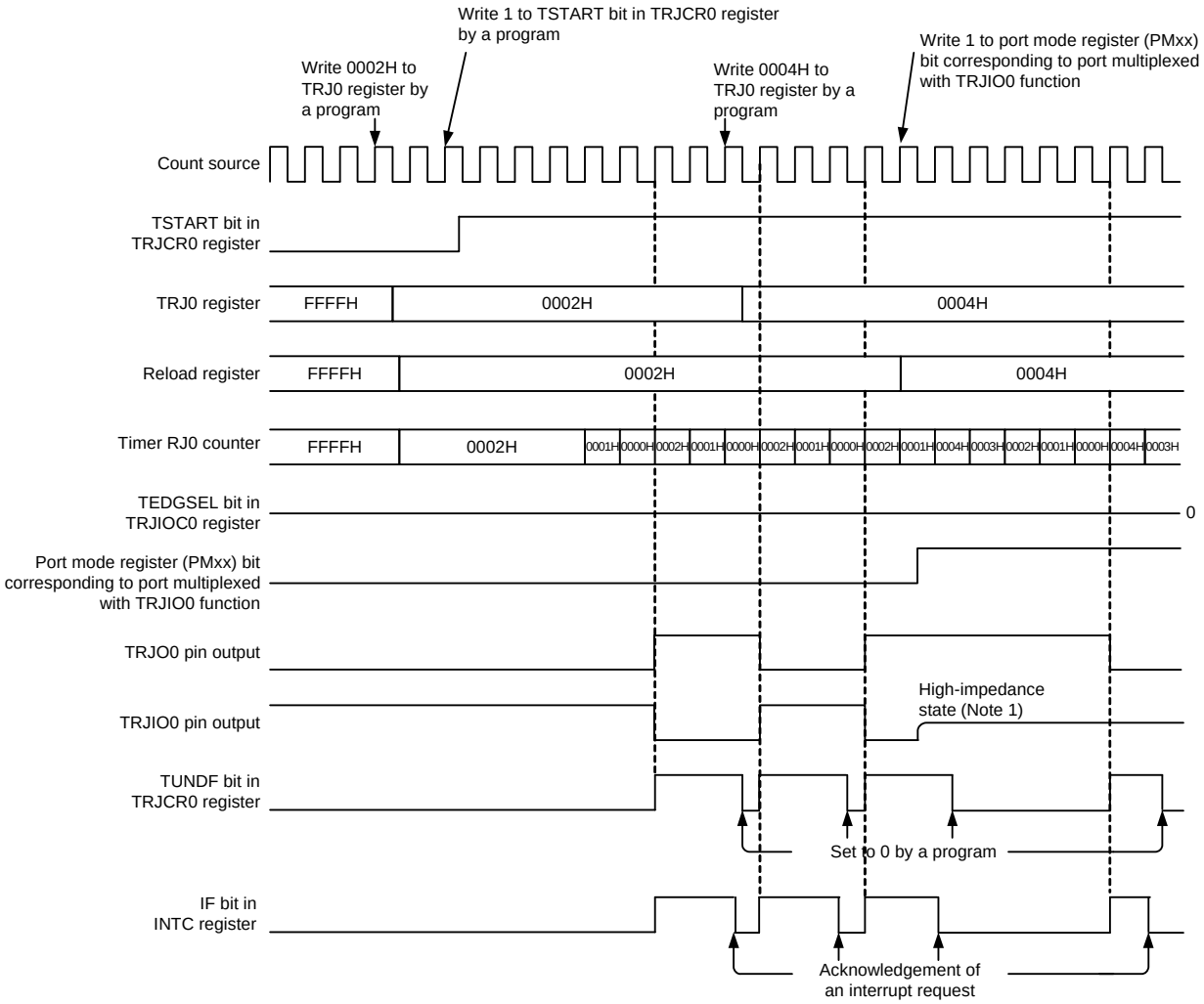
In addition, a pulse can be output from pins TRJIO0 and TRJO0. The output level is inverted each time an underflow occurs. The pulse output from the TRJO0 pin can be stopped by the TOENA bit in the TRJIOC0 register.

Also, the output level can be selected by the TEDGSEL bit in the TRJIOC0 register.

Figure 7-13 shows the Operation Example in Pulse Output Mode.

**Caution The 20-pin products do not have TRJIO0 and TRJO0 pins. Thus, the pulse output mode cannot be used.**

**Figure 7-13. Operation Example in Pulse Output Mode**




Note 1: The TRJIO0 pin becomes high impedance by output enable control on the port selected as the TRJIO0 function.

### 7.4.4 Event Counter Mode

In this mode, the counter is decremented by an external event signal (count source) input to the TRJIO0 pin. Various periods for counting events can be set by bits TIOGT0 and TIOGT1 in the TRJIOC0 register and the TRJISR0 register. In addition, the filter function for the TRJIO0 input can be specified by bits TIPF0 and TIPF1 in the TRJIOC0 register.

Also, the output from the TRJO0 pin can be toggled even in event counter mode.

When event counter mode is used, see **7. 5. 5 Procedure for Setting Pins TRJO0 and TRJIO0**.

Figure 7-14 shows the Operation Example in Event Counter Mode.

**Caution The 20-pin products do not have TRJIO0 and TRJO0 pins. Thus, the event counter mode cannot be used.**

**Figure 7-14. Operation Example 1 in Event Counter Mode**

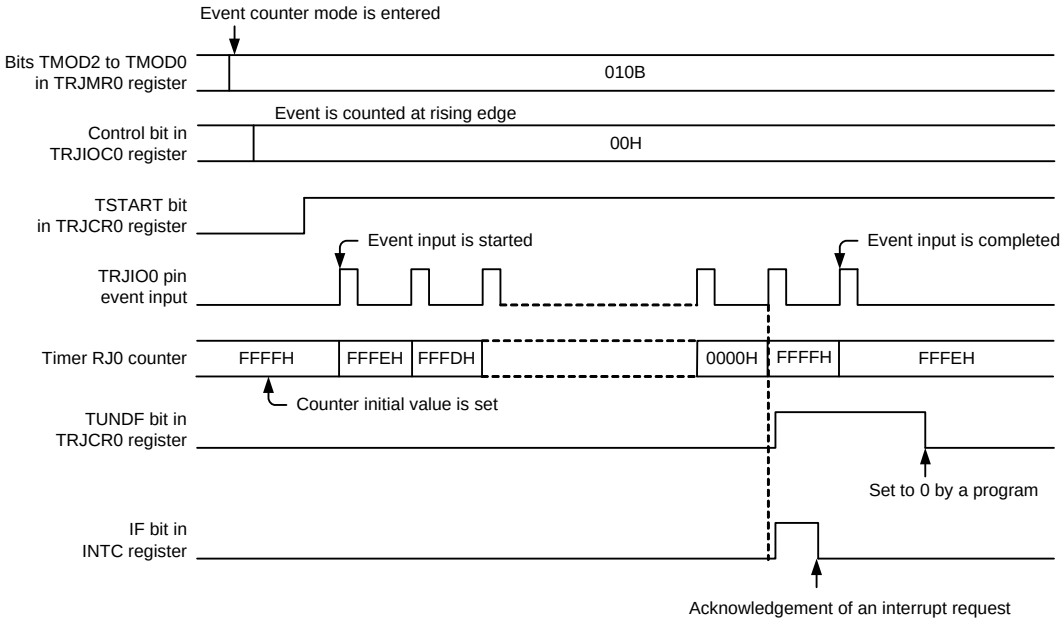
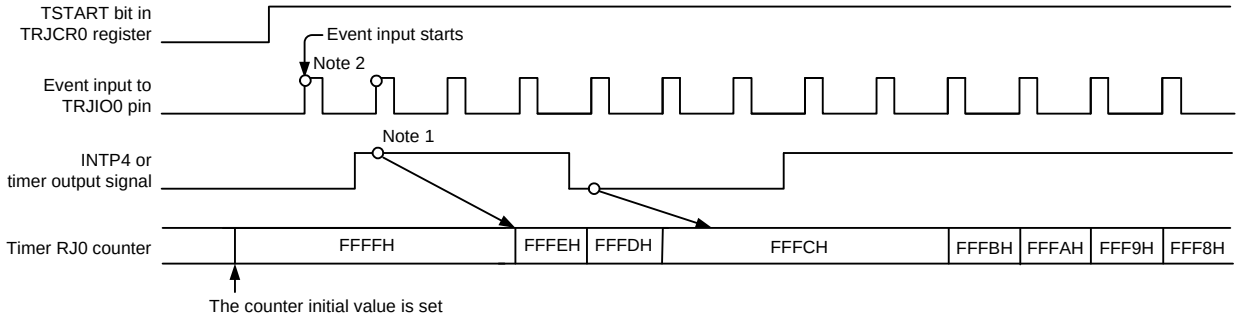



Figure 7-15 shows an operation example for counting during the specified period in event counter mode (bits TIOGT1 and TIOGT0 in the TRJIO0 register are set to 01B or 10B).

**Figure 7-15. Operation Example 2 in Event Counter Mode**

Timing example when the setting of operating mode is as follows:
TRJMR0 register: TMOD2, 1, 0 = 010B (event counter mode)
TRJIOC0 register: TIOGT1, 0 = 01B (event is counted during specified period for external interrupt pin)
TIPF1, 0 = 00B (no filter)
TEDGSEL = 0 (count at rising edge)
TRJISR0 register: RCCPSEL2 = 1 (high-level period is counted)




The following notes apply only when bits TIOGT1 and TIOGT0 in the TRJIOC0 register are 01B or 10B for the setting of operating mode in event count mode.

**Notes** 1. To control synchronization, there is a delay of two cycles of the count source until count operation is affected.
2. Count operation may be performed for two cycles of the count source immediately after the count is started, depending on the previous state before the count is stopped.
To disable the count for two cycles immediately after the count is started, write 1 to the TSTOP bit in the TRJCR0 register to initialize the internal circuit, and then make operation settings before starting count operation.

### 7.4.5 Pulse Width Measurement Mode

In this mode, the pulse width of an external signal input to the TRJIO0 pin is measured.

When the level specified by the TEDGSEL bit in the TRJIOC0 register is input to the TRJIO0 pin, the decrement is started with the selected count source. When the specified level on the TRJIO0 pin ends, the counter is stopped, the TEDGF bit in the TRJCR0 register is set to 1 (active edge received), and an interrupt request is generated. The measurement of pulse width data is performed by reading the count value while the counter is stopped. Also, when the counter underflows during measurement, the TUNDF bit in the TRJCR0 register is set to 1 (underflow) and an interrupt request is generated.

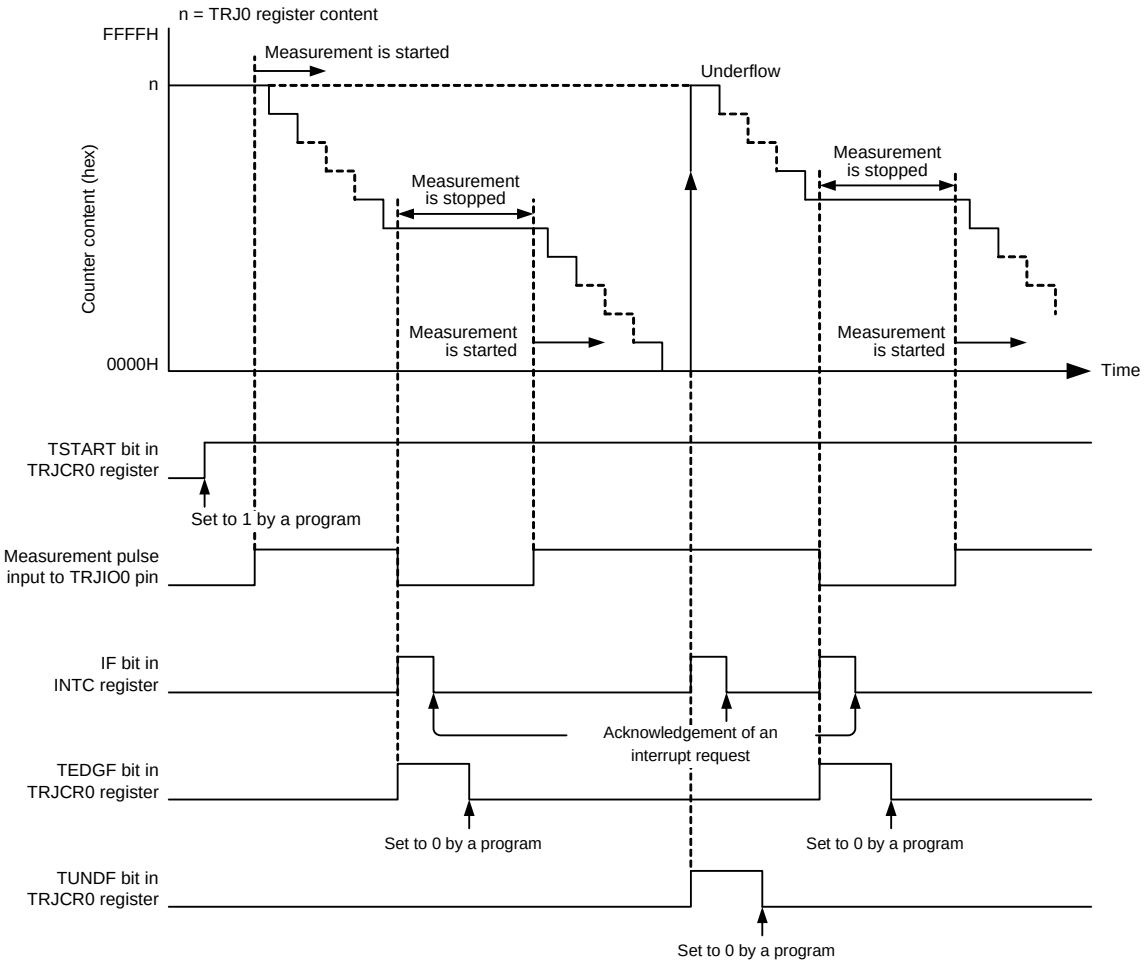
Figure 7-16 shows the Operation Example in Pulse Width Measurement Mode.

When accessing bits TEDGF and TUNDF in the TRJCR0 register, see **7. 5. 2 Access to Flags (Bits TEDGF and TUNDF in TRJCR0 Register)**.

**Caution The 20-pin products do not have TRJIO0 pin. Thus, the pulse width measurement mode cannot be used.**

**Figure 7-16. Operation Example in Pulse Width Measurement Mode**

This example applies when the high-level width of the measurement pulse is measured (TEDGSEL bit in TRJIOC0 register = 1)



### 7.4.6 Pulse Period Measurement Mode

In this mode, the pulse period of an external signal input to the TRJIO0 pin is measured.

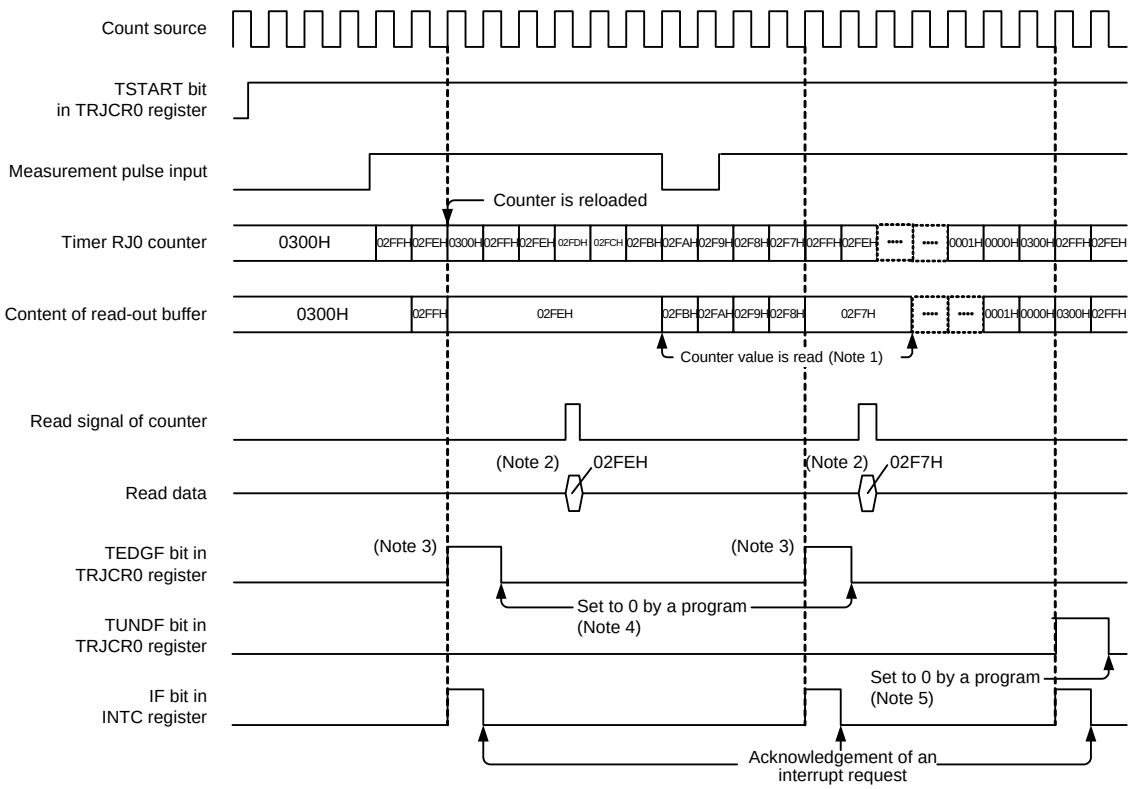
The counter is decremented by the count source selected by bits TCK0 to TCK2 in the TRJMR0 register. When a pulse with the period specified by the TEDGSEL bit in the TRJIOC0 register is input to the TRJIO0 pin, the count value is transferred to the read-out buffer at the rising edge of the count source. The value in the reload register is loaded to the counter at the next rising edge. Simultaneously, the TEDGF bit in the TRJCR0 register is set to 1 (active edge received) and an interrupt request is generated. The read-out buffer (TRJ0 register) is read at this time and the difference from the reload value is the period data of the input pulse. The period data is retained until the read-out buffer is read. When the counter underflows, the TUNDF bit in the TRJCR0 register is set to 1 (underflow) and an interrupt request is generated.

Figure 7-17 shows the Operation Example in Pulse Period Measurement Mode.

Only input pulses with a period longer than twice the period of the count source. Also, the low-level and high-level widths must be both longer than the period of the count source. If a pulse period shorter than these conditions is input, the input may be ignored.

**Caution The 20-pin products do not have TRJIO0 pin. Thus, the pulse period measurement mode cannot be used.**

**Figure 7-17. Operation Example in Pulse Period Measurement Mode**




This example applies when the initial value of the TRJ0 register is set to 0300H, the TEDGSEL bit in the TRJIOC0 register is set to 0, and the period from one rising edge to the next edge of the measurement pulse is measured.

**Notes:**

1. Reading from the TRJ0 register must be performed during the period from when the TEDGF bit is set to 1 (active edge received) until the next active edge is input. The content of the read-out buffer is retained until the TRJ0 register is read. If it is not read before the active edge is input, the measurement result of the previous period is retained.
2. When the TRJ0 register is read in pulse period measurement mode, the content of the read-out buffer is read.
3. When the active edge of the measurement pulse is input and then the set edge of an external pulse is input, the TEDGF bit in the TRJCR0 register is set to 1 (active edge received).
4. To set to 0 by a program, write 0 to the TEDGF bit in the TRJCR0 register using an 8-bit memory manipulation instruction.
5. To set to 0 by a program, write 0 to the TUNDF bit in the TRJCR0 register using an 8-bit memory manipulation instruction.

### 7.4.7 Coordination with Event Link Controller (ELC)

The ELC is only available in the RL78/F14.

Through coordination with the ELC, event input from the ELC can be set to be the count source. Bits TCK0 to TCK2 in the TRJMR0 register count at the rising edge of event input from the ELC. However, ELC input does not function in event counter mode.

The ELC setting procedure is shown below:

- Procedure for starting operation
  - (1) Set the event output destination select register (ELSELRn) for the event link controller (ELC).
  - (2) Set the operating mode for the event generation source.
  - (3) Set the mode for timer RJ.
  - (4) Start the count operation of timer RJ.
  - (5) Start the operation of the event generation source.
- Procedure for stopping operation
  - (1) Stop the operation of the event generation source.
  - (2) Stop the count operation of timer RJ.
  - (3) Set the event output destination select register (ELSELRn) for the event link controller (ELC) to 0.

### 7.4.8 Output Settings for Each Mode

Table 7-6 and Table 7-7 list the states of pins TRJO0 and TRJIO0 in each mode.

**Table 7-6. TRJO0 Pin Setting**

| Operating Mode | TRJIOC0 Register | | TRJO0 Pin Output |
|---|---|---|---|
| | TOENA Bit | TEDGSEL Bit | |
| All modes | 1 | 1 | Inverted output |
| | | 0 | Normal output |
| | 0 | 0 or 1 | Output disabled |

**Table 7-7. TRJIO0 Pin Setting**

| Operating Mode | TRJIOC0 Register | | TRJIO0 Pin I/O |
|---|---|---|---|
| | PMXX Bit [Note] | TEDGSEL Bit | |
| Timer mode | 0 or 1 | 0 or 1 | Input (Not used) |
| Pulse output mode | 1 | 0 or 1 | Output disabled (Hi-z output) |
| | 0 | 1 | Normal output |
| | | 0 | Inverted output |
| Event counter mode | 1 | 0 or 1 | Input |
| Pulse width measurement mode | | | |
| Pulse period measurement mode | | | |

**Note** The port mode register (PMxx) bit corresponding to port multiplexed with TRJIO0 function.

## 7.5 Notes on Timer RJ

### 7.5.1 Count Operation Start and Stop Control

- When event counter mode is set or the count source is set to other than the ELC

After 1 (count starts) is written to the TSTART bit in the TRJCR0 register while the count is stopped, the TCSTF bit in the TRJCR0 register remains 0 (count stops) for three cycles of the count source. Do not access the registers associated with timer RJ [Note] other than the TCSTF bit until this bit is set to 1 (count in progress).

After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF bit remains 1 for three cycles of the count source. When the TCSTF bit is set to 0, the count is stopped. Do not access the registers associated with timer RJ [Note] other than the TCSTF bit until this bit is set to 0.

Clear the interrupt register before changing the TSTART bit from 0 to 1. Refer to **CHAPTER 21 INTERRUPT FUNCTIONS** for details.

**Note** Registers associated with timer RJ: TRJ0, TRJCR0, TRJIOC0, TRJMR0, and TRJISR0

- When event counter mode is set or the count source is set to the ELC

After 1 (count starts) is written to the TSTART bit in the TRJCR0 register while the count is stopped, the TCSTF bit in the TRJCR0 register remains 0 (count stops) for two cycles of the CPU clock. Do not access the registers associated with timer RJ [Note] other than the TCSTF bit until this bit is set to 1 (count in progress).

After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF bit remains 1 for two cycles of the CPU clock. When the TCSTF bit is set to 0, the count is stopped. Do not access the registers associated with timer RJ [Note] other than the TCSTF bit until this bit is set to 0.

Clear the interrupt register before changing the TSTART bit from 0 to 1. Refer to **CHAPTER 21 INTERRUPT FUNCTIONS** for details.

The ELC is only available in the RL78/F14.

**Note** Registers associated with timer RJ: TRJ0, TRJCR0, TRJIOC0, TRJMR0, and TRJISR0

### 7.5.2 Access to Flags (Bits TEDGF and TUNDF in TRJCR0 Register)

Bits TEDGF and TUNDF in the TRJCR0 register are set to 0 by writing 0 by a program, but writing 1 to these bits has no effect. If a read-modify-write instruction is used to set the TRJCR0 register, bits TEDGF and TUNDF may be erroneously set to 0 depending on the timing, even when the TEDGF bit is set to 1 (active edge received) and the TUNDF bit is set to 1 (underflow) during execution of the instruction. Use an 8-bit memory manipulation instruction to access to the TRJCR0 register.

### 7.5.3 Access to Counter Register

When bits TSTART and TCSTF in the TRJCR0 register are both 1 (count starts), allow at least three cycles of the count source clock between writes when writing to the TRJ0 register successively.

### 7.5.4 When Changing Mode

The registers associated with timer RJ operating mode (TRJIOC0, TRJMR0, and TRJISR0) can be changed only when the count is stopped with both the TSTART and TCSTF bits set to 0 (count stops). Do not change these registers during count operation.

When the registers associated with timer RJ operating mode are changed, the values of bits TSTART and TCSTF are undefined. Write 0 (no active edge received) to the TEDGF bit and 0 (no underflow) to the TUNDF bit before starting the count.

### 7.5.5 Procedure for Setting Pins TRJO0 and TRJIO0

After a reset, the I/O ports multiplexed with pins TRJO0 and TRJIO0 function as input ports.

To output from pins TRJO0 and TRJIO0, use the following setting procedure.

The 20-pin products do not have TRJO0 and TRJIO0 pins.

Changing procedure

(1) Set the mode.

(2) Set the initial value/output enabled.

(3) Set the port register bits corresponding to pins TRJO0 and TRJIO0 to 0.

(4) Set the port mode register bits corresponding to pins TRJO0 and TRJIO0 to output mode.
(Output is started from pins TRJO0 and TRJIO0)

(5) Start the count (TSTART in TRJCR0 register = 1).

To input from the TRJIO0 pin, use the following setting procedure:

(1) Set the mode.

(2) Set the initial value/edge selected.

(3) Set the port mode register bit corresponding to TRJIO0 pin to input mode.
(Input is started from the TRJIO0 pin)

(4) Start the count (TSTART in TRJCR0 register = 1).

(5) Wait until the TCSTF bit in the TRJCR0 register is set to 1 (count in progress).
(In event counter mode only)

(6) Input an external event from the TRJIO0 pin.

(7) The processing on completion of the first measurement is invalid (the measured value is valid for the second and subsequent times). (In pulse width measurement mode and pulse period measurement mode only)

### 7.5.6 When Timer RJ is not Used

When timer RJ is not used, set bits TMOD2 to TMOD0 in the TRJMR0 register to 000B (timer mode) and set the TOENA bit in the TRJIOC0 register to 0 (TRJO output disabled).

### 7.5.7 When Timer RJ Operating Clock is Stopped

Supplying or stopping the timer RJ clock can be controlled by the TRJ0EN bit in the PER1 register. Note that the following SFRs cannot be accessed while the timer RJ clock is stopped. Make sure the timer RJ clock is supplied before accessing any of these registers.

Registers TRJ0, TRJCR0, TRJMR0, TRJIOC0, and TRJISR0.

### 7.5.8 Procedure for Setting STOP Mode (Event Counter Mode)

To perform event counter mode operation during STOP mode, first supply the timer RJ clock and then use the following procedure to enter STOP mode.

Setting procedure

(1) Set the operating mode.

(2) Start the count (TSTART = 1, TCSTF = 1).

(3) Stop supplying the timer RJ clock.

To stop event counter mode operation during STOP mode, use the following procedure to stop operation.

(1) Supply the timer RJ clock.

(2) Stop the count (TSTART = 0, TCSTF = 0)

### 7.5.9 Functional Restriction in STOP Mode (Event Counter Mode Only)

When event counter mode operation is performed during STOP mode, the digital filter function cannot be used.

### 7.5.10 When Count is Forcibly Stopped by TSTOP Bit

After the counter is forcibly stopped by the TSTOP bit in the TRJCR0 register, do not access the following SFRs for one cycle of the count source.

Registers TRJ0, TRJCR0, and TRJMR0

### 7.5.11 Digital Filter

When the digital filter is used, do not start timer operation for five cycles of the digital filter clock after setting bits TIPF1 and TIPF0.

Also, do not start timer operation for five cycles of the digital filter clock when the TEDGSEL bit in the TRJIOC register is changed while the digital filter is used.

### 7.5.12 When Selecting $f_{IL}$ as Count Source

When selecting $f_{IL}$ as the count source, set the WUTMMCK0 bit in the operation speed mode control register (OSMC) to 1. However, $f_{SUB}$ cannot be selected as the count source for timer RJ when $f_{SL}$ ($f_{IL}$) is selected as the count source for timer RD or the output clock for clock output/buzzer output.

# CHAPTER 8 TIMER RD

Timer RD contains two 16-bit timer units (timer RD0 and timer RD1).

## 8.1 Overview

Each of timer RD0 and timer RD1 has four I/O pins.
The timer RD operating clock ($f_{TRD}$) is selectable from $f_{CLK}$, $f_{MP}$, or $f_{SL}$.

Figure 8-1 shows the Timer RD Block Diagram and Table 8-1 lists the Timer RD Pin Configuration.
Timer RD has four modes:

- Timer mode
  - Input capture function — Transfer the counter value to a register with an external signal as the trigger
  - Output compare function — Detect register value matches with a counter (Pin output can be changed at detection)
  - PWM function — Output pulse of any width continuously

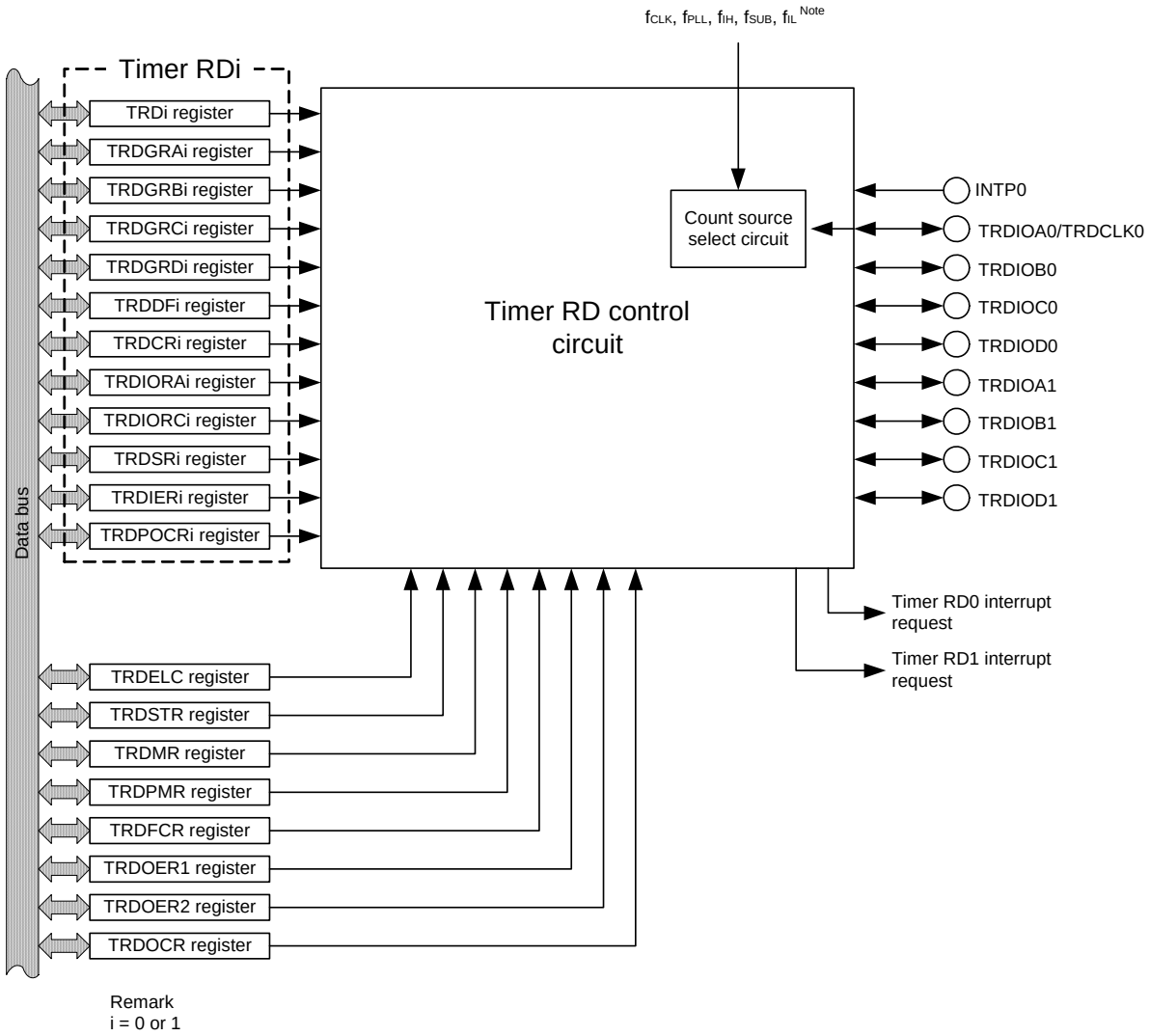
The following three modes use the PWM function.

- Reset synchronous PWM mode — Output three-phase waveforms (6) without sawtooth wave modulation and dead time
- Complementary PWM mode — Output three-phase waveforms (6) with triangular wave modulation and dead time
- PWM3 mode — Output PWM waveforms (2) with a fixed period

The timer mode input capture function, output compare function, and PWM function are equivalent in timer RD0 and timer RD1, and these functions can be selected individually for each pin. Also, a combination of these functions can be used in timer RD0 and timer RD1.

In reset synchronous PWM mode, complementary PWM mode, and PWM3 mode, a waveform is output with a combination of counters and registers in timer RD0 and timer RD1. Pin functions depend on the mode.

**Figure 8-1. Timer RD Block Diagram**




**Note** $f_{IH}$ can be selected when it is 64 MHz or 48 MHz. $f_{PLL}$ can be selected when it is over 32 MHz.

**Remark** i = 0 or 1

**Table 8-1. Timer RD Pin Configuration**

| Pin Name | Assigned Pin | I/O | Function |
|---|---|---|---|
| TRDIOA0/TRDCLK0 | P13 (P15) | Input/Output | Function varies depending on the mode.<br>Refer to descriptions of individual modes for details. |
| TRDIOB0 | P125 (P11) | Input/Output | |
| TRDIOC0 | P14 | Input/Output | |
| TRDIOD0 | P120 (P12) | Input/Output | |
| TRDIOA1 | P15 | Input/Output | |
| TRDIOB1 | P17 | Input/Output | |
| TRDIOC1 | P16 | Input/Output | |
| TRDIOD1 | P30 | Input/Output | |

## 8.2 Registers

Table 8-2 lists the Timer RD Register Configuration.

**Table 8-2. Timer RD Register Configuration**

| Register Name | Symbol | After Reset | Address | Access Size |
|---|---|---|---|---|
| Peripheral Enable Register 1 | PER1 | 00H | F02C0H | 1, 8 |
| Clock Select Register | CKSEL | 00H | F02C4H | 1, 8 |
| Timer RD ELC Register | TRDELC | 00H Note | F0260H | 1, 8 |
| Timer RD Start Register | TRDSTR | 0CH Note | F0263H | 8 |
| Timer RD Mode Register | TRDMR | 00H Note | F0264H | 1, 8 |
| Timer RD PWM Function Select Register | TRDPMR | 00H Note | F0265H | 1, 8 |
| Timer RD Function Control Register | TRDFCR | 80H Note | F0266H | 1, 8 |
| Timer RD Output Master Enable Register 1 | TRDOER1 | FFH Note | F0267H | 1, 8 |
| Timer RD Output Master Enable Register 2 | TRDOER2 | 00H Note | F0268H | 1, 8 |
| Timer RD Output Control Register | TRDOCR | 00H Note | F0269H | 1, 8 |
| Timer RD Digital Filter Function Select Register 0 | TRDDF0 | 00H Note | F026AH | 1, 8 |
| Timer RD Digital Filter Function Select Register 1 | TRDDF1 | 00H Note | F026BH | 1, 8 |
| Timer RD Control Register 0 | TRDCR0 | 00H Note | F0270H | 1, 8 |
| Timer RD I/O Control Register A0 | TRDIORA0 | 00H Note | F0271H | 1, 8 |
| Timer RD I/O Control Register C0 | TRDIORC0 | 88H Note | F0272H | 1, 8 |
| Timer RD Status Register 0 | TRDSR0 | 00H Note | F0273H | 1, 8 |
| Timer RD Interrupt Enable Register 0 | TRDIER0 | 00H Note | F0274H | 1, 8 |
| Timer RD PWM Function Output Level Control Register 0 | TRDPOCR0 | 00H Note | F0275H | 1, 8 |
| Timer RD Counter 0 | TRD0 | 0000H Note | F0276H | 16 |
| Timer RD General Register A0 | TRDGRA0 | FFFFH Note | F0278H | 16 |
| Timer RD General Register B0 | TRDGRB0 | FFFFH Note | F027AH | 16 |
| Timer RD General Register C0 | TRDGRC0 | FFFFH Note | FFF58H | 16 |
| Timer RD General Register D0 | TRDGRD0 | FFFFH Note | FFF5AH | 16 |
| <R> Timer RD Control Register 1 | TRDCR1 | 00H Note | F0280H | 1, 8 |
| Timer RD I/O Control Register A1 | TRDIORA1 | 00H Note | F0281H | 1, 8 |
| Timer RD I/O Control Register C1 | TRDIORC1 | 88H Note | F0282H | 1, 8 |
| Timer RD Status Register 1 | TRDSR1 | 00H Note | F0283H | 1, 8 |
| Timer RD Interrupt Enable Register 1 | TRDIER1 | 00H Note | F0284H | 1, 8 |
| Timer RD PWM Function Output Level Control Register 1 | TRDPOCR1 | 00H Note | F0285H | 1, 8 |
| Timer RD Counter 1 | TRD1 | 0000H Note | F0286H | 16 |
| Timer RD General Register A1 | TRDGRA1 | FFFFH Note | F0288H | 16 |
| Timer RD General Register B1 | TRDGRB1 | FFFFH Note | F028AH | 16 |
| Timer RD General Register C1 | TRDGRC1 | FFFFH Note | FFF5CH | 16 |
| Timer RD General Register D1 | TRDGRD1 | FFFFH Note | FFF5EH | 16 |
| PWM output delay control register 0 | PWMDLY0 | 0000H | F0228H | 16 |
| Port Register 1 | P1 | 00H | FFF01H | 1, 8 |
| Port Mode Register 1 | PM1 | FFH | FFF21H | 1, 8 |
| Port Register 3 | P3 | 00H | FFF03H | 1, 8 |
| Port Mode Register 3 | PM3 | FFH | FFF23H | 1, 8 |
| Port Register 12 | P12 | - | FFF0CH | 1, 8 |
| Port Mode Register 12 | PM12 | FFH | FFF2CH | 1, 8 |
| PLL Control Register | PLLCTL | 00H | F02C5H | 1, 8 |
| PLL Status Register | PLLSTS | 00H | F02C6H | 1, 8 |
| $f_{MP}$ Clock Division Register | MDIV | 00H/01H | F02C7H | 8 |
| System Clock Control Register | CKC | 00H | FFFA4H | 1, 8 |

**Note** The timer RD SFRs are undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

### 8.2.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use timer RD, be sure to set bit 4 (TRD0EN) to 1.

Set the PER1 register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 8-2. Format of Peripheral Enable Register 1 (PER1)**

Address: F02C0H After reset: 00H R/W

| Symbol | <7> | 6 | <5> | <4> | <3> | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PER1 | DACEN[Note] | 0 | CMPEN[Note] | TRD0EN | DTCEN | 0 | 0 | TRJ0EN |

| TRD0EN | Control of timer RD input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by timer RD cannot be written.<br>• Timer RD is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by timer RD can be read and written. |

**Note** Only for RL78/F14.

**Cautions 1. When setting timer RD, be sure to set the TRD0EN bit to 1 first. If TRD0EN = 0, writing to a control register of timer RD is ignored, and all read values are default values (except for port mode registers 1, 3, and 12 (PM1, PM3, and PM12) and port registers 1, 3, and 12 (P1, P3, and P12).**

**2. Be sure to clear the following bits to 0.**
**RL78/F13: bits 1, 2, and 5 to 7**
**RL78/F14: bits 1, 2, and 6**

**3. When selecting $f_{IF}$ (64 MHz or 48 MHz) as the count source, set $f_{CLK}$ to $f_{IH}$. When selecting $f_{PLL}$ (over 32 MHz) as the count source, set $f_{CLK}$ to $f_{PLL}$. When selecting $f_{SUB}$ or $f_{IL}$ as the count source to access the timer RD-related registers, set $f_{CLK}$ to $f_{SUB}$ or $f_{IL}$, respectively.**

### 8.2.2 Clock Select Register (CKSEL)

This register is used to select the CPU clock ($f_{SUB}$/$f_{IL}$) and the clocks for the timer RJ, timer RD, and clock output/buzzer output. Together with the CMC register, the SELLOSC bit is used to set the operation mode of the subsystem clock. For details, see **Figure 5-3 Format of Clock Operation Mode Control Register (CMC)**.

Set the CKSEL register by a 1-bit or 8-bit memory manipulation instruction.

Writing to the CKSEL register is disabled when the GCSC bit of the IAWCTL register is set to 1.

**Figure 8-3. Format of Clock Select Register (CKSEL)**

Address: F02C4H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| CKSEL | 0 | 0 | 0 | 0 | 0 | TRD_ CKSEL | 0 | SELLOSC Notes 5, 6 |

| TRD_ CKSEL | Control of TRD clock selection |
|---|---|
| 0 | Selects $f_{CLK}$ or $f_{MP}$ Note 1. |
| 1 | Selects $f_{SL}$ Note 2. |

| SELLOSC Notes 5, 6 | Control of sub/low-speed on-chip oscillator selection clock ($f_{SL}$) selection |
|---|---|
| 0 | Selects $f_{SUB}$ Note 3 |
| 1 | Selects $f_{IL}$ Note 4 |

**Notes 1.** When FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and PLLDIV1 = 1 ($f_{PLL}$ > 32 MHz) in the PLLCTL register, set the TRD_CKSEL bit to 0.
When FRQSEL4 = 1 in the user option byte (000C2H/020C2H) or PLLDIV1 = 1 ($f_{PLL}$ > 32 MHz) in the PLLCTL register, the timer RD operating clock ($f_{TRD}$) becomes $f_{MP}$.

**2.** When $f_{SL}$ is selected as the timer RD operating clock ($f_{TRD}$), $f_{SL}$ should be selected as the CPU clock
(set the CSS bit in the CKC register to 1).

**3.** When setting $f_{SUB}$ as the CPU/peripheral hardware clock, first set the SELLOSC bit to 0 and then set the CSS bit in the CKC register to 1.

**4.** When setting $f_{IL}$ as the CPU/peripheral hardware clock, first set the SELLOSC bit to 1 and then set the CSS bit in the CKC register to 1.

**5.** When the SELLOSC bit is set to 1, the low-speed on-chip oscillator operates.

**6.** When setting the CKSEL register in the 20-, 30-, or 32-pin products, set the SELLOSC bit to 1.

### 8.2.3 Timer RD ELC Register (TRDELC)

**Figure 8-4. Format of Timer RD ELC Register (TRDELC)**

Address: F0260H After Reset: 00H

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDELC | — | — | ELCOBE1 | ELCICE1 | — | — | ELCOBE0 | ELCICE0 |

| Bits 7 to 6 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| ELCOBE1 | ELC event input 1 enable for timer RD pulse output forced cutoff | R/W |
|---|---|---|
| 0 | Forced cutoff is disabled | R/W |
| 1 | Forced cutoff is enabled | |

| ELCICE1 | ELC event input 1 select for timer RD input capture D1 | R/W |
|---|---|---|
| 0 | Input capture D1 is selected | R/W |
| 1 | Event input 1 from the event link controller (ELC) is selected | |

| Bits 3 to 2 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| ELCOBE0 | ELC event input 0 enable for timer RD pulse output forced cutoff | R/W |
|---|---|---|
| 0 | Forced cutoff is disabled | R/W |
| 1 | Forced cutoff is enabled | |

| ELCICE0 | ELC event input 0 select for timer RD input capture D0 | R/W |
|---|---|---|
| 0 | Input capture D0 is selected | R/W |
| 1 | Event input 0 from the event link controller (ELC) is selected | |

**Caution The timer RD ELC register (TRDELC) is only available in the RL78/F14. Do not access the register in other products.**

### 8.2.4 Timer RD Start Register (TRDSTR)

Set the TRDSTR register by an 8-bit memory manipulation instruction. See **8. 5. 1 (1) TRDSTR Register in 8.5 Notes on Timer RD**.

**Figure 8-5. Format of Timer RD Start Register (TRDSTR)**

Address: F0263H After Reset: 0CH [Note 1]

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDSTR | — | — | — | — | CSEL1 | CSEL0 | TSTART1 | TSTART0 |

| Bits 7 to 4 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| CSEL1 | TRD1 count operation select [Note 2] | R/W |
|---|---|---|
| 0 | Count stops at compare match with TRDGRA1 register | R/W |
| 1 | Count continues after compare match with TRDGRA1 register [Note 3] | |

| CSEL0 | TRD0 count operation select | R/W |
|---|---|---|
| 0 | Count stops at compare match with TRDGRA0 register | R/W |
| 1 | Count continues after compare match with TRDGRA0 register [Note 3] | |

| TSTART1 | TRD1 count start flag [Notes 4, 5] | R/W |
|---|---|---|
| 0 | Count stops | R/W |
| 1 | Count starts | |

| TSTART0 | TRD0 count start flag [Notes 6, 7] | R/W |
|---|---|---|
| 0 | Count stops | R/W |
| 1 | Count starts | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** Do not use in PWM3 mode.

**3.** Set to 1 for the input capture function.

**4.** Write 0 to the TSTART1 bit while the CSEL1 bit is set to 1.

**5.** When the CSEL1 bit is 0 and a compare match signal (TRDIOA1) is generated, this flag is set to 0 (count stops).

**6.** Write 0 to the TSTART0 bit while the CSEL0 bit is set to 1.

**7.** When the CSEL0 bit is 0 and a compare match signal (TRDIOA0) is generated, this flag is set to 0 (count stops).

### 8.2.5 Timer RD Mode Register (TRDMR)

**Figure 8-6. Format of Timer RD Mode Register (TRDMR)**

Address: F0264H After Reset:00H [Note 1]

| Symbol | <7> | <6> | <5> | <4> | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| TRDMR | TRDBFD1 | TRDBFC1 | TRDBFD0 | TRDBFC0 | 0 | 0 | 0 | TRDSYNC |

| TRDBFD1 | TRDGRD1 register function select [Note 2] | R/W |
|---|---|---|
| 0 | General register | R/W |
| 1 | Buffer register for TRDGRB1 register | |

| TRDBFC1 | TRDGRC1 register function select [Note 2] | R/W |
|---|---|---|
| 0 | General register | R/W |
| 1 | Buffer register for TRDGRA1 register | |

| TRDBFD0 | TRDGRD0 register function select [Note 2] | R/W |
|---|---|---|
| 0 | General register | R/W |
| 1 | Buffer register for TRDGRB0 register | |

| TRDBFC0 | TRDGRC0 register function select [Notes 2, 3] | R/W |
|---|---|---|
| 0 | General register | R/W |
| 1 | Buffer register for TRDGRA0 register | |

| Bits 3 to1 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| TRDSYNC | Timer RD synchronous [Note 4] | R/W |
|---|---|---|
| 0 | TRD0 and TRD1 operate independently | R/W |
| 1 | TRD0 and TRD1 operate synchronously | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** In the output compare function, if 0 (TRDGRji register output pin is changed) is selected for the IOj3 (j = C or D) bit in the TRDIORCi (i = 0 or 1) register, set the TRDBFji bit in the TRDMR register to 0.

**3.** Set to 0 (general register) in complementary PWM mode.

**4.** Set to 0 (TRD0 and TRD1 operate independently) in reset synchronous PWM mode, complementary PWM mode, and PWM3 mode.

### 8.2.6 Timer RD PWM Function Select Register (TRDPMR)

**Figure 8-7. Format of Timer RD PWM Function Select Register (TRDPMR) [Timer Mode]**

Address: F0265H After Reset:00H [Note]

| Symbol | 7 | <6> | <5> | <4> | 3 | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| TRDPMR | 0 | TRDPWMD1 | TRDPWMC1 | TRDPWMB1 | 0 | TRDPWMD0 | TRDPWMC0 | TRDPWMB0 |

| Bit 7 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| TRDPWMD1 | PWM function of TRDIOD1 select | R/W |
|---|---|---|
| 0 | Input capture function or output compare function | R/W |
| 1 | PWM function | |

| TRDPWMC1 | PWM function of TRDIOC1 select | R/W |
|---|---|---|
| 0 | Input capture function or output compare function | R/W |
| 1 | PWM function | |

| TRDPWMB1 | PWM function of TRDIOB1 select | R/W |
|---|---|---|
| 0 | Input capture function or output compare function | R/W |
| 1 | PWM function | |

| Bit 3 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| TRDPWMD0 | PWM function of TRDIOD0 select | R/W |
|---|---|---|
| 0 | Input capture function or output compare function | R/W |
| 1 | PWM function | |

| TRDPWMC0 | PWM function of TRDIOC0 select | R/W |
|---|---|---|
| 0 | Input capture function or output compare function | R/W |
| 1 | PWM function | |

| TRDPWMB0 | PWM function of TRDIOB0 select | R/W |
|---|---|---|
| 0 | Input capture function or output compare function | R/W |
| 1 | PWM function | |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

### 8.2.7 Timer RD Function Control Register (TRDFCR)

**Figure 8-8. Format of Timer RD Function Control Register (TRDFCR)**

Address: F0266H After Reset: 80H Note 1

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDFCR | PWM3 | STCLK | 0 | 0 | OLS1 | OLS0 | CMD1 | CMD0 |

| PWM3 | PWM3 mode select Note 2 | R/W |
|---|---|---|
| • In the timer mode, set to 1 (other than PWM3 mode).<br>• In PWM3 mode, set to 0 (PWM3 mode).<br>• Disabled in reset synchronous and complementary PWM modes. | | R/W |

| STCLK | External clock input select | R/W |
|---|---|---|
| • In the timer mode, the reset synchronous PWM mode, and the complementary PWM mode,<br>0: External clock input disabled<br>1: External clock input enabled<br>• In PWM3 mode, set to 0 (external clock input disabled). | | R/W |

| Bits 5 to 4 | Reserved | R/W |
|---|---|---|
| 0 | Set to 0. | R/W |

| OLS1 | Counter-phase output level select<br>(in reset synchronous PWM mode or complementary PWM mode) | R/W |
|---|---|---|
| • In reset synchronous and complementary PWM modes,<br>0: High initial output and low active level<br>1: Low initial output and high active level<br>• Disabled in timer and PWM3 modes. | | R/W |

| OLS0 | Phase output level select<br>(in reset synchronous PWM mode or complementary PWM mode) | R/W |
|---|---|---|
| • In reset synchronous and complementary PWM modes,<br>0: High initial output and low active level<br>1: Low initial output and high active level<br>• Disabled in timer and PWM3 modes. | | R/W |

| CMD1 | CMD0 | Combination mode select Notes 3, 4 | R/W |
|---|---|---|---|
| • In timer and PWM3 modes, set to 00B (timer mode or PWM3 mode).<br>• In reset synchronous PWM mode, set to 01B (reset synchronous PWM mode).<br>• In complementary PWM mode,<br>CMD1 CMD0<br>1 0: Complementary PWM mode (transfer from the buffer register to the general register when TRD1 underflows)<br>1 1: Complementary PWM mode (transfer from the buffer register to the general register at compare match between registers TRD0 and TRDGRA0)<br>Other than the above: Do not set. | | | R/W |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** When bits CMD1 and CMD0 are set to 00B (timer mode or PWM3 mode), the setting of the PWM3 bit is enabled.

**3.** Set bits CMD0 and CMD1 when both the TSTART0 and TSTART1 bits in the TRDSTR register are set to 0 (count stops).

**4.** When bits CMD1 and CMD0 are set to 01B, 10B, or 11B, the MCU enters reset synchronous PWM mode or complementary PWM mode regardless of the settings of the TRDPMR register.

### 8.2.8 Timer RD Output Master Enable Register 1 (TRDOER1)

**Figure 8-9. Format of Timer RD Output Master Enable Register 1 (TRDOER1) [Output Compare Function, PWM Function, Reset Synchronous PWM Mode, Complementary PWM Mode, and PWM3 Mode]**

Address: F0267H After Reset: FFH [Note 1]

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDOER1 | ED1 | EC1 | EB1 | EA1 | ED0 | EC0 | EB0 | EA0 |

| ED1 | TRDIOD1 output disable [Note 2] | R/W |
|---|---|---|
| 0 | Output enabled | R/W |
| 1 | Output disabled (TRDIOD1 pin functions as an I/O port.) | |

| EC1 | TRDIOC1 output disable [Note 2] | R/W |
|---|---|---|
| 0 | Output enabled | R/W |
| 1 | Output disabled (TRDIOC1 pin functions as an I/O port.) | |

| EB1 | TRDIOB1 output disable [Note 2] | R/W |
|---|---|---|
| 0 | Output enabled | R/W |
| 1 | Output disabled (TRDIOB1 pin functions as an I/O port.) | |

| EA1 | TRDIOA1 output disable [Notes 2, 3] | R/W |
|---|---|---|
| 0 | Output enabled | R/W |
| 1 | Output disabled (TRDIOA1 pin functions as an I/O port) | |

| ED0 | TRDIOD0 output disable [Note 2] | R/W |
|---|---|---|
| 0 | Output enabled | R/W |
| 1 | Output disabled (TRDIOD0 pin functions as an I/O port.) | |

| EC0 | TRDIOC0 output disable [Note 2] | R/W |
|---|---|---|
| 0 | Output enabled | R/W |
| 1 | Output disabled (TRDIOC0 pin functions as an I/O port.) | |

| EB0 | TRDIOB0 output disable | R/W |
|---|---|---|
| 0 | Output enabled | R/W |
| 1 | Output disabled (TRDIOB0 pin functions as an I/O port.) | |

| EA0 | TRDIOA0 output disable [Notes 3, 4] | R/W |
|---|---|---|
| 0 | Output enabled | R/W |
| 1 | Output disabled (TRDIOA0 pin functions as an I/O port) | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** Set to 1 in PWM3 mode.

**3.** Set to 1 in PWM function.

**4.** Set to 1 in reset synchronous PWM mode and complementary PWM mode.

### 8.2.9 Timer RD Output Master Enable Register 2 (TRDOER2)

**Figure 8-10. Format of Timer RD Output Master Enable Register 2 (TRDOER2)**
**[PWM Function, Reset Synchronous PWM Mode, Complementary PWM Mode, and PWM3 Mode]**

Address: F0268H After Reset: 00H [Note 1]

| Symbol | <7> | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| TRDOER2 | TRDPTO | 0 | 0 | 0 | 0 | 0 | 0 | TRDSHUTS |

| TRDPTO | INTP0 of pulse output forced cutoff signal input enabled [Note 2] | R/W |
|---|---|---|
| 0 | Pulse output forced cutoff input disabled | R/W |
| 1 | Pulse output forced cutoff input enabled<br>(The TRDSHUTS bit is set to 1 when a low level is applied to the INTP0 pin.) | |

| Bits 6 to 1 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| TRDSHUTS | Forced cutoff flag | R/W |
|---|---|---|
| 0 | Not forcibly cut off | R/W |
| 1 | Forcibly cut off | |
| This bit is set to 1 when the pulse is forcibly cut off by an INTP0 or ELC [Note 3] input event. This bit is not automatically cleared. To stop the forced cutoff of the pulse, write 0 to this bit while the count is stopped (TSTARTi = 0). The pulse is also forcibly cut off when 1 is written to the TRDSHUTS bit in an enabled mode. | | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** See **8.3.1 (4) Pulse Output Forced Cutoff**.

**3.** The ELC is only available in the RL78/F14.

### 8.2.10 Timer RD Output Control Register (TRDOCR)

Write to the TRDOCR register when bits TSTART0 and TSTART1 in the TRDSTR register are both 0 (count stops).

**Figure 8-11. Format of Timer RD Output Control Register (TRDOCR) [Output Compare Function]**

Address: F0269H After Reset: 00H Note 1

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDOCR | TOD1 | TOC1 | TOB1 | TOA1 | TOD0 | TOC0 | TOB0 | TOA0 |

| TOD1 | TRDIOD1 initial output level select Note 2 | R/W |
|---|---|---|
| 0 | Low initial output | R/W |
| 1 | High initial output | |

| TOC1 | TRDIOC1 initial output level select Note 2 | R/W |
|---|---|---|
| 0 | Low initial output | R/W |
| 1 | High initial output | |

| TOB1 | TRDIOB1 initial output level select Note 2 | R/W |
|---|---|---|
| 0 | Low initial output | R/W |
| 1 | High initial output | |

| TOA1 | TRDIOA1 initial output level select | R/W |
|---|---|---|
| 0 | Low initial output | R/W |
| 1 | High initial output | |

| TOD0 | TRDIOD0 initial output level select Note 2 | R/W |
|---|---|---|
| 0 | Low initial output | R/W |
| 1 | High initial output | |

| TOC0 | TRDIOC0 initial output level select Note 2 | R/W |
|---|---|---|
| 0 | Low initial output | R/W |
| 1 | High initial output | |

| TOB0 | TRDIOB0 initial output level select Note 2 | R/W |
|---|---|---|
| 0 | Low initial output | R/W |
| 1 | High initial output | |

| TOA0 | TRDIOA0 initial output level select | R/W |
|---|---|---|
| 0 | Low initial output | R/W |
| 1 | High initial output | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** If the pin function is set for waveform output, the initial output level is output when the TRDOCR register is set.

**Figure 8-12. Format of Timer RD Output Control Register (TRDOCR) [PWM Function]**

Address: F0269H After Reset: 00H Note 1

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDOCR | TOD1 | TOC1 | TOB1 | TOA1 | TOD0 | TOC0 | TOB0 | TOA0 |

| TOD1 | TRDIOD1 initial output level select Note 2 | R/W |
|---|---|---|
| 0 | Initial output is not active level | R/W |
| 1 | Initial output is active level | |

| TOC1 | TRDIOC1 initial output level select Note 2 | R/W |
|---|---|---|
| 0 | Initial output is not active level | R/W |
| 1 | Initial output is active level | |

| TOB1 | TRDIOB1 initial output level select Note 2 | R/W |
|---|---|---|
| 0 | Initial output is not active level | R/W |
| 1 | Initial output is active level | |

| TOA1 | TRDIOA1 initial output level select | R/W |
|---|---|---|
| Set to 0. | | R/W |

| TOD0 | TRDIOD0 initial output level select Note 2 | R/W |
|---|---|---|
| 0 | Initial output is not active level | R/W |
| 1 | Initial output is active level | |

| TOC0 | TRDIOC0 initial output level select Note 2 | R/W |
|---|---|---|
| 0 | Initial output is not active level | R/W |
| 1 | Initial output is active level | |
| Enabled in reset synchronous and complementary PWM modes. | | |

| TOB0 | TRDIOB0 initial output level select Note 2 | R/W |
|---|---|---|
| 0 | Initial output is not active level | R/W |
| 1 | Initial output is active level | |

| TOA0 | TRDIOA0 initial output level select | R/W |
|---|---|---|
| Set to 0. | | R/W |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** If the pin function is set for waveform output, the initial output level is output when the TRDOCR register is set.

**Figure 8-13. Format of Timer RD Output Control Register (TRDOCR) [PWM3 Mode]**

Address: F0269H After Reset: 00H [Note 1]

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDOCR | TOD1 | TOC1 | TOB1 | TOA1 | TOD0 | TOC0 | TOB0 | TOA0 |

| TOD1 | TRDIOD1 initial output level select | R/W |
|---|---|---|
| Disabled in PWM3 mode. | | R/W |

| TOC1 | TRDIOC1 initial output level select | R/W |
|---|---|---|
| Disabled in PWM3 mode. | | R/W |

| TOB1 | TRDIOB1 initial output level select | R/W |
|---|---|---|
| Disabled in PWM3 mode. | | R/W |

| TOA1 | TRDIOA1 initial output level select | R/W |
|---|---|---|
| Disabled in PWM3 mode. | | R/W |

| TOD0 | TRDIOD0 initial output level select | R/W |
|---|---|---|
| Disabled in PWM3 mode. | | R/W |

| TOC0 | TRDIOC0 initial output level select | R/W |
|---|---|---|
| Disabled in PWM3 mode. | | R/W |

| TOB0 | TRDIOB0 initial output level select [Note 2] | R/W |
|---|---|---|
| 0 | Low initial output, high active level, high output at TRDGRB1 compare match, and low output at TRDGRB0 compare match | R/W |
| 1 | High initial output, low active level, low output at TRDGRB1 compare match, and high output at TRDGRB0 compare match | |

| TOA0 | TRDIOA0 initial output level select | R/W |
|---|---|---|
| 0 | Low initial output, high active level, high output at TRDGRA1 compare match, and low output at TRDGRA0 compare match | R/W |
| 1 | High initial output, low active level, low output at TRDGRA1 compare match, and high output at TRDGRA0 compare match | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** If the pin function is set for waveform output, the initial output level is output when the TRDOCR register is set.

### 8.2.11 Timer RD Digital Filter Function Select Register i (TRDDFi) (i = 0 or 1)

**Figure 8-14. Format of Timer RD Digital Filter Function Select Register i (TRDDFi) (i = 0 or 1) [Input Capture Function]**

Address: F026AH (TRDDF0), F026BH (TRDDF1) After Reset: 00H Note 1

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDDFi | DFCK1 | DFCK0 | PENB1 | PENB0 | DFD | DFC | DFB | DFA |

| DFCK1 | DFCK0 | Clock select for digital filter function Note 2 | R/W |
|---|---|---|---|
| 0 | 0 | $f_{TRD}/32$ | R/W |
| 0 | 1 | $f_{TRD}/8$ | |
| 1 | 0 | $f_{TRD}$ | |
| 1 | 1 | Count source (clock selected by bits TCK0 to TCK2 in the TRDCRi register) | |

| PENB1 | PENB0 | TRDIOBi pin pulse forced cutoff control | R/W |
|---|---|---|---|
| 0 | 0 | Set to 00B. | R/W |

| DFD | TRDIODi pin digital filter function select | R/W |
|---|---|---|
| 0 | Function is not used | R/W |
| 1 | Function is used | |
| If the digital filter is enabled, edge detection is performed after five or more cycles of the digital filter sampling clock have elapsed. | | |

| DFC | TRDIOCi pin digital filter function select | R/W |
|---|---|---|
| 0 | Function is not used | R/W |
| 1 | Function is used | |
| If the digital filter is enabled, edge detection is performed after five or more cycles of the digital filter sampling clock have elapsed. | | |

| DFB | TRDIOBi pin digital filter function select | R/W |
|---|---|---|
| 0 | Function is not used | R/W |
| 1 | Function is used | |
| If the digital filter is enabled, edge detection is performed after five or more cycles of the digital filter sampling clock have elapsed. | | |

| DFA | TRDIOAi pin digital filter function select | R/W |
|---|---|---|
| 0 | Function is not used | R/W |
| 1 | Function is used | |
| If the digital filter is enabled, edge detection is performed after five or more cycles of the digital filter sampling clock have elapsed. | | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** Set bits DFCK0 and DFCK1 before starting count operation.

**Figure 8-15. Format of Timer RD Digital Filter Function Select Register i (TRDDFi) (i = 0 or 1)**
**[PWM Function, Reset Synchronous PWM Mode, Complementary PWM Mode, and PWM3 Mode]**

Address: F026AH (TRDDF0), F026BH (TRDDF1) After Reset: 00H [Note]

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDDFi | DFCK1 | DFCK0 | PENB1 | PENB0 | DFD | DFC | DFB | DFA |

| DFCK1 | DFCK0 | TRDIOAi pin pulse forced cutoff control | R/W |
|---|---|---|---|
| 0 | 0 | Forced cutoff disabled | R/W |
| 0 | 1 | High-impedance output | |
| 1 | 0 | Low output | |
| 1 | 1 | High output | |
| Set these bits to 00B (forced cutoff disabled) if the corresponding pin is not used as a timer RD output port in these modes. Also, set these bits while the count is stopped. | | | |

| PENB1 | PENB0 | TRDIOBi pin pulse forced cutoff control | R/W |
|---|---|---|---|
| 0 | 0 | Forced cutoff disabled | R/W |
| 0 | 1 | High-impedance output | |
| 1 | 0 | Low output | |
| 1 | 1 | High output | |
| Set these bits to 00B (forced cutoff disabled) if the corresponding pin is not used as a timer RD output port in these modes. Also, set these bits while the count is stopped. | | | |

| DFD | DFC | TRDIOCi pin pulse forced cutoff control | R/W |
|---|---|---|---|
| 0 | 0 | Forced cutoff disabled | R/W |
| 0 | 1 | High-impedance output | |
| 1 | 0 | Low output | |
| 1 | 1 | High output | |
| Set these bits to 00B (forced cutoff disabled) if the corresponding pin is not used as a timer RD output port in these modes. Also, set these bits while the count is stopped. | | | |

| DFB | DFA | TRDIODi pin pulse forced cutoff control | R/W |
|---|---|---|---|
| 0 | 0 | Forced cutoff disabled | R/W |
| 0 | 1 | High-impedance output | |
| 1 | 0 | Low output | |
| 1 | 1 | High output | |
| Set these bits to 00B (forced cutoff disabled) if the corresponding pin is not used as a timer RD output port in these modes. Also, set these bits while the count is stopped. | | | |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

### 8.2.12 Timer RD Control Register i (TRDCRi) (i = 0 or 1)

The TRDCR1 register is not used in reset synchronous PWM mode or PWM3 mode.

**Figure 8-16. Format of Timer RD Control Register i (TRDCRi) (i = 0 or 1)**
**[Input Capture Function and Output Compare Function]**

Address: F0270H (TRDCR0), F0280H (TRDCR1) After Reset: 00H **Note 1**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDCRi | CCLR2 | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TCK2 | TCK1 | TCK0 |

| CCLR2 | CCLR1 | CCLR0 | TRDi counter clear select | R/W |
|---|---|---|---|---|
| 0 | 0 | 0 | Clear disabled (free-running operation) | R/W |
| 0 | 0 | 1 | Clear by input capture/compare match with TRDGRAi | |
| 0 | 1 | 0 | Clear by input capture/compare match with TRDGRBi | |
| 0 | 1 | 1 | Synchronous clear (clear simultaneously with other timer RDi counter) **Note 2** | |
| 1 | 0 | 0 | Do not set. | |
| 1 | 0 | 1 | Clear by input capture/compare match with TRDGRCi | |
| 1 | 1 | 0 | Clear by input capture/compare match with TRDGRDi | |
| 1 | 1 | 1 | Do not set. | |

| CKEG1 | CKEG0 | External clock edge select **Note 3** | R/W |
|---|---|---|---|
| 0 | 0 | Count at the rising edge | R/W |
| 0 | 1 | Count at the falling edge | |
| 1 | 0 | Count at both edges | |
| 1 | 1 | Do not set. | |

| TCK2 | TCK1 | TCK0 | Count source select | R/W |
|---|---|---|---|---|
| 0 | 0 | 0 | $f_{TRD}$ **Note 4** | R/W |
| 0 | 0 | 1 | $f_{TRD}/2$ **Notes 4, 6** | |
| 0 | 1 | 0 | $f_{TRD}/4$ **Notes 4, 6** | |
| 0 | 1 | 1 | $f_{TRD}/8$ **Notes 4, 6** | |
| 1 | 0 | 0 | $f_{TRD}/32$ **Notes 4, 6** | |
| 1 | 0 | 1 | TRDCLK input **Note 5** | |
| 1 | 1 | 0 | Do not set. | |
| 1 | 1 | 1 | Do not set. | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** Enabled when the TRDSYNC bit in the TRDMR register is 1 (TRD0 and TRD1 operate synchronously).

**3.** Valid when bits TCK2 to TCK0 are set to 101B (TRDCLK input) and the STCLK bit is set to 1 (external clock input enabled).

**4.** As the timer RD operating clock ($f_{TRD}$), $f_{CLK}$ is selected when FRQSEL4 = 0 in the user option byte (000C2H/020C2H), (PLLDIV1 = 0 or SELPLLS = 0), and TRD_CKSEL = 0. $f_{IH}$ is selected when FRQSEL4 = 1 and TRD_CKSEL = 0. $f_{PLL}$ is selected when (PLLDIV1 = 1 and SELPLLS = 1) and TRD_CKSEL = 0. $f_{SUB}$ is selected when SELLOSC = 0 and TRD_CKSEL = 1. $f_{IL}$ is selected when SELLOSC = 1 and TRD_CKSEL = 1. For details, see **Figure 8-40**.
When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

**5.** Valid when the STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).

**6.** With this setting, select $f_{CLK}$ as the timer RD operating clock ($f_{TRD}$).

**Figure 8-17. Format of Timer RD Control Register i (TRDCRi) (i = 0 or 1) [PWM Mode]**

Address: F0270H (TRDCR0), F0280H (TRDCR1) After Reset: 00H Note 1

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDCRi | CCLR2 | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TCK2 | TCK1 | TCK0 |

| CCLR2 | CCLR1 | CCLR0 | TRDi counter clear select | R/W |
|---|---|---|---|---|
| Set to 001B (TRDi register is cleared at compare match with TRDGRAi register). | | | | R/W |

| CKEG1 | CKEG0 | External clock edge select Note 2 | R/W |
|---|---|---|---|
| 0 | 0 | Count at the rising edge | R/W |
| 0 | 1 | Count at the falling edge | |
| 1 | 0 | Count at both edges | |
| 1 | 1 | Do not set. | |

| TCK2 | TCK1 | TCK0 | Count source select | R/W |
|---|---|---|---|---|
| 0 | 0 | 0 | $f_{CLK}$, $f_{IH}$, $f_{PLL}$, $f_{SUB}$, $f_{IL}$ Note 3 | R/W |
| 0 | 0 | 1 | $f_{CLK}/2$ | |
| 0 | 1 | 0 | $f_{CLK}/4$ | |
| 0 | 1 | 1 | $f_{CLK}/8$ | |
| 1 | 0 | 0 | $f_{CLK}/32$ | |
| 1 | 0 | 1 | TRDCLK input Note 4 | |
| 1 | 1 | 0 | Do not set. | |
| 1 | 1 | 1 | Do not set. | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** Valid when bits TCK2 to TCK0 are set to 101B (TRDCLK input) and the STCLK bit is set to 1 (external clock input enabled).

**3.** $f_{CLK}$ is selected when FRQSEL4 = 0 in the user option byte (000C2H/020C2H), (PLLDIV1 = 0 or SELPLLS = 0), and TRD_CKSEL = 0. $f_{IH}$ is selected when FRQSEL4 = 1 and TRD_CKSEL = 0. $f_{PLL}$ is selected when (PLLDIV1 = 1 and SELPLLS = 1) and TRD_CKSEL = 0. $f_{SUB}$ is selected when SELLOSC = 0 and TRD_CKSEL = 1. $f_{IL}$ is selected when SELLOSC = 1 and TRD_CKSEL = 1. For details, see **Figure 8-40**.

When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

**4.** Valid when the STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).

**Figure 8-18. Format of Timer RD Control Register 0 (TRDCR0) [Reset Synchronous PWM Mode]**

Address: F0270H (TRDCR0) After Reset: 00H **Note 1**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDCR0 | CCLR2 | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TCK2 | TCK1 | TCK0 |

| CCLR2 | CCLR1 | CCLR0 | TRD0 counter clear select | R/W |
|---|---|---|---|---|
| Set to 001B (TRD0 register is cleared at compare match with TRDGRA0 register). | | | | R/W |

| CKEG1 | CKEG0 | External clock edge select **Note 2** | R/W |
|---|---|---|---|
| 0 | 0 | Count at the rising edge | R/W |
| 0 | 1 | Count at the falling edge | |
| 1 | 0 | Count at both edges | |
| 1 | 1 | Do not set. | |

| TCK2 | TCK1 | TCK0 | Count source select | R/W |
|---|---|---|---|---|
| 0 | 0 | 0 | $f_{CLK}$, $f_{IH}$, $f_{PLL}$, $f_{SUB}$, $f_{IL}$ **Note 3** | R/W |
| 0 | 0 | 1 | $f_{CLK}/2$ | |
| 0 | 1 | 0 | $f_{CLK}/4$ | |
| 0 | 1 | 1 | $f_{CLK}/8$ | |
| 1 | 0 | 0 | $f_{CLK}/32$ | |
| 1 | 0 | 1 | TRDCLK input **Note 4** | |
| 1 | 1 | 0 | Do not set. | |
| 1 | 1 | 1 | Do not set. | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** Valid when bits TCK2 to TCK0 are set to 101B (TRDCLK input) and the STCLK bit is set to 1 (external clock input enabled).

**3.** $f_{CLK}$ is selected when FRQSEL4 = 0 in the user option byte (000C2H/020C2H), (PLLDIV1 = 0 or SELPLLS = 0), and TRD_CKSEL = 0. $f_{IH}$ is selected when FRQSEL4 = 1 and TRD_CKSEL = 0. $f_{PLL}$ is selected when (PLLDIV1 = 1 and SELPLLS = 1) and TRD_CKSEL = 0. $f_{SUB}$ is selected when SELLOSC = 0 and TRD_CKSEL = 1. $f_{IL}$ is selected when SELLOSC = 1 and TRD_CKSEL = 1. For details, see **Figure 8-40**.

When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

**4.** Valid when the STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).

**Figure 8-19. Format of Timer RD Control Register 0 (TRDCR0) [Complementary PWM Mode]**

Address: F0270H (TRDCR0) After Reset: 00H Note 1

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDCR0 | CCLR2 | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TCK2 | TCK1 | TCK0 |

| CCLR2 | CCLR1 | CCLR0 | TRD0 counter clear select | R/W |
|---|---|---|---|---|
| Set to 000B (clear disabled (free-running operation)). | | | | R/W |

| CKEG1 | CKEG0 | External clock edge select Notes 2, 3 | R/W |
|---|---|---|---|
| 0 | 0 | Count at the rising edge | R/W |
| 0 | 1 | Count at the falling edge | |
| 1 | 0 | Count at both edges | |
| 1 | 1 | Do not set. | |

| TCK2 | TCK1 | TCK0 | Count source select | R/W |
|---|---|---|---|---|
| 0 | 0 | 0 | $f_{CLK}$, $f_{IH}$, $f_{PLL}$, $f_{SUB}$, $f_{IL}$ Note 4 | R/W |
| 0 | 0 | 1 | $f_{CLK}/2$ | |
| 0 | 1 | 0 | $f_{CLK}/4$ | |
| 0 | 1 | 1 | $f_{CLK}/8$ | |
| 1 | 0 | 0 | $f_{CLK}/32$ | |
| 1 | 0 | 1 | TRDCLK input Note 5 | |
| 1 | 1 | 0 | Do not set. | |
| 1 | 1 | 1 | Do not set. | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** Valid when bits TCK2 to TCK0 are set to 101B (TRDCLK input) and the STCLK bit is set to 1 (external clock input enabled).

**3.** Set the same value to bits TCK0 to TCK2, CKEG0, and CKEG1 in registers TRDCR0 and TRDCR1.

**4.** $f_{CLK}$ is selected when FRQSEL4 = 0 in the user option byte (000C2H/020C2H), (PLLDIV1 = 0 or SELPLLS = 0), and TRD_CKSEL = 0. $f_{IH}$ is selected when FRQSEL4 = 1 and TRD_CKSEL = 0. $f_{PLL}$ is selected when (PLLDIV1 = 1 and SELPLLS = 1) and TRD_CKSEL = 0. $f_{SUB}$ is selected when SELLOSC = 0 and TRD_CKSEL = 1. $f_{IL}$ is selected when SELLOSC = 1 and TRD_CKSEL = 1. For details, see **Figure 8-40**.

When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

**5.** Valid when the STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).

**Figure 8-20. Format of Timer RD Control Register 0 (TRDCR0) [PWM3 Mode]**

Address: F0270H (TRDCR0) After Reset: 00H [Note 1]

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDCR0 | CCLR2 | CCLR1 | CCLR0 | CKEG1 | CKEG0 | TCK2 | TCK1 | TCK0 |

| CCLR2 | CCLR1 | CCLR0 | TRD0 counter clear select | R/W |
|---|---|---|---|---|
| Set to 001B (TRD0 register is cleared at compare match with TRDGRA0 register). | | | | R/W |

| CKEG1 | CKEG0 | External clock edge select | R/W |
|---|---|---|---|
| Disabled in PWM3 mode. | | | R/W |

| TCK2 | TCK1 | TCK0 | Count source select | R/W |
|---|---|---|---|---|
| 0 | 0 | 0 | $f_{CLK}$, $f_{IH}$, $f_{PLL}$, $f_{SUB}$, $f_{IL}$ [Note 2] | R/W |
| 0 | 0 | 1 | $f_{CLK}/2$ | |
| 0 | 1 | 0 | $f_{CLK}/4$ | |
| 0 | 1 | 1 | $f_{CLK}/8$ | |
| 1 | 0 | 0 | $f_{CLK}/32$ | |
| 1 | 0 | 1 | Do not set. | |
| 1 | 1 | 0 | Do not set. | |
| 1 | 1 | 1 | Do not set. | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** $f_{CLK}$ is selected when FRQSEL4 = 0 in the user option byte (000C2H/020C2H), (PLLDIV1 = 0 or SELPLLS = 0), and TRD_CKSEL = 0. $f_{IH}$ is selected when FRQSEL4 = 1 and TRD_CKSEL = 0. $f_{PLL}$ is selected when (PLLDIV1 = 1 and SELPLLS = 1) and TRD_CKSEL = 0. $f_{SUB}$ is selected when SELLOSC = 0 and TRD_CKSEL = 1. $f_{IL}$ is selected when SELLOSC = 1 and TRD_CKSEL = 1. For details, see **Figure 8-40**.

When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

### 8.2.13 Timer RD I/O Control Register Ai (TRDIORAi) (i = 0 or 1)

**Figure 8-21. Format of Timer RD I/O Control Register Ai (TRDIORAi) (i = 0 or 1) [Input Capture Function]**

Address: F0271H (TRDIORA0), F0281H (TRDIORA1) After Reset: 00H Note 1

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDIORAi | — | IOB2 | IOB1 | IOB0 | 0 | IOA2 | IOA1 | IOA0 |

| Bit 7 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| IOB2 | TRDGRBi mode select Note 2 | R/W |
|---|---|---|
| Set to 1 (input capture) in the input capture function. | | R/W |

| IOB1 | IOB0 | TRDGRBi control | R/W |
|---|---|---|---|
| 0 | 0 | Input capture to TRDGRBi at the rising edge | R/W |
| 0 | 1 | Input capture to TRDGRBi at the falling edge | |
| 1 | 0 | Input capture to TRDGRBi at both edges | |
| 1 | 1 | Do not set. | |

| Bit 3 | Reserved | R/W |
|---|---|---|
| 0 | Set to 0. | R/W |

| IOA2 | TRDGRAi mode select Note 3 | R/W |
|---|---|---|
| Set to 1 (input capture) in the input capture function. | | R/W |

| IOA1 | IOA0 | TRDGRAi control | R/W |
|---|---|---|---|
| 0 | 0 | Input capture to TRDGRAi at the rising edge | R/W |
| 0 | 1 | Input capture to TRDGRAi at the falling edge | |
| 1 | 0 | Input capture to TRDGRAi at both edges | |
| 1 | 1 | Do not set. | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** If 1 (buffer register for TRDGRBi register) is selected for the TRDBFDi bit in the TRDMR register, set the same value to the IOB2 bit in the TRDIORAi register and the IOD2 bit in the TRDIORCi register.

**3.** If 1 (buffer register for TRDGRAi register) is selected for the TRDBFCi bit in the TRDMR register, set the same value to the IOA2 bit in the TRDIORAi register and the IOC2 bit in the TRDIORCi register.

**Figure 8-22. Format of Timer RD I/O Control Register Ai (TRDIORAi) (i = 0 or 1) [Output Compare Function]**

Address: F0271H (TRDIORA0), F0281H (TRDIORA1) After Reset: 00H Note 1

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDIORAi | — | IOB2 | IOB1 | IOB0 | 0 | IOA2 | IOA1 | IOA0 |

| Bit 7 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| IOB2 | TRDGRBi mode select Note 2 | R/W |
|---|---|---|
| Set to 0 (output compare) in the output compare function. | | R/W |

| IOB1 | IOB0 | TRDGRBi control | R/W |
|---|---|---|---|
| 0 | 0 | Pin output by compare match is disabled (TRDIOBi pin functions as an I/O port) | R/W |
| 0 | 1 | Low output by compare match with TRDGRBi | |
| 1 | 0 | High output by compare match with TRDGRBi | |
| 1 | 1 | Toggle output by compare match with TRDGRBi | |

| Bit 3 | Reserved | R/W |
|---|---|---|
| 0 | Set to 0. | R/W |

| IOA2 | TRDGRAi mode select Note 3 | R/W |
|---|---|---|
| Set to 0 (output compare) in the output compare function. | | R/W |

| IOA1 | IOA0 | TRDGRAi control | R/W |
|---|---|---|---|
| 0 | 0 | Pin output by compare match is disabled (TRDIOAi pin functions as an I/O port) | R/W |
| 0 | 1 | Low output by compare match with TRDGRAi | |
| 1 | 0 | High output by compare match with TRDGRAi | |
| 1 | 1 | Toggle output by compare match with TRDGRAi | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** If 1 (buffer register for TRDGRBi register) is selected for the TRDBFDi bit in the TRDMR register, set the same value to the IOB2 bit in the TRDIORAi register and the IOD2 bit in the TRDIORCi register.

**3.** If 1 (buffer register for TRDGRAi register) is selected for the TRDBFCi bit in the TRDMR register, set the same value to the IOA2 bit in the TRDIORAi register and the IOC2 bit in the TRDIORCi register.

### 8.2.14 Timer RD I/O Control Register Ci (TRDIORCi) (i = 0 or 1)

**Figure 8-23. Format of Timer RD I/O Control Register Ci (TRDIORCi) [Input Capture Function]**

Address: F0272H (TRDIORC0), F0282H (TRDIORC1) After Reset: 88H Note 1

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDIORCi | IOD3 | IOD2 | IOD1 | IOD0 | IOC3 | IOC2 | IOC1 | IOC0 |

| IOD3 | TRDGRDi register function select | R/W |
|---|---|---|
| Set to 1 (general register or buffer register) in the input capture function. | | R/W |

| IOD2 | TRDGRDi mode select Note 2 | R/W |
|---|---|---|
| Set to 1 (input capture) in the input capture function. | | R/W |

| IOD1 | IOD0 | TRDGRDi control | R/W |
|---|---|---|---|
| 0 | 0 | Input capture to TRDGRDi at the rising edge | R/W |
| 0 | 1 | Input capture to TRDGRDi at the falling edge | |
| 1 | 0 | Input capture to TRDGRDi at both edges | |
| 1 | 1 | Do not set. | |

| IOC3 | TRDGRCi register function select | R/W |
|---|---|---|
| Set to 1 (general register or buffer register) in the input capture function. | | R/W |

| IOC2 | TRDGRCi mode select Note 3 | R/W |
|---|---|---|
| Set to 1 (input capture) in the input capture function. | | R/W |

| IOC1 | IOC0 | TRDGRCi control | R/W |
|---|---|---|---|
| 0 | 0 | Input capture to TRDGRCi at the rising edge | R/W |
| 0 | 1 | Input capture to TRDGRCi at the falling edge | |
| 1 | 0 | Input capture to TRDGRCi at both edges | |
| 1 | 1 | Do not set. | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** If 1 (buffer register for TRDGRBi register) is selected for the TRDBFDi bit in the TRDMR register, set the same value to the IOB2 bit in the TRDIORAi register and the IOD2 bit in the TRDIORCi register.

**3.** If 1 (buffer register for TRDGRAi register) is selected for the TRDBFCi bit in the TRDMR register, set the same value to the IOA2 bit in the TRDIORAi register and the IOC2 bit in the TRDIORCi register.

**Figure 8-24. Format of Timer RD I/O Control Register Ci (TRDIORCi) (i = 0 or 1) [Output Compare Function]**

Address: F0272H (TRDIORC0), F0282H (TRDIORC1) After Reset: 88H Note 1

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDIORCi | IOD3 | IOD2 | IOD1 | IOD0 | IOC3 | IOC2 | IOC1 | IOC0 |

| IOD3 | TRDGRDi register function select | R/W |
|---|---|---|
| 0 | TRDIOBi output register<br>(see **8.3.3 (2) Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi**) | R/W |
| 1 | General register or buffer register | |

| IOD2 | TRDGRDi mode select Note 2 | R/W |
|---|---|---|
| Set to 0 (output compare) in the output compare function. | | R/W |

| IOD1 | IOD0 | TRDGRDi control | R/W |
|---|---|---|---|
| 0 | 0 | Pin output by compare match is disabled | R/W |
| 0 | 1 | Low output by compare match with TRDGRDi | |
| 1 | 0 | High output by compare match with TRDGRDi | |
| 1 | 1 | Toggle output by compare match with TRDGRDi | |

| IOC3 | TRDGRCi register function select | R/W |
|---|---|---|
| 0 | TRDIOAi output register<br>(see **8.3.3 (2) Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi**) | R/W |
| 1 | General register or buffer register | |

| IOC2 | TRDGRCi mode select Note 3 | R/W |
|---|---|---|
| Set to 0 (output compare) in the output compare function. | | R/W |

| IOC1 | IOC0 | TRDGRCi control | R/W |
|---|---|---|---|
| 0 | 0 | Pin output by compare match is disabled | R/W |
| 0 | 1 | Low output by compare match with TRDGRCi | |
| 1 | 0 | High output by compare match with TRDGRCi | |
| 1 | 1 | Toggle output by compare match with TRDGRCi | |

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** If 1 (buffer register for TRDGRBi register) is selected for the TRDBFDi bit in the TRDMR register, set the same value to the IOB2 bit in the TRDIORAi register and the IOD2 bit in the TRDIORCi register.

**3.** If 1 (buffer register for TRDGRAi register) is selected for the TRDBFCi bit in the TRDMR register, set the same value to the IOA2 bit in the TRDIORAi register and the IOC2 bit in the TRDIORCi register.

### 8.2.15 Timer RD Status Register i (TRDSRi) (i = 0 or 1)

**Figure 8-25. Format of Timer RD Status Register i (TRDSRi) (i = 0 or 1) [Input Capture Function]**

Address: F0273H (TRDSR0), F0283H (TRDSR1) After Reset: 00H Note 1

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDSRi | — | — | UDF | OVF | IMFD | IMFC | IMFB | IMFA |

| Bits 7 to 6 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| UDF | Underflow flag Note 2 | R/W |
|---|---|---|
| Disabled in the input capture function. | | R/W |

| OVF | Overflow flag Note 3 | R/W |
|---|---|---|
| [Source for setting to 0]<br>Write 0 after reading. Note 4<br>[Source for setting to 1]<br>When the TRDi register overflows | | R/W |

| IMFD | Input capture/compare match flag D Note 7 | R/W |
|---|---|---|
| [Source for setting to 0]<br>Write 0 after reading. Note 4<br>[Source for setting to 1]<br>Input edge of TRDIODi pin Note 5 | | R/W |

| IMFC | Input capture/compare match flag C Note 7 | R/W |
|---|---|---|
| [Source for setting to 0]<br>Write 0 after reading. Note 4<br>[Source for setting to 1]<br>Input edge of TRDIOCi pin Note 5 | | R/W |

| IMFB | Input capture/compare match flag B Note 7 | R/W |
|---|---|---|
| [Source for setting to 0]<br>Write 0 after reading. Note 4<br>[Source for setting to 1]<br>Input edge of TRDIOBi pin Note 6 | | R/W |

| IMFA | Input capture/compare match flag A Note 7 | R/W |
|---|---|---|
| [Source for setting to 0]<br>Write 0 after reading. Note 4<br>[Source for setting to 1]<br>Input edge of TRDIOAi pin Note 6 | | R/W |

**(Notes are listed on the next page.)**

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set fCLK to fIH and TRD0EN = 1 before reading.

**2.** Nothing is assigned to bit 5 in the TRDSR0 register. The write value must be 0 for bit 5. The read value is 0.

**3.** When the counter value of timer RDi changes from FFFFH to 0000H, the overflow flag is set to 1. Also, if the counter value of timer RDi changes from FFFFH to 0000H due to an input capture/compare match during operation according to the settings of bits CCLR0 to CCLR2 in the TRDCRi register, the overflow flag is set to 1.

**4.** The writing results are as follows:

- Writing 1 has no effect.
- If the read value is 0, the bit remains unchanged even if 0 is written to it. (Even if the bit is changed from 0 to 1 after reading and then 0 is written to it, it remains 1.)
- If the read value is 1, writing 0 to the bit sets it to 0.
  Use either (a) or (b) described below to clear each bit of the TRDSRi register.
  (a) Set the TRDIERi register to 00H (disabling all interrupts) and then write 0 to all of the status flags.
  (b) When at least one bit in the TRDIERi register has the setting 1 and the status flag of an interrupt source enabled by the corresponding bit is 1, write 0 to all of the status flag bits whose settings are 1 in the TRDSRi register.

**5.** Edge selected by bits IOk1 and IOk0 (k = C or D) in the TRDIORCi register.
Including when the TRDBFki bit in the TRDMR register is 1 (TRDGRki is buffer register).

**6.** Edge selected by bits IOj1 and IOj0 (j = A or B) in the TRDIORAi register.

**7.** When the DTC is used, bits IMFA, IMFB, IMFC, and IMFD are set to 1 after DTC transfer is completed.

**Figure 8-26. Format of Timer RD Status Register i (TRDSRi) (i = 0 or 1)**
**[Functions Other Than Input Capture Function]**

Address: F0273H (TRDSR0), F0283H (TRDSR1) After Reset: 00H Note 1

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDSRi | — | — | UDF | OVF | IMFD | IMFC | IMFB | IMFA |

| Bits 7 to 6 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| UDF | Underflow flag Note 2 | R/W |
|---|---|---|
| In complementary PWM mode<br>[Source for setting to 0]<br>Write 0 after reading. Note 3<br>[Sources for setting to 1]<br>When TRDi underflows.<br>Enabled only in complementary PWM mode. | | R/W |

| OVF | Overflow flag Note 4 | R/W |
|---|---|---|
| [Source for setting to 0]<br>Write 0 after reading. Note 3<br>[Source for setting to 1]<br>When the TRDi register overflows | | R/W |

| IMFD | Input capture/compare match flag D Note 6 | R/W |
|---|---|---|
| [Source for setting to 0]<br>Write 0 after reading. Note 3<br>[Source for setting to 1]<br>When the values of TRDi and TRDGRDi match. Note 5 | | R/W |

| IMFC | Input capture/compare match flag C Note 6 | R/W |
|---|---|---|
| [Source for setting to 0]<br>Write 0 after reading. Note 3<br>[Source for setting to 1]<br>When the values of TRDi and TRDGRCi match. Note 5 | | R/W |

| IMFB | Input capture/compare match flag B Note 6 | R/W |
|---|---|---|
| [Source for setting to 0]<br>Write 0 after reading. Note 3<br>[Source for setting to 1]<br>When the values of TRDi and TRDGRBi match. | | R/W |

| IMFA | Input capture/compare match flag A Note 6 | R/W |
|---|---|---|
| [Source for setting to 0]<br>Write 0 after reading. Note 3<br>[Source for setting to 1]<br>When the values of TRDi and TRDGRAi match. | | R/W |

**(Notes are listed on the next page.)**

**Notes 1.** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**2.** Nothing is assigned to bit 5 in the TRDSR0 register. The write value must be 0 for bit 5. The read value is 0.

**3.** The writing results are as follows:

- Writing 1 has no effect.
- If the read value is 0, the bit remains unchanged even if 0 is written to it. (Even if the bit is changed from 0 to 1 after reading and then 0 is written to it, it remains 1.)
- If the read value is 1, writing 0 to the bit sets it to 0.

Use either (a) or (b) described below to clear each bit of the TRDSRi register.

(a) Set the TRDIERi register to 00H (disabling all interrupts) and then write 0 to all of the status flags.

(b) When at least one bit in the TRDIERi register has the setting 1 and the status flag of an interrupt source enabled by the corresponding bit is 1, write 0 to all of the status flag bits whose settings are 1 in the TRDSRi register.

**4.** When the counter value of timer RDi changes from FFFFH to 0000H, the overflow flag is set to 1. Also, if the counter value of timer RDi changes from FFFFH to 0000H due to an input capture/compare match during operation according to the settings of bits CCLR0 to CCLR2 in the TRDCRi register, the overflow flag is set to 1.

**5.** Including when the TRDBFki bit (k = C or D) in the TRDMR register is set to 1 (TRDGRKi is buffer register).

**6.** When the DTC is used, bits IMFA, IMFB, IMFC, and IMFD are set to 1 after DTC transfer is completed.

### 8.2.16 Timer RD Interrupt Enable Register i (TRDIERi) (i = 0 or 1)

**Figure 8-27. Format of Timer RD Interrupt Enable Register i (TRDIERi) (i = 0 or 1)**

Address: F0274H (TRDIER0), F0284H (TRDIER1) After Reset: 00H [Note]

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDIERi | — | — | — | OVIE | IMIED | IMIEC | IMIEB | IMIEA |

| Bits 7 to 5 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| OVIE | Overflow/underflow interrupt enable | R/W |
|---|---|---|
| 0 | Interrupt (OVI) by bits OVF and UDF disabled | R/W |
| 1 | Interrupt (OVI) by bits OVF and UDF enabled | |

| IMIED | Input capture/compare match interrupt enable D | R/W |
|---|---|---|
| 0 | Interrupt (IMID) by the IMFD bit is disabled | R/W |
| 1 | Interrupt (IMID) by the IMFD bit is enabled | |

| IMIEC | Input capture/compare match interrupt enable C | R/W |
|---|---|---|
| 0 | Interrupt (IMIC) by the IMFC bit is disabled | R/W |
| 1 | Interrupt (IMIC) by the IMFC bit is enabled | |

| IMIEB | Input capture/compare match interrupt enable B | R/W |
|---|---|---|
| 0 | Interrupt (IMIB) by the IMFB bit is disabled | R/W |
| 1 | Interrupt (IMIB) by the IMFB bit is enabled | |

| IMIEA | Input capture/compare match interrupt enable A | R/W |
|---|---|---|
| 0 | Interrupt (IMIA) by the IMFA bit is disabled | R/W |
| 1 | Interrupt (IMIA) by the IMFA bit is enabled | |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

### 8.2.17 Timer RD PWM Function Output Level Control Register i (TRDPOCRi) (i = 0 or 1)

Settings to the TRDPOCRi register are enabled only in PWM function. When not in PWM function, they are disabled.

**Figure 8-28. Format of Timer RD PWM Function Output Level Control Register i (TRDPOCRi) (i= 0 or 1) [PWM Function]**

Address: F0275H (TRDPOCR0), F0285H (TRDPOCR1) After Reset: 00H [Note]

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TRDPOCRi | — | — | — | — | — | POLD | POLC | POLB |

| Bits 7 to 3 | Nothing is assigned | R/W |
|---|---|---|
| — | The write value must be 0. The read value is 0. | R |

| POLD | PWM function output level control D | R/W |
|---|---|---|
| 0 | TRDIODi output level is low active | R/W |
| 1 | TRDIODi output level is high active | |

| POLC | PWM function output level control C | R/W |
|---|---|---|
| 0 | TRDIOCi output level is low active | R/W |
| 1 | TRDIOCi output level is high active | |

| POLB | PWM function output level control B | R/W |
|---|---|---|
| 0 | TRDIOBi output level is low active | R/W |
| 1 | TRDIOBi output level is high active | |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

### 8.2.18 Timer RD Counter i (TRDi) (i = 0 or 1)

**[Timer Mode]**

Access the TRDi register in 16-bit units. Do not access it in 8-bit units.

**[Reset Synchronous PWM Mode and PWM3 Mode]**

Access the TRD0 register in 16-bit units. Do not access it in 8-bit units. The TRD1 register is not used in reset synchronous PWM mode and PWM3 mode.

**[Complementary PWM Mode (TRD0)]**

Access the TRD0 register in 16-bit units. Do not access it in 8-bit units.

**[Complementary PWM Mode (TRD1)]**

Access the TRD1 register in 16-bit units. Do not access it in 8-bit units.

**Figure 8-29. Format of Timer RD Counter i (TRDi) (i = 0 or 1) [Timer Mode]**

Address: F0276H (TRD0), F0286H (TRD1) After Reset: 0000H **Note**

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRDi | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| — | Function | Setting Range | R/W |
|---|---|---|---|
| Bits 15 to 0 | Count the count source. Count operation is incremented.<br>When an overflow occurs, the OVF bit in the TRDSRi register is set to 1. | 0000H to FFFFH | R/W |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**Figure 8-30. Format of Timer RD Counter i (TRDi) (i = 0 or 1) [Reset Synchronous PWM Mode and PWM3 Mode]**

Address: F0276H (TRD0), F0286H (TRD1) After Reset: 0000H **Note**

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRDi | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| — | Function | Setting Range | R/W |
|---|---|---|---|
| Bits 15 to 0 | Count the count source. Count operation is incremented.<br>When an overflow occurs, the OVF bit in the TRDSR0 register is set to 1. | 0000H to FFFFH | R/W |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**Figure 8-31. Format of Timer RD Counter i (TRDi) (i = 0 or 1) [Complementary PWM Mode (TRD0)]**

Address: F0276H (TRD0), F0286H (TRD1) After Reset: 0000H **Note**

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRDi | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| — | Function | Setting Range | R/W |
|---|---|---|---|
| Bits 15 to 0 | Dead time must be set.<br>Count the count source. Count operation is incremented or decremented.<br>When an overflow occurs, the OVF bit in the TRDSR0 register is set to 1. | 0001H to FFFFH | R/W |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**Figure 8-32. Format of Timer RD Counter i (TRDi) (i = 0 or 1) [Complementary PWM Mode (TRD1)]**

Address: F0276H (TRD0), F0286H (TRD1) After Reset: 0000H **Note**

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRDi | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| — | Function | Setting Range | R/W |
|---|---|---|---|
| Bits 15 to 0 | Set to 0000H.<br>Count the count source. Count operation is incremented or decremented.<br>When an underflow occurs, the UDF bit in the TRDSR1 register is set to 1. | 0000H to FFFFH | R/W |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

### 8.2.19 Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi,TRDGRCi, TRDGRDi) (i = 0 or 1)

**[Input Capture Function]**

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

The following registers are disabled in the input capture function:
TRDOER1, TRDOER2, TRDOCR, TRDPOCR0, and TRDPOCR1

Set the pulse width of the input capture signal applied to the TRDIOji pin to three or more cycles of the timer RD operating clock ($f_{TRD}$) when no digital filter is used (j = A, B, C, or D) (when no digital filter is used).

**[Output Compare Function]**

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

The following registers are disabled in the output compare function:
TRDDF0, TRDDF1, TRDPOCR0, and TRDPOCR1

**[PWM Function]**

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

The following registers are disabled in PWM function:
TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDIORA1, and TRDIORC1

**[Reset Synchronous PWM Mode]**

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

The following registers are disabled in reset synchronous PWM mode:
TRDPMR, TRDOCR **Note**, TRDDF0, TRDDF1,TRDIORA0, TRDIORC0, TRDPOCR0, TRDIORA1, TRDIORC1, and TRDPOCR1

**Note** The TOC0 bit in the TRDOCR register is enabled as an initial output setting of TRDIOC0 in reset synchronous PWM mode and complementary PWM mode.

**[Complementary PWM Mode]**

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.
The TRDGRC0 register is not used in complementary PWM mode.

The following registers are disabled in complementary PWM mode.
<R> TRDPMR, TRDOCR **Note**, TRDIORA0, TRDIORC0, TRDPOCR0, TRDIORA1, TRDIORC1, and TRDPOCR1

**Note** The TOC0 bit in the TRDOCR register is enabled as an initial output setting of TRDIOC0 in reset synchronous PWM mode and complementary PWM mode.

Since values cannot be written to the TRDGRB0, TRDGRA1, or TRDGRB1 register directly after count operation starts (prohibited item), use the TRDGRD0, TRDGRC1, or TRDGRD1 register as a buffer register.

However, to write data to the TRDGRD0, TRDGRC1, or TRDGRD1 register, set bits TRDBFD0, TRDBFC1, and TRDBFD1 to 0 (general register). After this, bits TRDBFD0, TRDBFC1, and TRDBFD1 may be set to 1 (buffer register).

**[PWM3 Mode]**

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

The following registers are disabled in PWM3 mode:
TRDPMR, TRDDF0, TRDDF1, TRDIORA0, TRDIORC0, TRDPOCR0, TRDIORA1, TRDIORC1, and TRDPOCR1

Registers TRDGRC0, TRDGRC1, TRDGRD0, and TRDGRD1 are not used in PWM3 mode. To use them as buffer registers, set bits TRDBFC0, TRDBFC1, TRDBFD0, and TRDBFD1 to 0 (general register) and write a value to the TRDGRC0, TRDGRC1, TRDGRD0, or TRDGRD1 register. After this, bits TRDBFC0, TRDBFC1, TRDBFD0, and TRDBFD1 may be set to 1 (buffer register).

**Figure 8-33. Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi,TRDGRCi, TRDGRDi) (i = 0 or 1) [Input Capture Function]**

Address: F0278H (TRDGRA0), F027AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0288H (TRDGRA1), F028AH (TRDGRB1), FFF5CH (TRDGRC1), FFF5EH (TRDGRD1)

After Reset: FFFFH [Note]

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRDGRAi<br>TRDGRBi<br>TRDGRCi<br>TRDGRDi | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| — | Function | R/W |
|---|---|---|
| Bits 15 to 0 | See **Table 8-3 TRDGRji Register Functions in Input Capture Function**. | R/W |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**Table 8-3. TRDGRji Register Functions in Input Capture Function**

| Register | Setting | Register Function | Input-Capture Input Pin |
|---|---|---|---|
| TRDGRAi | — | General register. The value of the TRDi register can be read at input capture. | TRDIOAi |
| TRDGRBi | | | TRDIOBi |
| TRDGRCi | TRDBFCi = 0 | General register. The value of the TRDi register can be read at input capture. | TRDIOCi |
| TRDGRDi | TRDBFDi = 0 | | TRDIODi |
| TRDGRCi | TRDBFCi = 1 | Buffer register. The value of the TRDi register can be read at input capture (see **8. 3. 1 (2) Buffer Operation**). | TRDIOAi |
| TRDGRDi | TRDBFDi = 1 | | TRDIOBi |

**Remark** i = 0 or 1, j = A, B, C, or D
TRDBFCi, TRDBFDi: Bits in TRDMR register

**Figure 8-34. Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, TRDGRDi) (i = 0 or 1) [Output Compare Function]**

Address: F0278H (TRDGRA0), F027AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0288H (TRDGRA1), F028AH (TRDGRB1), FFF5CH (TRDGRC1), FFF5EH (TRDGRD1)

After Reset: FFFFH [Note]

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRDGRAi<br>TRDGRBi<br>TRDGRCi<br>TRDGRDi | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| — | Function | R/W |
|---|---|---|
| Bits 15 to 0 | See **Table 8-4 TRDGRji Register Functions in Output Compare Function**. | R/W |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**Table 8-4. TRDGRji Register Functions in Output Compare Function**

| Register | Setting | | Register Function | | Output-Compare Output Pin |
|---|---|---|---|---|---|
| | TRDBFji | IOj3 | | | |
| TRDGRAi | — | — | General register. Write the compare value. | | TRDIOAi |
| TRDGRBi | | | | | TRDIOBi |
| TRDGRCi | 0 | 1 | General register. Write the compare value. | | TRDIOCi |
| TRDGRDi | | | | | TRDIODi |
| TRDGRCi | 1 | 1 | Buffer register. Write the next compare value (see **8. 3. 1 (2) Buffer Operation**). | | TRDIOAi |
| TRDGRDi | | | | | TRDIOBi |
| TRDGRCi | 0 | 0 | TRDIOAi output control | (See **8. 3. 3 (2) Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi**.) | TRDIOAi |
| TRDGRDi | | | TRDIOBi output control | | TRDIOBi |

**Caution When the setting of bits TCK2 to TCK0 in the TRDCRi register is 000B ($f_{CLK}$, $f_{IH}$, $f_{PLL}$, $f_{SUB}$, and $f_{IL}$) and the compare value is set to 0000H, a request signal to the data transfer controller (DTC) and the event link controller (ELC) is generated only once immediately after the count starts. When the compare value is 0001H or higher, a request signal is generated each time a compare match occurs.**

**Remark** i = 0 or 1, j = A, B, C, or D
TRDBFji: Bit in TRDMR register, IOj3: Bit in TRDIORCi register

**Figure 8-35. Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi,TRDGRCi, TRDGRDi) (i = 0 or 1) [PWM Mode]**

Address: F0278H (TRDGRA0), F027AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0288H (TRDGRA1), F028AH (TRDGRB1), FFF5CH (TRDGRC1), FFF5EH (TRDGRD1)

After Reset: FFFFH Note

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRDGRAi<br>TRDGRBi<br>TRDGRCi<br>TRDGRDi | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| — | Function | R/W |
|---|---|---|
| Bits 15 to 0 | See **Table 8-5 TRDGRji Register Functions in PWM Function**. | R/W |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**Table 8-5. TRDGRji Register Functions in PWM Function**

| Register | Setting | Register Function | PWM Output Pin |
|---|---|---|---|
| TRDGRAi | — | General register. Set the PWM period. | — |
| TRDGRBi | — | General register. Set the changing point of PWM output. | TRDIOBi |
| TRDGRCi | TRDBFCi = 0 | General register. Set the changing point of PWM output. | TRDIOCi |
| TRDGRDi | TRDBFDi = 0 | | TRDIODi |
| TRDGRCi | TRDBFCi = 1 | Buffer register. Set the next PWM period (see **8. 3. 1 (2) Buffer Operation**). | — |
| TRDGRDi | TRDBFDi = 1 | Buffer register. Set the changing point of the next PWM output (see **8. 3. 1 (2) Buffer Operation**). | TRDIOBi |

**Caution When the setting of bits TCK2 to TCK0 in the TRDCRi register is 000B ($f_{CLK}$, $f_{IH}$, $f_{PLL}$, $f_{SUB}$, and $f_{IL}$) and the compare value is set to 0000H, a request signal to the data transfer controller (DTC) and the event link controller (ELC) is generated only once immediately after the count starts. When the compare value is 0001H or higher, a request signal is generated each time a compare match occurs.**

**Remark** i = 0 or 1, j = A, B, C, or D
TRDBFCi, TRDBFDi: Bits in TRDMR register

**Figure 8-36. Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi,TRDGRCi, TRDGRDi) (i = 0 or 1) [Reset Synchronous PWM Mode]**

Address: F0278H (TRDGRA0), F027AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0288H (TRDGRA1), F028AH (TRDGRB1), FFF5CH (TRDGRC1), FFF5EH (TRDGRD1)

After Reset: FFFFH [Note]

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRDGRAi<br>TRDGRBi<br>TRDGRCi<br>TRDGRDi | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| — | Function | R/W |
|---|---|---|
| Bits 15 to 0 | See **Table 8-6 TRDGRji Register Functions in Reset Synchronous PWM Mode**. | R/W |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**Table 8-6. TRDGRji Register Functions in Reset Synchronous PWM Mode**

| Register | Setting | Register Function | PWM Output Pin |
|---|---|---|---|
| TRDGRA0 | — | General register. Set the PWM period. | (TRDIOC0, output inverted every PWM period) |
| TRDGRB0 | — | General register. Set the changing point of PWM1 output. | TRDIOB0<br>TRDIOD0 |
| TRDGRC0 | TRDBFC0 = 0 | (Not used in reset synchronous PWM mode.) | — |
| TRDGRD0 | TRDBFC0 = 0 | | |
| TRDGRA1 | — | General register. Set the changing point of PWM2 output. | TRDIOA1<br>TRDIOC1 |
| TRDGRB1 | — | General register. Set the changing point of PWM3 output. | TRDIOB1<br>TRDIOD1 |
| TRDGRC1 | TRDBFC1= 0 | (Not used in reset synchronous PWM mode.) | — |
| TRDGRD1 | TRDBFD1 = 0 | | |
| TRDGRC0 | TRDBFC0 = 1 | Buffer register. Set the next PWM period (see **8. 3. 1 (2) Buffer Operation**). | (TRDIOC0, output inverted every PWM period) |
| TRDGRD0 | TRDBFD0 = 1 | Buffer register. Set the changing point of the next PWM1 (see **8. 3. 1 (2) Buffer Operation**). | TRDIOB0<br>TRDIOD0 |
| TRDGRC1 | TRDBFC1 = 1 | Buffer register. Set the changing point of the next PWM2 (see **8. 3. 1 (2) Buffer Operation**). | TRDIOA1<br>TRDIOC1 |
| TRDGRD1 | TRDBFD1 = 1 | Buffer register. Set the changing point of the next PWM3 (see **8. 3. 1 (2) Buffer Operation**). | TRDIOB1<br>TRDIOD1 |

**Caution When the setting of bits TCK2 to TCK0 in the TRDCR0 register is 000B ($f_{CLK}$, $f_{IH}$, $f_{PLL}$, $f_{SUB}$, and $f_{IL}$) and the compare value is set to 0000H, a request signal to the data transfer controller (DTC) and the event link controller (ELC) is generated only once immediately after the count starts. When the compare value is 0001H or higher, a request signal is generated each time a compare match occurs.**

**Remark** i = 0 or 1, j = A, B, C, or D
TRDBFC0, TRDBFD0, TRDBFC1, TRDBFD1: Bits in TRDMR register

**Figure 8-37. Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi,TRDGRCi, TRDGRDi) (i = 0 or 1) [Complementary PWM Mode]**

Address: F0278H (TRDGRA0), F027AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0288H (TRDGRA1), F028AH (TRDGRB1), FFF5CH (TRDGRC1), FFF5EH (TRDGRD1)

After Reset: FFFFH [Note]

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRDGRAi<br>TRDGRBi<br>TRDGRCi<br>TRDGRDi | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| — | Function | R/W |
|---|---|---|
| Bits 15 to 0 | See **Table 8-7 TRDGRji Register Functions in Complementary PWM Mode**. | R/W |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**Table 8-7. TRDGRji Register Functions in Complementary PWM Mode**

| Register | Setting | Register Function | PWM Output Pin |
|---|---|---|---|
| TRDGRA0 | — | General register. Set the PWM period at initialization.<br>Setting range: ≥ Value set in TRD0 register<br>≤ FFFFh - value set in TRD0 register<br>Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts). | (TRDIOC0, output inverted every half period) |
| TRDGRB0 | — | General register. Set the changing point of PWM1 output at initialization.<br>Setting range: ≥ Value set in TRD0 register<br>≤ Value set in TRDGRA0 register - value set in TRD0 register<br>Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts). | TRDIOB0<br>TRDIOD0 |
| TRDGRA1 | — | General register. Set the changing point of PWM2 output at initialization.<br>Setting range: ≥ Value set in TRD0 register<br>≤ Value set in TRDGRA0 register - value set in TRD0 register<br>Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts). | TRDIOA1<br>TRDIOC1 |
| TRDGRB1 | — | General register. Set the changing point of PWM3 output at initialization.<br>Setting range: ≥ Value set in TRD0 register<br>≤ Value set in TRDGRA0 register - value set in TRD0 register<br>Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts). | TRDIOB1<br>TRDIOD1 |
| TRDGRC0 | — | (Not used in complementary PWM mode.) | — |
| TRDGRD0 | TRDBFD0 = 1 | Buffer register. Set the changing point of next PWM1 output (see **8. 3. 1 (2) Buffer Operation**).<br>Setting range: ≥ Value set in TRD0 register<br>≤ Value set in TRDGRA0 register - value set in TRD0 register<br>Set this register to the same value as the TRDGRB0 register for initialization. | TRDIOB0<br>TRDIOD0 |
| TRDGRC1 | TRDBFC1 = 1 | Buffer register. Set the changing point of next PWM2 output (see **8. 3. 1 (2) Buffer Operation**).<br>Setting range: ≥ Value set in TRD0 register<br>≤ Value set in TRDGRA0 register - value set in TRD0 register<br>Set this register to the same value as the TRDGRA1 register for initialization. | TRDIOA1<br>TRDIOC1 |
| TRDGRD1 | TRDBFD1 = 1 | Buffer register. Set the changing point of next PWM3 output (see **8. 3. 1 (2) Buffer Operation**).<br>Setting range: ≥ Value set in TRD0 register<br>≤ Value set in TRDGRA0 register - value set in TRD0 register<br>Set this register to the same value as the TRDGRB1 register for initialization. | TRDIOB1<br>TRDIOD1 |

**Caution When the setting of bits TCK2 to TCK0 in the TRDCRi register is 000B ($f_{CLK}$, $f_{IH}$, $f_{PLL}$, $f_{SUB}$, and $f_{IL}$) and the compare value is set to 0000H, a request signal to the data transfer controller (DTC) and the event link controller (ELC) is generated only once immediately after the count starts. When the compare value is 0001H or higher, a request signal is generated each time a compare match occurs.**

**Remark** i = 0 or 1, j = A, B, C, or D
TRDBFD0, TRDBFC1, TRDBFD1: Bits in TRDMR register

**Figure 8-38. Format of Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi,TRDGRCi, TRDGRDi) (i = 0 or 1) [PWM3 Mode]**

Address: F0278H (TRDGRA0), F027AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0288H (TRDGRA1), F028AH (TRDGRB1), FFF5CH (TRDGRC1), FFF5EH (TRDGRD1)

After Reset: FFFFH [Note]

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRDGRAi<br>TRDGRBi<br>TRDGRCi<br>TRDGRDi | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| — | Function | R/W |
|---|---|---|
| Bits 15 to 0 | See **Table 8-8 TRDGRji Register Functions in PWM3 Mode**. | R/W |

**Note** The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**Table 8-8. TRDGRji Register Functions in PWM3 Mode**

| Register | Setting | Register Function | PWM Output Pin |
|---|---|---|---|
| TRDGRA0 | — | General register. Set the PWM period .<br>Setting range: ≥ Value set in TRDGRA1 register | TRDIOA0 |
| TRDGRA1 | — | General register. Set the changing point (active level timing) of PWM output<br>Setting range: ≤ Value set in TRDGRA0 register | — |
| TRDGRB0 | — | General register. Set the changing point (the timing for returning to initial output level) of PWM output.<br>Setting range: ≥ Value set in TRDGRB1 register and ≤ Value set in TRDGRA0 register | TRDIOB0 |
| TRDGRB1 | — | General register. Set the changing point (active level timing) of PWM output<br>Setting range: ≤ Value set in TRDGRB0 register | — |
| TRDGRC0 | TRDBFC0 = 0 | (Not used in PWM3 mode.) | — |
| TRDGRC1 | TRDBFC1 = 0 | | |
| TRDGRD0 | TRDBFD0 = 0 | | |
| TRDGRD1 | TRDBFD1 = 0 | | |
| TRDGRC0 | TRDBFC0 = 1 | Buffer register. Set the next PWM period (see **8. 3. 1 (2) Buffer Operation**).<br>Setting range: ≥ Value set in TRDGRC1 register | TRDIOA0 |
| TRDGRC1 | TRDBFC1 = 1 | Buffer register. Set the changing point of next PWM output<br>(see **8. 3. 1 (2) Buffer Operation**).<br>Setting range: ≤ Value set in TRDGRC0 register | — |
| TRDGRD0 | TRDBFD0 = 1 | Buffer register. Set the changing point of next PWM output<br>(see **8. 3. 1 (2) Buffer Operation**).<br>Setting range: ≥ Value set in TRDGRD1 register and ≤ Value set in TRDGRC0 register | TRDIOB0 |
| TRDGRD1 | TRDBFD1 = 1 | Buffer register. Set the changing point of next PWM output<br>(see **8. 3. 1 (2) Buffer Operation**).<br>Setting range: ≤ Value set in TRDGRD0 register | — |

**Caution When the setting of bits TCK2 to TCK0 in the TRDCR0 register is 000B ($f_{CLK}$, $f_{IH}$, $f_{PLL}$, $f_{SUB}$, and $f_{IL}$) and the compare value is set to 0000H, a request signal to the data transfer controller (DTC) and the event link controller (ELC) is generated only once immediately after the count starts. When the compare value is 0001H or higher, a request signal is generated each time a compare match occurs.**

**Remark** i = 0 or 1, j = A, B, C, or D
TRDBFC0, TRDBFD0, TRDBFC1, TRDBFD1: Bits in TRDMR register

### 8.2.20 PWM Output Delay Control Register 0 (PWMDLY0)

This register controls output delay of PWM output signal output from the TRDIOj0 and TRDIOj1 pins.

Set the PWMDLY0 register by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

PWMDLY0 register

Address: F0229H After Reset: 00H R/W

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| PWMDLY0 | TRDD11Note | TRDD10Note | TRDC11Note | TRDC10Note | TRDB11 | TRDB10 | TRDA11 | TRDA10 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Address: F0228H After Reset: 00H R/W

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PWMDLY0 | TRDD01 | TRDD00 | TRDC01 | TRDC00 | TRDB01 | TRDB00 | TRDA01 | TRDA00 |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

**Table 8-9. PWM Output Delay Period Control by TRDIOj1 of Timer RD1**

| TRDj11 | TRDj10 | PWM Output Delay Control by TRDIOj1 of Timer RD1 |
|---|---|---|
| 0 | 0 | No delay. |
| 0 | 1 | TRDIOj1 signal delayed by one cycle of the timer RD operating clock ($f_{TRD}$). |
| 1 | 0 | TRDIOj1 signal delayed by two cycles of the timer RD operating clock ($f_{TRD}$). |
| 1 | 1 | TRDIOj1 signal delayed by three cycles of the timer RD operating clock ($f_{TRD}$). |

j = A, B, C, or D

**Table 8-10. PWM Output Delay Period Control by TRDIOj0 of Timer RD0**

| TRDj01 | TRDj00 | PWM Output Delay Control by TRDIOj0 of Timer RD0 |
|---|---|---|
| 0 | 0 | No delay. |
| 0 | 1 | TRDIOj0 signal delayed by one cycle of the timer RD operating clock ($f_{TRD}$). |
| 1 | 0 | TRDIOj0 signal delayed by two cycles of the timer RD operating clock ($f_{TRD}$). |
| 1 | 1 | TRDIOj0 signal delayed by three cycles of the timer RD operating clock ($f_{TRD}$). |

j = A, B, C, or D

**Note** If this register is set for a delay, this affects PWM outputs of TRDIOC1 and TRDIOD1, but doesn't affect the operation of the timer output signal to internally connected peripheral functions.

**Cautions 1. Set the PWMDLY0 register before outputting a PWM signal.**

**2. Access the PWMDLY0 register in 16 bits. 1-bit access and 8-bit access are prohibited.**

**3. If this register is not used for PWM output, it should be cleared to 0. This is because the timer output is delayed by the output delay setting shown above even when the timer output mode other than PWM output mode is selected.**

**4. When setting this register after the PWM output is stopped, wait for four cycles of the timer RD operating clock ($f_{TRD}$) before the setting.**

**5. When using SNZOUT, set TRDC0n to 0 before entering STOP mode (n = 0, 1).**

**6. Even if this register is set for a delay, this doesn't affect the operation of other pin functions multiplexed on the same pin as the TRDIOji pin function (j = A, B, C, D, i = 0, 1).**

### 8.2.21 Port mode registers (PM1, PM3, PM12)

These registers set input/output of port in 1-bit units. PM1, PM3, and PM12 are used in timer RD.

When using the ports (P13/TRDIOA0, P16/TRDIOC1, etc.) to be shared with the timer output pin for timer output, set the bit in the port mode register (PMxx) and the bit in the port register (Pxx) corresponding to each port to 0.

Example: When using P13/TRDIOA0 for timer output
Set the PM13 bit of port mode register 1 to 0.
Set the P13 bit of port register 1 to 0.

When using the ports (P13/TRDIOA0, P16/TRDIOC1, etc.) to be shared with the timer input pin for timer input, set the bit in the port mode register (PMxx) corresponding to each port to 1. At this time, the bit in the port register (Pxx) may be 0 or 1.

Example: When using P13/TRDIOA0 for timer input
Set the PM13 bit of port mode register 1 to 1.
Set the P13 bit of port register 1 to 0 or 1.

The PM1, PM3, and PM12 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to FFH.

**Figure 8-39. Format of Port Mode Registers (PM1, PM3, PM12) (100-pin products)**

Address: FFF21H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM1 | PM17 | PM16 | PM15 | PM14 | PM13 | PM12 | PM11 | PM10 |

Address: FFF23H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM3 | 1 | 1 | 1 | PM34 | PM33 | PM32 | PM31 | PM30 |

Address: FFF2CH After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM12 | PM127 | PM126 | PM125 | 1 | 1 | 1 | 1 | PM120 |

| PMmn | Pmn pin I/O mode selection<br>(m = 1, 3, 12; n = 0 to 7) |
|---|---|
| 0 | Output mode (output buffer on) |
| 1 | Input mode (output buffer off) |

**Remark** The figure shown above presents the format of port mode registers of the 100-pin products.

## 8.3 Operation

### 8.3.1 Items Common to Multiple Modes

(1) Count Sources

The count source selection method is the same in all modes. However, the external clock cannot be selected in PWM3 mode.

**Table 8-11. Count Source Selection**

| Count Source | Selection |
|---|---|
| $f_{CLK}$, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ | The FRQSEL4 bit in the user option byte (000C2H/020C2H) is cleared to 0 and the PLLDIV1 bit in the PLL control register (PLLCTL) is cleared to 0, or the SELPLLS bit in the PLL status register (PLLSTS) is cleared to 0 and the TRD_CKSEL bit in the clock select register (CKSEL) is cleared to 0.<br>A count source is selected by bits TCK2 to TCK0 in the TRDCRi register. |
| $f_{IH}$ | The MCM0 bit in the clock system control register (CKC) is cleared to 0, the SELPLL bit in the PLL control register (PLLCTL) is cleared to 0, the FRQSEL4 bit in the user option byte (000C2H/020C2H) is set to 1, and the TRD_CKSEL bit in the clock select register (CKSEL) is cleared to 0.<br>A count source is selected by bits TCK2 to TCK0 in the TRDCRi register. |
| $f_{PLL}$ | The SELPLL bit in the PLL control register (PLLCTL) is set to 1, the PLLDIV1 bit in the PLL control register (PLLCTL) is set to 1, the SELPLLS bit in the PLL status register (PLLSTS) is set to 1, and the TRD_CKSEL bit in the clock select register (CKSEL) is cleared to 0.<br>The count source is selected by bits TCK2 to TCK0 in the TRDCRi register. |
| $f_{SUB}$ | The SELLOSC bit in the clock select register (CKSEL) is cleared to 0 and the TRD_CKSEL bit in the clock select register (CKSEL) is set to 1.<br>A count source is selected by bits TCK2 to TCK0 in the TRDCRi register. |
| $f_{IL}$ | The SELLOSC bit in the clock select register (CKSEL) is set to 1 and the TRD_CKSEL bit in clock select register (CKSEL) is set to 1.<br>A count source is selected by bits TCK2 to TCK0 in the TRDCRi register. |
| External signal input to TRDCLK0 pin | The STCLK bit in the TRDFCR register is set to 1 (external clock input enabled).<br>Bits TCK2 to TCK0 in the TRDCRi register are set to 101B.<br>The active edge is selected by bits CKEG1 and CKEG0 in the TRDCRi register.<br>The PM bit in the PM register of the port used as a TRDCLK0 pin is set to 1 (input mode). |

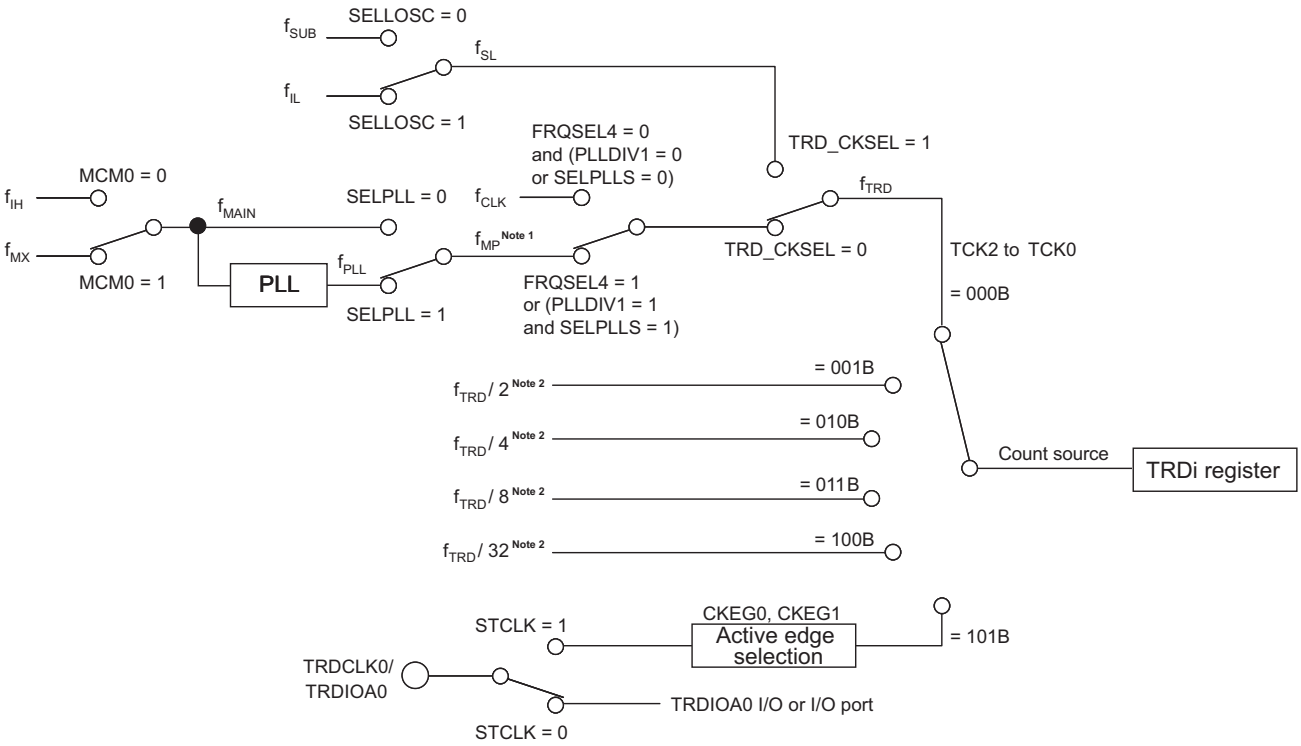
**Remark** i = 0 or 1

**Notes on count source setting is described below.**

**Notes on Count Source Setting of Timer RD**

| Count Source | Notes on Setting |
|---|---|
| $f_{CLK}$, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ | Set the TRD_CKSEL bit in the CKSEL register to 0 ($f_{CLK}/f_{MP}$ is selected), the FRQSEL4 bit in the user option byte (000C2H/020C2H) to 0 ($f_{IH} \leq 32$ MHz), and the PLLDIV1 bit in the PLLCTL register to 0 ($f_{PLL} \leq 32$ MHz).<br>Do not set $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, or $f_{CLK}/32$ when FRQSEL4 = 1.<br>The count sources cannot be used when SNOOZE status is output. |
| $f_{IH}$, $f_{PLL}$ | When $f_{IH}/f_{PLL}$ (64 MHz or 48 MHz) or $f_{PLL}$ (over 32 MHz) is used, set the CSS bit in the CKC register to 0 ($f_{CLK} = f_{MP}$ is selected).<br>When $f_{IH}/f_{PLL}$ (64 MHz or 48 MHz) is used, set bits MDIV2 to MDIV0 in the MDIV register to 001B ($f_{MP}/2$ is selected)<br>When $f_{IH}/f_{PLL}$ (64 MHz or 48 MHz) or $f_{PLL}$ (over 32 MHz) is used, set the TRD_CKSEL bit in the CKSEL register to 0 ($f_{CLK}/f_{MP}$ is selected).<br>When $f_{IH}/f_{PLL}$ (64 MHz or 48 MHz) or $f_{PLL}$ (over 32 MHz) is used, set the CSS bit, MDIV2 to MDIV0 bits, and TRD_CKSEL bit before setting the TRD0EN bit in the PER1 register.<br>Set the $f_{CLK}$ to the clock source which is same as the count source before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).<br>Setting the FRQSEL4 bit in the user option byte (000C2H/020C2H) to 1 ($f_{IH}$ = 64 MHz/48 MHz) and the PLLDIV bit in the PLLCTL register to 1 ($f_{PLL}$ > 32 MHz) is prohibited.<br>The count sources cannot be used when SNOOZE status is output. |
| $f_{SUB}$, $f_{IL}$ | When the CPU accesses the timer RD register, set the CSS bit in the CKC register to 1 ($f_{CLK} = f_{SL}$ is selected).<br>Set the $f_{CLK}$ to the clock source which is same as the count source before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).<br>The count sources should be set when SNOOZE status is output. |

**Figure 8-40. Count Source Block Diagram**




**Notes 1.** $f_{IH}$ cannot be selected when it is 64 MHz or 48 MHz. $f_{PLL}$ can be selected when it is over 32 MHz.

**2.** With this setting, select $f_{CLK}$ as the timer RD operating clock ($f_{TRD}$).

**Remark** i = 0 or 1

TCK0 to TCK2, CKEG0, CKEG1: Bits in TRDCRi register

STCLK: Bit in TRDFCR register

FREQSEL4: Bit in user option byte (000C2H/020C2H)

MCM0: Bit in CKC register

SELPLL: Bit in PLLCTL register

PLLDIV1: Bit in PLLCTL register

SELPLLS: Bit in PLLSTS register

SELLOSC: Bit in CKSEL register

TDC_CKSEL: Bit in CKSEL register

Set the pulse width of the external clock applied to the TRDCLK0 pin to three or more cycles of the timer RD operating clock ($f_{TRD}$).

(2) Buffer Operation

The TRDGRCi register (i = 0 or 1) can be used as the buffer register for the TRDGRAi register, and the TRDGRDi register can be used as the buffer register for the TRDGRBi register by means of bits TRDBFCi and TRDBFDi in the TRDMR register.

- TRDGRAi buffer register: TRDGRCi register
- TRDGRBi buffer register: TRDGRDi register

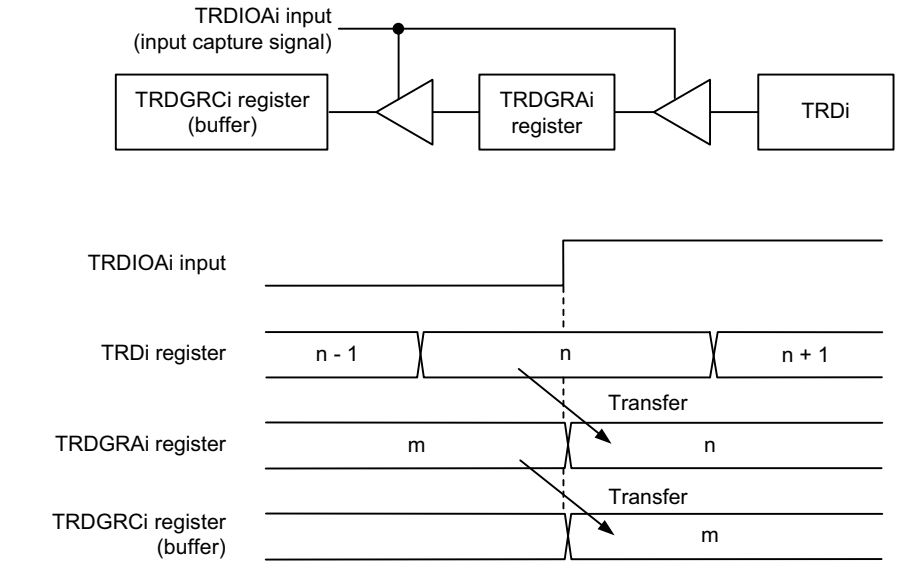
Buffer operation depends on the mode. Table 8-12 lists the Buffer Operation in Each Mode.

**Table 8-12. Buffer Operation in Each Mode**

| Function and Mode | | Transfer Timing | Transfer Register |
|---|---|---|---|
| Timer mode | Input capture function | Input capture signal input | Transfer content of TRDGRAi (TRDGRBi) register to buffer register |
| | Output compare function | Compare match with TRDi register and TRDGRAi (TRDGRBi) register | Transfer content of buffer register to TRDGRAi (TRDGRBi) register |
| | PWM function | | |
| Reset synchronous PWM mode | | Compare match with TRD0 register and TRDGRA0 register | Transfer content of buffer register to TRDGRAi (TRDGRBi) register |
| Complementary PWM mode | | Compare match with TRD0 register and TRDGRA0 register<br>TRD1 register underflow | Transfer content of buffer register to registers TRDGRB0, TRDGRA1, and TRDGRB1 |
| PWM3 mode | | Compare match with TRD0 register and TRDGRA0 register | Transfer content of buffer register to TRDGRAi (TRDGRBi) register |

**Remark** i = 0 or 1

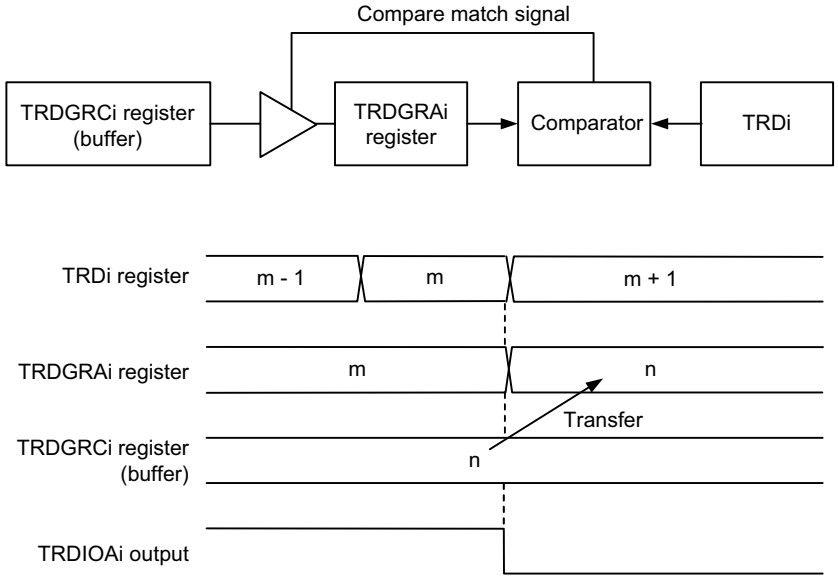
**Figure 8-41. Buffer Operation in Input Capture Function**




Remark
i = 0 or 1

The above diagram applies under the following conditions:
• The TRDBFCi bit in the TRDMR register is set to 1 (TRDGRCi register is buffer register for TRDGRAi register).
• Bits IOA2 to IOA0 in the TRDIORAi register are set to 100B (input capture at the rising edge).

**Figure 8-42. Buffer Operation in Output Compare Function**




Remark
i = 0 or 1

The above diagram applies under the following conditions:
• The TRDBFCi bit in the TRDMR register is set to 1 (TRDGRCi register is buffer register for TRDGRAi register).
• Bits IOA2 to IOA0 in the TRDIORAi register are set to 001B (low output by compare match).

Perform the following for the timer mode (input capture and output compare functions).
When using the TRDGRCi (i = 0 or 1) register as the buffer register for the TRDGRAi register

- Set the IOC3 bit in the TRDIORCi register to 1 (general register or buffer register).
- Set the IOC2 bit in the TRDIORCi register to the same value as the IOA2 bit in the TRDIORAi register.

When using the TRDGRDi register as the buffer register for the TRDGRBi register

- Set the IOD3 bit in the TRDIORCi register to 1 (general register or buffer register).
- Set the IOD2 bit in the TRDIORCi register to the same value as the IOB2 bit in the TRDIORAi register.

In the input capture function, when the TRDGRCi register or TRDGRDi register is used as a buffer register, the IMFC bit or IMFD bit in the TRDSRi register is set to 1 at the input edge of the TRDIOCi pin or TRDIODi pin.

When also using registers TRDGRCi and TRDGRDi as buffer registers for the output compare function, PWM function, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode, bits IMFC and IMFD in the TRDSRi register are set to 1 by a compare match with the TRDi register.

(3) Synchronous Operation

The TRD1 register is synchronized with the TRD0 register

- Synchronous preset

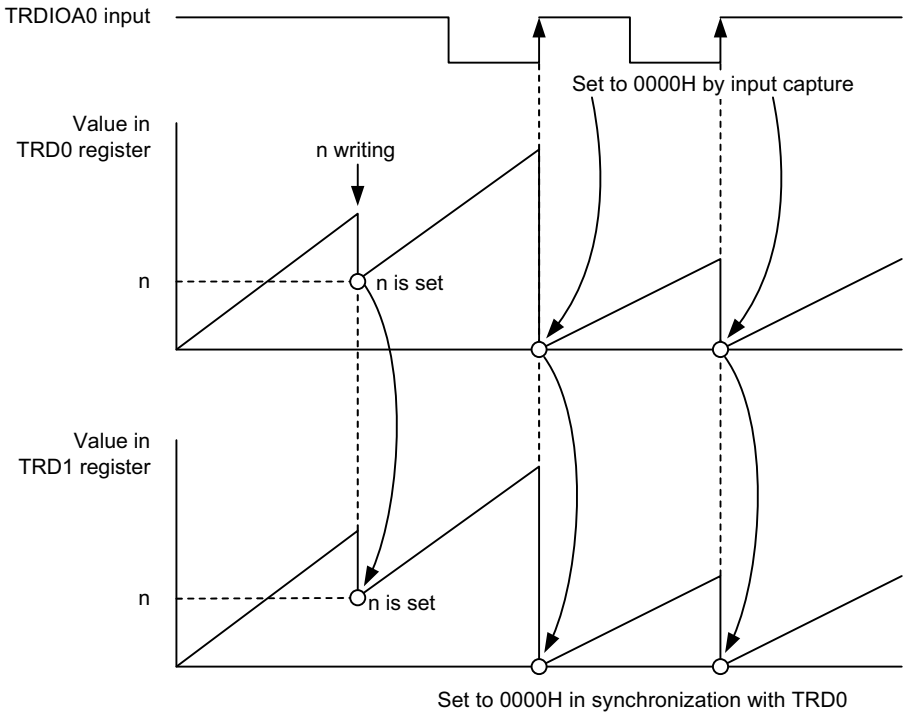
  When the TRDSYNC bit in the TRDMR register is set to 1 (synchronous operation), the data is written to both the TRD0 and TRD1 registers after writing to the TRDi register.

- Synchronous clear

  When the TRDSYNC bit is 1 and bits CCLR2 to CCLR0 in the TRDCR0 register are 011B (synchronous clear), the TRD0 register is set to 0000H at the same time as the TRD1 register is set to 0000H.

  Also, when the TRDSYNC bit is 1 and bits CCLR2 to CCLR0 are 011B (synchronous clear), the TRD1 register is set to 0000H at the same time as the TRD0 register is set to 0000H.

**Figure 8-43. Synchronous Operation**




The above diagram applies under the following conditions:

- The TRDSYNC bit in the TRDMR register is set to 1 (synchronous operation).
- Bits CCLR2 to CCLR0 in the TRDCR0 register are set to 001B (TRD0 is set to 0000H by input capture).
  Bits CCLR2 to CCLR0 in the TRDCR1 register are set to 011B (TRD1 is set to 0000H in synchronization with TRD0).
- Bits IOA2 to IOA0 in the TRDIORA0 register are set to 100B.
- Bits CMD1 to CMD0 in the TRDFCR register are set to 00B.
  The PWM 3 bit in the TRDFCR register is set to 1.

} (Input capture at the rising edge of TRDIOA0 input)

(4) Pulse Output Forced Cutoff

In the PWM function, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode, the TRDIOji output pin (i = 0 or 1, j = A, B, C, or D) can be forcibly set to an I/O port by the INTP0 pin input, and pulse output can be cut off.

The pins used for output in these functions or modes can function as the output pin of timer RD when the corresponding bit in the TRDOER1 register is set to 0 (timer RD output enabled). When the TRDPTO bit in the TRDOER2 register is 1 (pulse output forced cutoff signal input INTP0 enabled), the output pin used as a timer RD output port outputs the output value set by the DFCK1, DFCK0, PENB1, PENB0, DFD, DFC, DFB, or DFA bit in the TRDDF0 or TRDDF1 register.

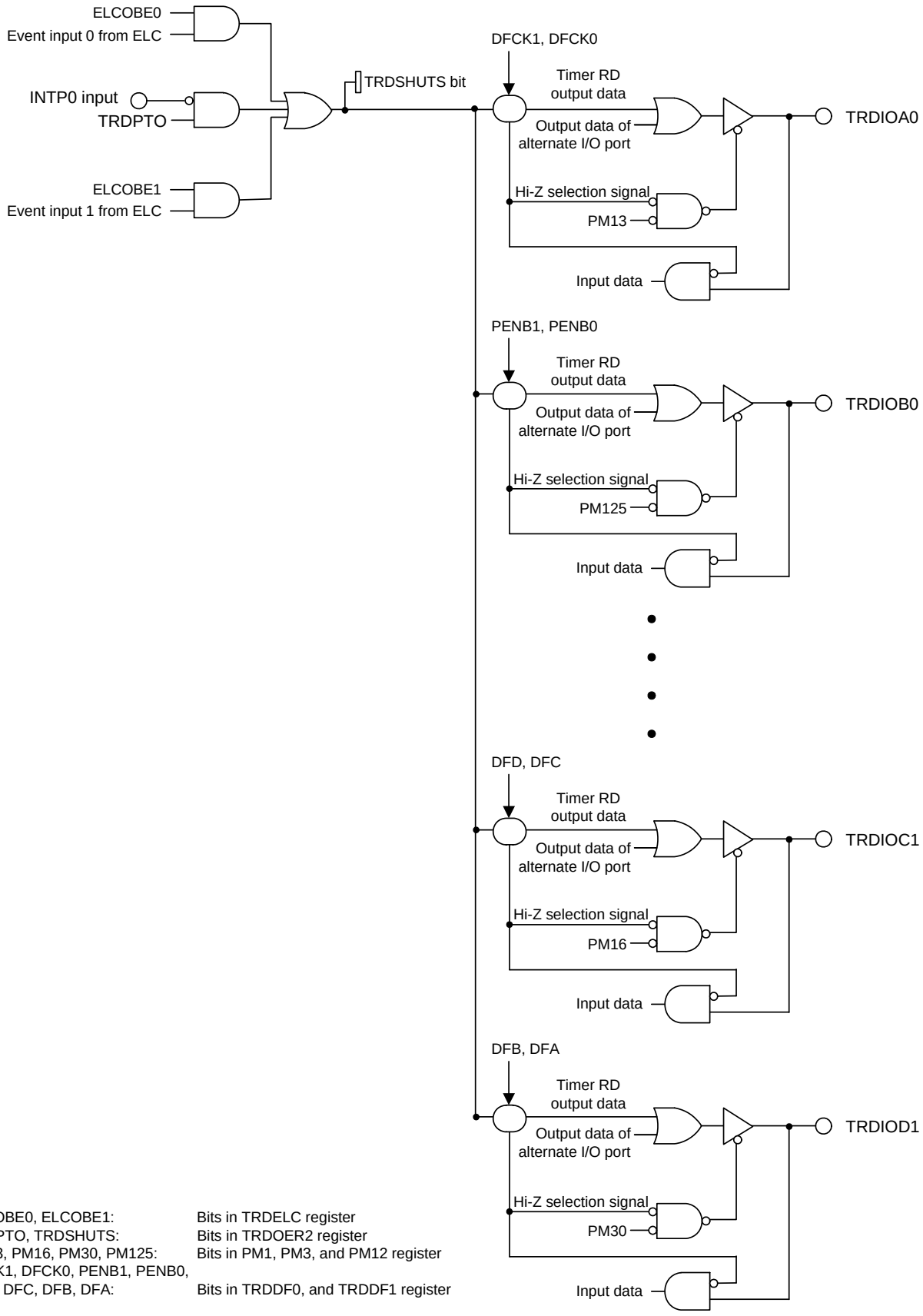
Make the following settings to use this function:

- Set the pin state when the pulse output is forcibly cut off (high impedance, low output, or high output) using TRDDFi.
- Refer to **8.3.1 (5) Event Input from Event Link Controller (ELC)** for details on pulse forced cutoff by ELC **Note** event input.
- When pulse output is forcibly cut out, the TRDSHUTS bit in the TRDOER2 register is set to 1. To suspend the forced cutoff of the pulse output, set the TRDSHUTS bit to 0 while the count is stopped (TSTARTi = 0).
- Set the TRDPTO bit in the TRDOER2 register to 1 (pulse output forced cutoff signal input INTP0 enabled).

**Note** The ELC is only available in the RL78/F14.

<R>

**Figure 8-44. Pulse Output Forced Cutoff**




| | |
|---|---|
| ELCOBE0, ELCOBE1: | Bits in TRDELC register |
| TRDPTO, TRDSHUTS: | Bits in TRDOER2 register |
| PM13, PM16, PM30, PM125: | Bits in PM1, PM3, and PM12 register |
| DFCK1, DFCK0, PENB1, PENB0, DFD, DFC, DFB, DFA: | Bits in TRDDF0, and TRDDF1 register |

(5) Event Input from Event Link Controller (ELC)

Timer RD performs two operations by event input from the ELC.

The ELC is only available in the RL78/F14.

**(a) Input capture operation D0/D1**

Timer RD performs input capture operation D0/D1 by event input from the ELC. The IMFD bit in the TRDSRi register is set to 1 at this time. To use this function, select the input capture function in timer mode and set the ELCICE0 or ELCICE1 bit in the TRDELC register to 1. This function is disabled in any other modes (for the output compare function in timer mode, PWM function, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode).

**(b) Pulse output forced cutoff operation** Note

The pulse output is forcibly cutoff by event input from the ELC. To use this function, select pulse output mode (PWM function, reset synchronous PWM mode, complementary PWM mode, or PWM3 mode) and set the ELCOBE0 or ELCOBE1 bit to 1. This function is disabled for the input capture function in timer mode.

**Note** The pulse output is cutoff during the low input period for forced cutoff from the INTP0 pin, but the pulse output is cutoff once by a single event input from the ELC for forced cutoff by the ELC event.

[Setting Procedure]

(1) Set timer RD as the ELC event link destination.

(2) Set bits ELCICEi (i = 0 or 1) and ELCOBEi (i = 0 or 1) to 1 in the TRDELC register.

(6) Event Output to Event Link Controller (ELC)/DTC/A/D Converter Trigger Select Register 0 (ADTRGS0)

Table 8-13 lists the Timer RD Modes and Event Output to ELC/DTC/ADTRGS0 register.

**Table 8-13. Timer RD Modes and Event Output to ELC/DTC/ADTRGS0 Register**

| Used Mode | Output Source | ELC Note 1 | DTC | ADTRGS0 register Note 2 |
|---|---|---|---|---|
| Input capture function | TRDIOA0 edge detection set by bits IOA1 and IOA0 in the TRDIORA0 register | Available | Available | - |
| | TRDIOB0 edge detection set by bits IOB1 and IOB0 in the TRDIORA0 register | Available | Available | Available |
| | TRDIOC0 edge detection set by bits IOC1 and IOC0 in the TRDIORC0 register | - | Available | - |
| | TRDIOD0 edge detection set by bits IOD1 and IOD0 in the TRDIORC0 register | - | Available | - |
| | TRDIOA1 edge detection set by bits IOA1 and IOA0 in the TRDIORA1 register | Available | Available | - |
| | TRDIOB1 edge detection set by bits IOB1 and IOB0 in the TRDIORA1 register | Available | Available | - |
| | TRDIOC1 edge detection set by bits IOC1 and IOC0 in the TRDIORC1 register | - | Available | - |
| | TRDIOD1 edge detection set by bits IOD1 and IOD0 in the TRDIORC1 register | - | Available | - |
| Output compare function, PWM function, reset synchronous PWM mode, complementary PWM mode, and PWM3 mode | Compare match between registers TRD0 and TRDGRA0 | Available | Available | - |
| | Compare match between registers TRD0 and TRDGRB0 | Available | Available | Available |
| | Compare match between registers TRD0 and TRDGRC0 | - | Available | - |
| | Compare match between registers TRD0 and TRDGRD0 | - | Available | - |
| | Compare match between registers TRD1 and TRDGRA1 | Available | Available | - |
| | Compare match between registers TRD1 and TRDGRB1 | Available | Available | - |
| | Compare match between registers TRD1 and TRDGRC1 | - | Available | - |
| | Compare match between registers TRD1 and TRDGRD1 | - | Available | - |
| Complementary PWM mode | TRD1 register underflow | Available | - | - |

**Notes 1.** The ELC is only available in the RL78/F14.

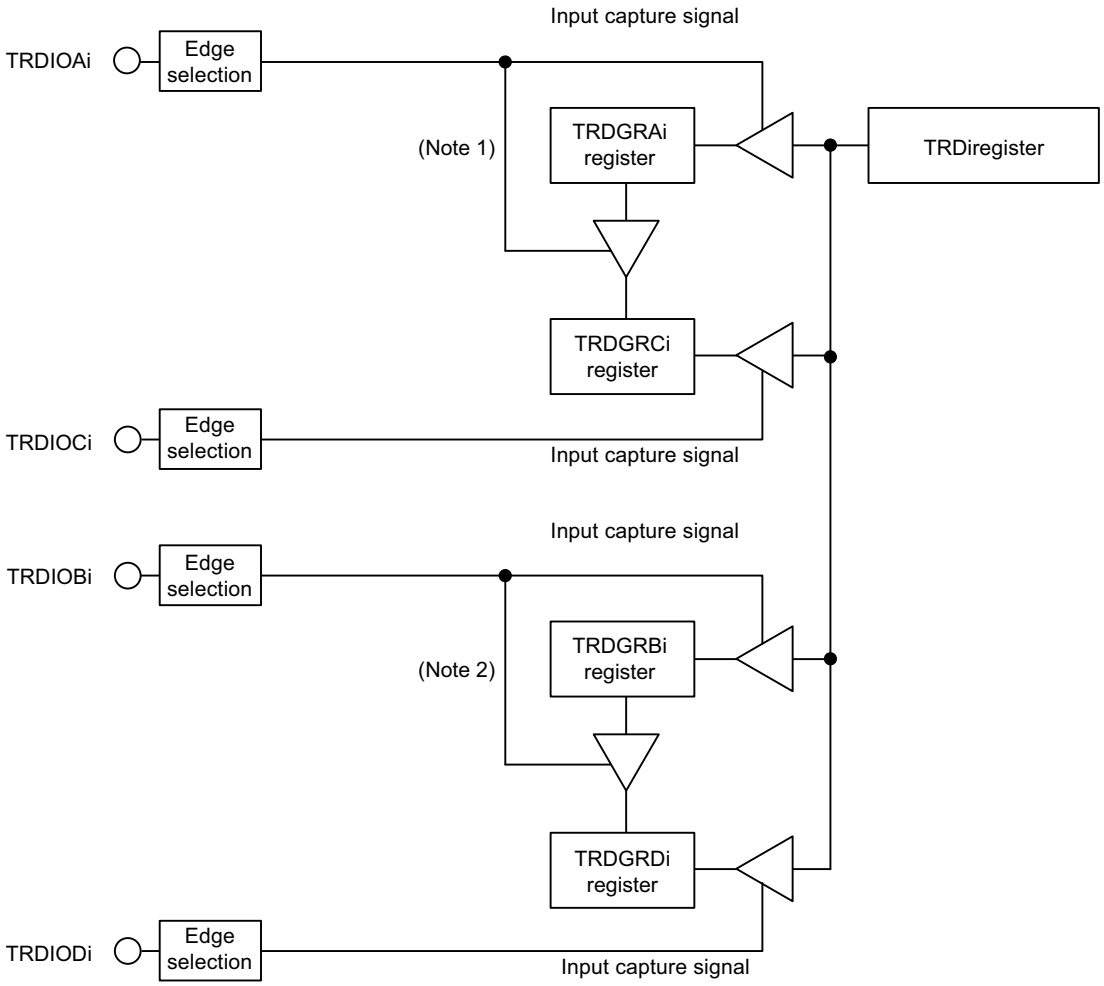
**2.** The A/D converter trigger select register 0 (ADTRGS0) is only available in the RL78/F13 (LIN incorporated) and RL78/F13 (CAN and LIN incorporated).

### 8.3.2 Input Capture Function

The input capture function measures the external signal width and period. The content of the TRDi register (counter) is transferred to the TRDGRji register as a trigger of the TRDIOji pin (i = 0 or 1, j = A, B, C, or D) external signal (input capture). Since this function is enabled with a combination of the TRDIOji pin and TRDGRji register, the input capture function, or any other mode or function, can be selected for each individual pin.

Figure 8-45 shows the Block Diagram of Input Capture Function, Table 8-14 lists the Input Capture Function Specifications, and Figure 8-46 shows an Operation Example of Input Capture Function.

**Figure 8-45. Block Diagram of Input Capture Function**




Remark
i = 0 or 1

Notes:
1. When the TRDBFCi bit in the TRDMR register is set to 1 (TRDGRCi register is buffer register for TRDGRAi register).
2. When the TRDBFDi bit in the TRDMR register is set to 1 (TRDGRDi register is buffer register for TRDGRBi register).

**Table 8-14. Input Capture Function Specifications**

| Item | Specification |
|---|---|
| Count sources Note 1 | $f_{CLK}$, $f_{PLL}$, $f_{IH}$, $f_{SUB}$, $f_{PL}$<br>External signal input to the TRDCLK0 pin (active edge selected by a program) |
| Count operations | Increment |
| Count period | When bits CCLR2 to CCLR0 in the TRDCRi register are set to 000B (free-running operation).<br>1/fk × 65536 fk: Frequency of count source |
| Count start condition | 1 (count starts) is written to the TSTARTi bit in the TRDSTR register. |
| Count stop condition | 0 (count stops) is written to the TSTARTi bit in the TRDSTR register when the CSELi bit in the TRDSTR register is set to 1. |
| Interrupt request generation timing | • Input capture (active edge of TRDIOji input)<br>• TRDi register overflow |
| TRDIOA0 pin function | I/O port, input-capture input, or TRDCLK (external clock) input |
| TRDIOB0, TRDIOC0, TRDIOD0, TRDIOA1 to TRDIOD1 pin function | I/O port or input-capture input (selectable for each pin) |
| INTP0 pin function | Not used (port or INTP0 interrupt input) |
| Read from timer | The count value can be read by reading the TRDi register. |
| Write to timer | • When the TRDSYNC bit in the TRDMR register is 0 (timer RD0 and timer RD1 operate independently).<br>Data can be written to the TRDi register.<br>• When the TRDSYNC bit in the TRDMR register is 1 (timer RD0 and timer RD1 operate synchronously).<br>Data can be written to both the TRD0 and TRD1 registers by writing to the TRDi register. |
| Selectable functions | • Input-capture input pin selection<br>Either one pin or multiple pins of TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi.<br>• Input-capture input active edge selection<br>Rising edge, falling edge, or both rising and falling edges<br>• Timing for setting the TRDi register to 0000H.<br>At overflow or input capture<br>• Buffer operation (see **8. 3. 1 (2) Buffer Operation**)<br>• Synchronous operation (see **8. 3. 1 (3) Synchronous Operation**)<br>• Digital filter.<br>The TRDIOji input is sampled, and when the sampled input level match three times, that level is determined.<br>• Input capture operation by event input from event link controller (ELC). Note 2 |

**Notes 1.** When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

**2.** The ELC is only available in the RL78/F14.

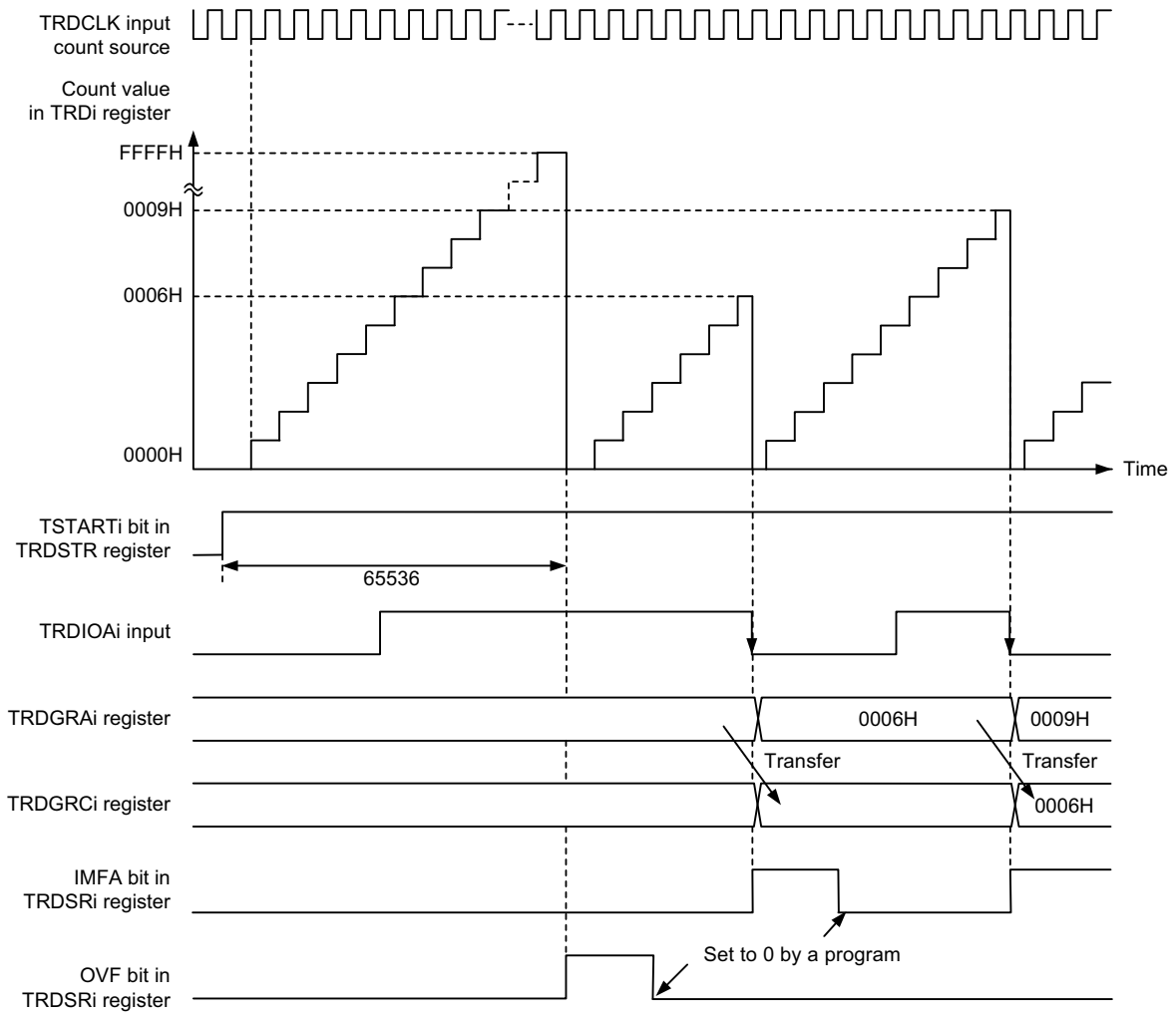
**Remark** i = 0 or 1, j = A, B, C, or D

(1) Operation Example

By setting bits CCLR0 to CCLR2 in the TRDCRi register (i = 0 or 1), the timer RDi counter value is reset by an input capture/compare match. Figure 8-46 shows an operation example with bits CCLR2 to CCLR0 set to 001B.

If the input capture operation has been set to clear the count during operation and is performed when the timer count value is FFFFH, depending on the timing between the count source and input capture operation interrupt flags bits IMFA to IMFD and OVF in the TRDSRi register may be set to 1 simultaneously.

**Figure 8-46. Operation Example of Input Capture Function**




Remark
i = 0 or 1

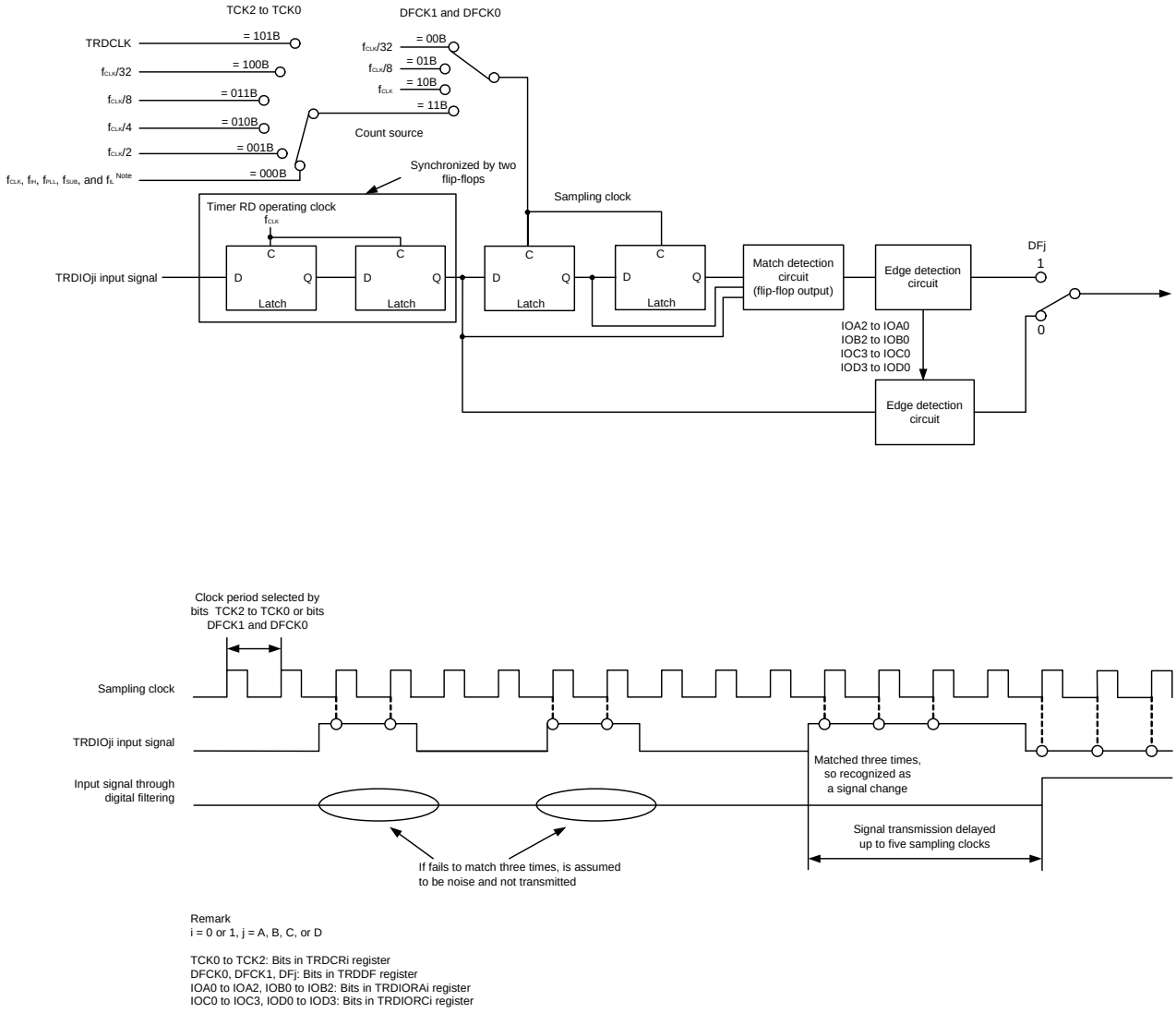
The above diagram applies under the following conditions:
Bits CCLR2 to CCLR0 in the TRDCRi register are set to 001B (TRDi register is set to 0000H by TRDGRAi register input capture).
Bits TCK2 to TCK0 in the TRDCRi register are set to 101B (TRDCLK input for the count source).
Bits CKEG1 and CKEG0 in the TRDCRi register are set to 01B (count at the falling edge for the count source).
Bits IOA2 to IOA0 in the TRDIORAi register are set to 101B (input capture at the falling edge of TRDIOAi input).
The TRDBFCi bit in the TRDMR register is set to 1 (TRDGRCi register is buffer register for TRDGRAi register).

(2) Digital Filter

The TRDIOji input (i = 0 or 1, j = A, B, C, or D) is sampled, and when the sampled input level matches three times, its level is determined. Select the digital filter function and sampling clock using the TRDDFi register.

Figure 8-47 shows the Block Diagram of Digital Filter.

**Figure 8-47. Block Diagram of Digital Filter**




Remark
i = 0 or 1, j = A, B, C, or D

TCK0 to TCK2: Bits in TRDCRi register
DFCK0, DFCK1, DFj: Bits in TRDDF register
IOA0 to IOA2, IOB0 to IOB2: Bits in TRDIORAi register
IOC0 to IOC3, IOD0 to IOD3: Bits in TRDIORCi register

**Notes 1.** As the timer RD operating clock ($f_{TRD}$), $f_{CLK}$ is selected when FRQSEL4 = 0 in the user option byte (000C2H/020C2H), (PLLDIV1 = 0 or SELPLLS = 0), and TRD_CKSEL = 0. $f_{IH}$ is selected when FRQSEL4 = 1 and TRD_CKSEL = 0. $f_{PLL}$ is selected when (PLLDIV1 = 1 and SELPLLS = 1) and TRD_CKSEL = 0. $f_{SUB}$ is selected when SELLOSC = 0 and TRD_CKSEL = 1. $f_{IL}$ is selected when SELLOSC = 1 and TRD_CKSEL = 1. For details, see **Figure 8-40**.

When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

**2.** With this setting, select $f_{CLK}$ as the timer RD operating clock ($f_{TRD}$).

### 8.3.3 Output Compare Function

This function detects matches (compare match) between the content of the TRDGRji register (j = A, B, C, or D) and the content of the TRDi register (counter) (i = 0 or 1). When the contents match, an arbitrary level is output from the TRDIOji pin. Since this function is enabled with a combination of the TRDIOji pin and TRDGRji register, the output compare function, or any other mode or function, can be selected for each individual pin.

Figure 8-48 shows the Block Diagram of Output Compare Function, Table 8-13 lists the Output Compare Function Specifications, and Figure 8-49 shows an Operation Example of Output Compare Function.

**Figure 8-48. Block Diagram of Output Compare Function**

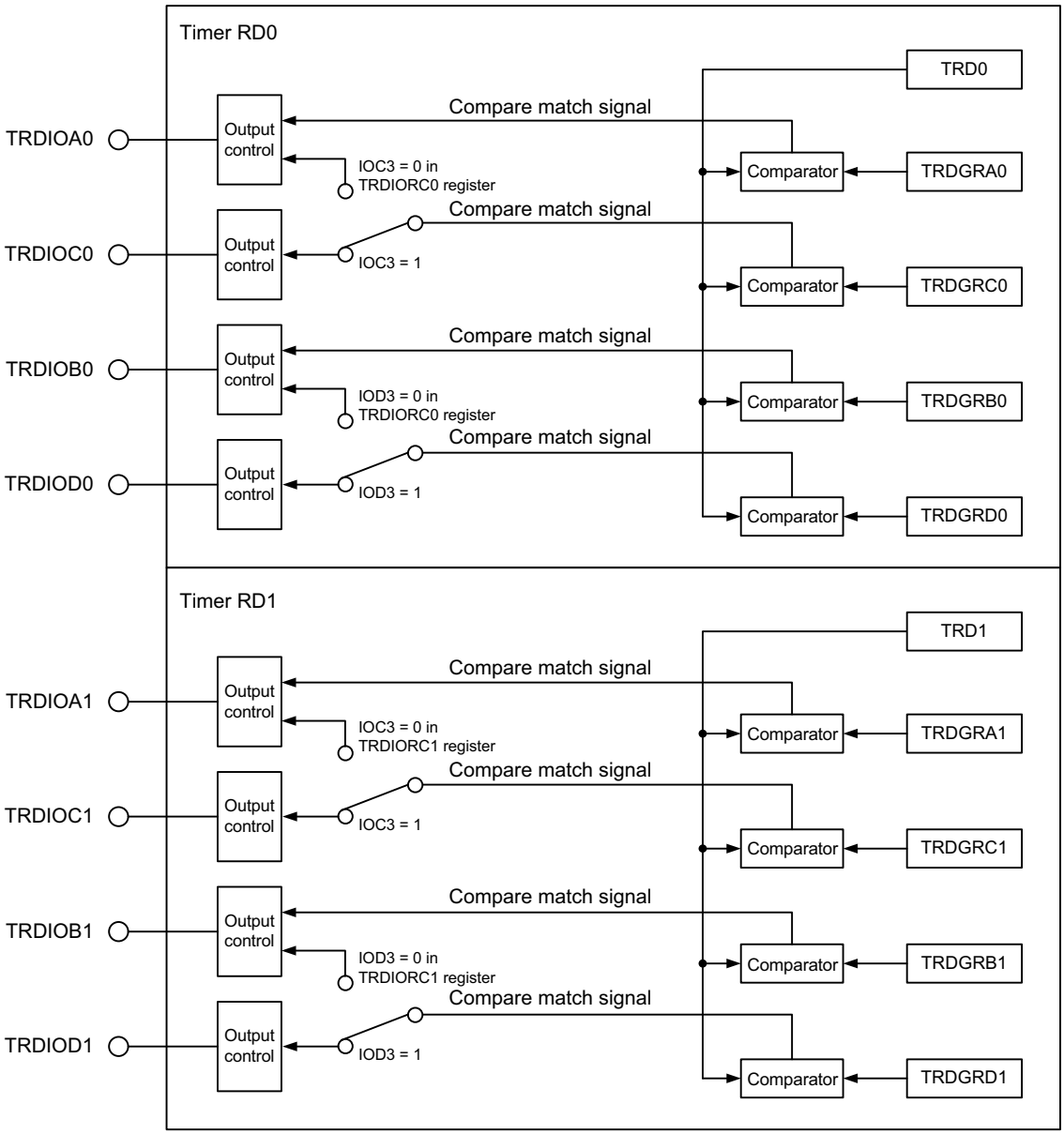


**Table 8-15. Output Compare Function Specifications**

| Item | Specification |
|---|---|
| Count sources Note | $f_{CLK}$, $f_{PLL}$, $f_{IH}$, $f_{SUB}$, $f_{IL}$<br>External signal input to the TRDCLK0 pin (active edge selected by a program) |
| Count operations | Increment |
| Count period | • When bits CCLR2 to CCLR0 in the TRDCRi register are set to 000B (free-running operation).<br>1/fk × 65536 fk: Frequency of count source<br>• When bits CCLR1 and CCLR0 in the TRDCRi register are set to 01B or 10B (TRDi register is set to 0000H at compare match with TRDGRji register).<br>1/fk × (n + 1)<br>n: Value set in the TRDGRji register |
| Waveform output timing | Compare match (contents of registers TRDi and TRDGRji match) |
| Count start condition | 1 (count starts) is written to the TSTARTi bit in the TRDSTR register. |
| Count stop conditions | • 0 (count stops) is written to the TSTARTi bit in the TRDSTR register when the CSELi bit in the TRDSTR register is set to 1.<br>The output compare output pin holds the output level before the count stops.<br>• When the CSELi bit in the TRDSTR register is set to 0, the count stops at the compare match with the TRDGRAi register.<br>The output compare output pin holds the level after output change by compare match. |
| Interrupt request generation timing | • Compare match (contents of registers TRDi and TRDGRji match)<br>• TRDi register overflow |
| TRDIOA0 pin function | I/O port, output-compare output, or TRDCLK (external clock) input |
| TRDIOB0, TRDIOC0, TRDIOD0, TRDIOA1 to TRDIOD1 pin function | I/O port or output-compare output (selectable for each pin) |
| INTP0 pin function | Port or INTP0 interrupt input |
| Read from timer | The count value can be read by reading the TRDi register. |
| Write to timer | • When the TRDSYNC bit in the TRDMR register is set to 0 (timer RD0 and timer RD1 operate independently). Data can be written to the TRDi register.<br>• When the TRDSYNC bit in the TRDMR register is set to 1 (timer RD0 and timer RD1 operate synchronously). Data can be written to both the TRD0 and TRD1 registers by writing to the TRDi register. |
| Selectable functions | • Output-compare output pin selection<br>Either one pin or multiple pins of TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi.<br>• Output level selection at compare match<br>Low output, high output, or inverted output level<br>• Initial output level selection<br>The level can be set for the period from the count start to the compare match.<br>• Timing for setting the TRDi register to 0000H<br>Overflow or compare match in the TRDGRAi register<br>• Buffer operation (see **8. 3. 1 (2) Buffer Operation**)<br>• Synchronous operation (see **8. 3. 1 (3) Synchronous Operation**)<br>• Changing output pins for registers TRDGRCi and TRDGRDi<br>The TRDGRCi register can be used as output control of the TRDIOAi pin and the TRDGRDi register can be used as output control of the TRDIOBi pin.<br>• Pulse output forced cutoff signal input (see **8. 3. 1 (4) Pulse Output Forced Cutoff**)<br>• Timer RD can be used as the internal timer without output. |

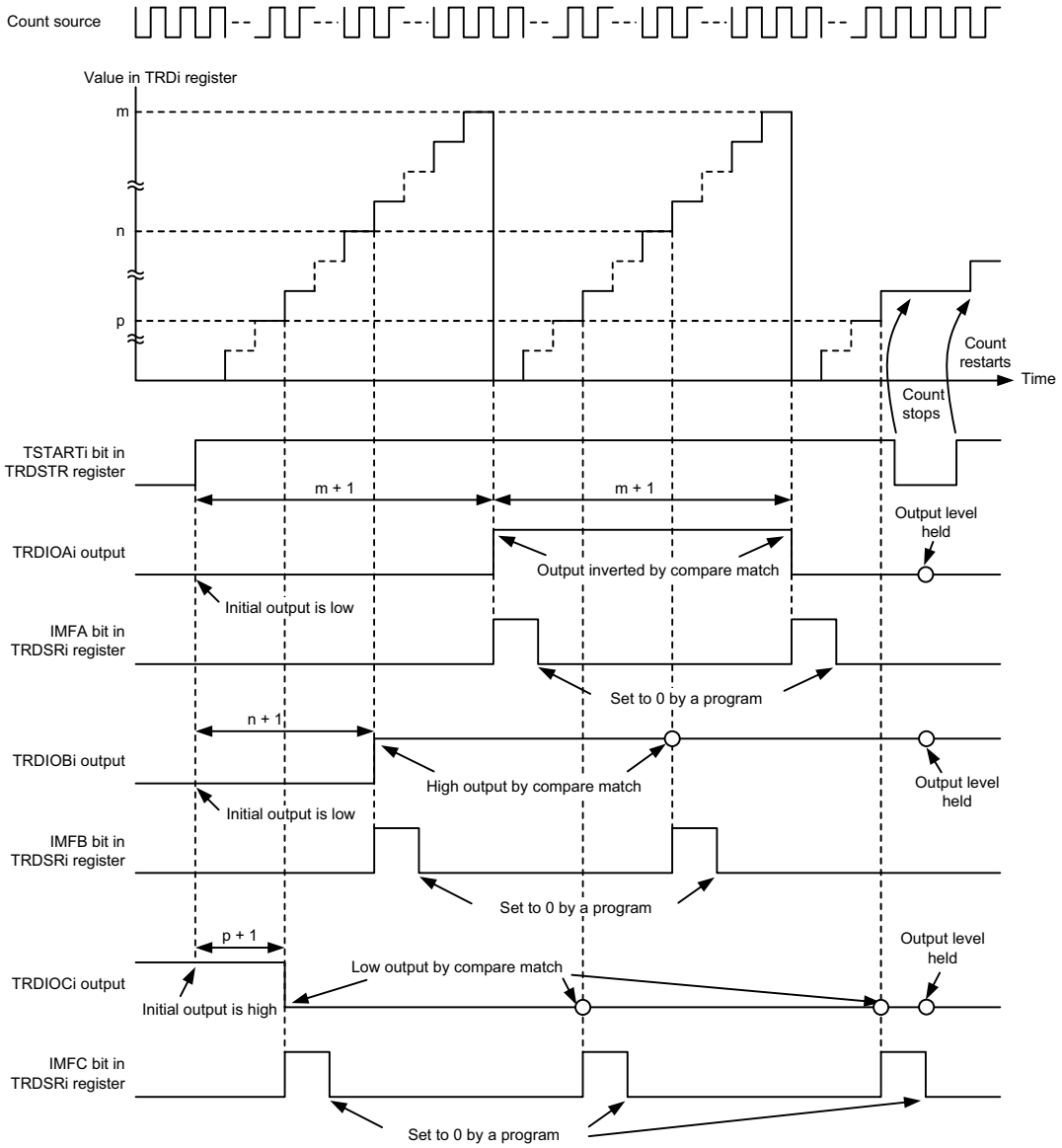
**Note** When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

**Remark** i = 0 or 1, j = A, B, C, or D

(1) Operation Example

By setting bits CCLR0 to CCLR2 in the TRDCRi register (i = 0 or 1), the timer RDi counter value is reset by an input capture/compare match. If the expected compare value is FFFFH at this time, FFFFH changes to 0000H, same as the overflow operation, and the overflow flag is set to 1.

**Figure 8-49. Operation Example of Output Compare Function**




Remark
i = 0 or 1
M: Value set in TRDGRAi register
n: Value set in TRDGRBi register
p: Value set in TRDGRCi register

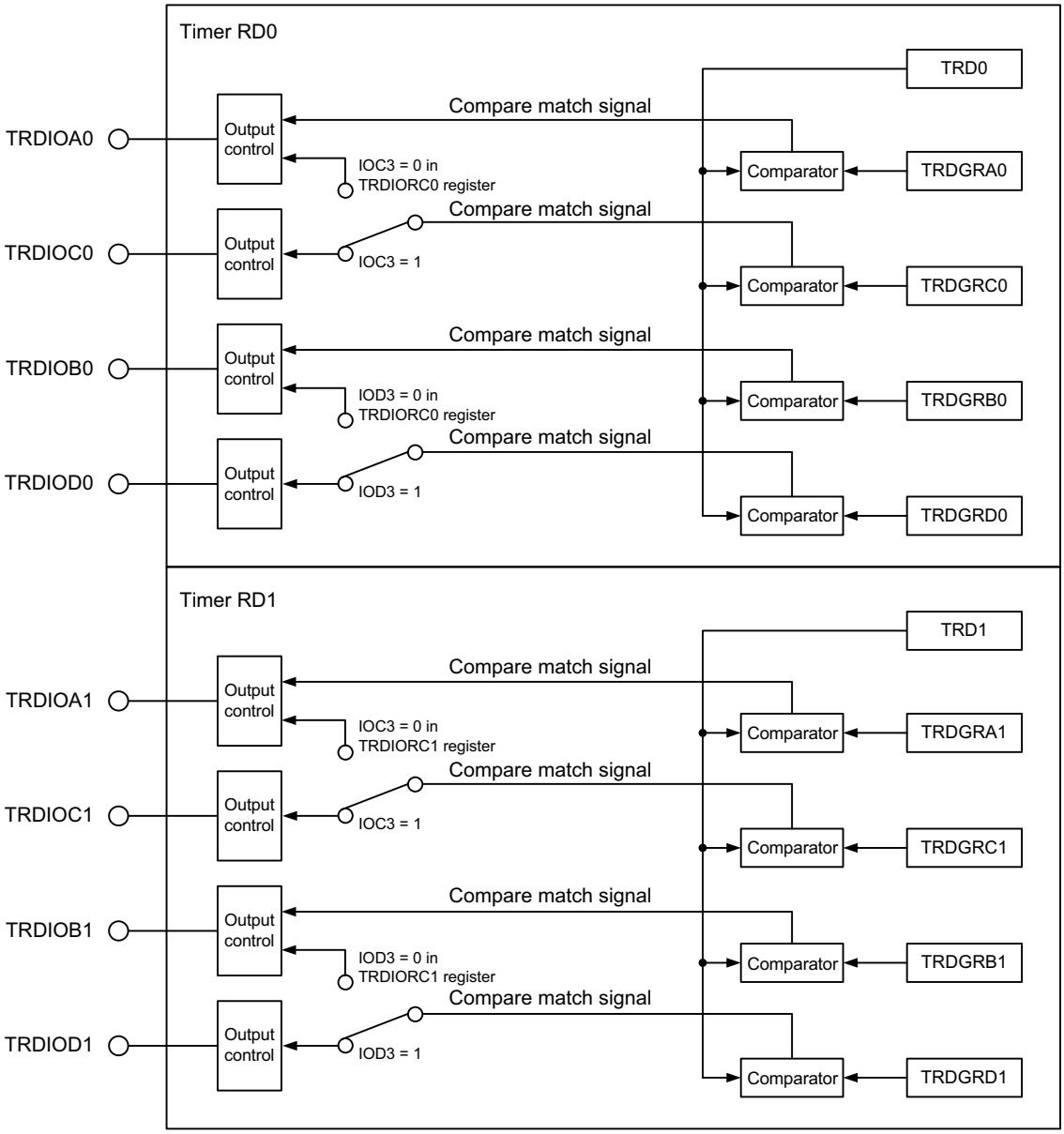
The above diagram applies under the following conditions :
The CSELi bit in the TRDSTR register is set to 1 (TRDi is not stopped by compare match).
Bits TRDBFCi and TRDBFDi in the TRDMR register are set to 0 (TRDGRCi and TRDGRDi do not operate as buffers).
Bits EAi, EBi, and ECi in the TRDOER1 register are set to 0 (TRDIOAi, TRDIOBi and TRDIOCi output enabled).
Bits CCLR2 to CCLR0 in the TRDCRi register are set to 001B (TRDi is set to 0000H by compare match with TRDGRAi).
Bits TOAi and TOBi in the TRDOCR register is set to 0 (initial output is low until compare match), the TOCi bit is set to 1 (initial output is high until compare match).
Bits IOA2 to IOA0 in the TRDIORAi register are set to 011B (TRDIOAi output inverted at TRDGRAi compare match).
Bits IOB2 to IOB0 in the TRDIORAi register are set to 010B (TRDIOBi high output at TRDGRBi compare match).
Bits IOC3 to IOC0 in the TRDIORCi register are set to 1001B (TRDIOCi low output at TRDGRCi register compare match).
Bits IOD3 to IOD0 in the TRDIORCi register are set to 1000B (TRDGRDi register does not control TRDIOBi pin output. Pin output by compare match is disabled).

(2) Changing Output Pins in Registers TRDGRCi (i = 0 or 1) and TRDGRDi

The TRDGRCi register can be used for output control of the TRDIOAi pin, and the TRDGRDi register can be used for output control of the TRDIOBi pin. Therefore, each pin output can be controlled as follows:

- TRDIOAi output is controlled by the values in registers TRDGRAi and TRDGRCi.
- TRDIOBi output is controlled by the values in registers TRDGRBi and TRDGRDi.

**Figure 8-50. Changing Output Pins in Registers TRDGRCi and TRDGRDi**

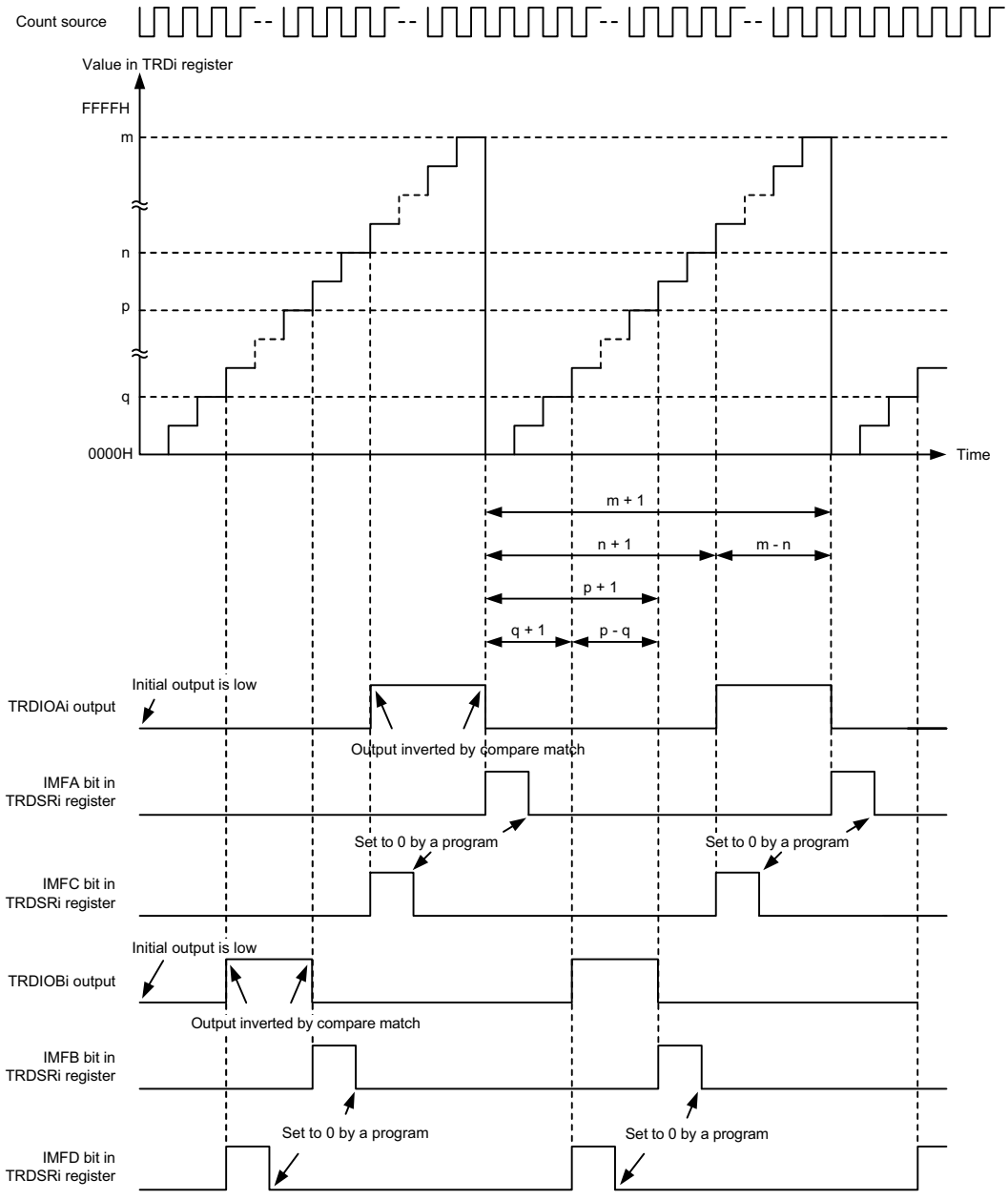



Change output pins in registers TRDGRCi and TRDGRDi as follows:

- Select 0 (TRDGRji register output pin is changed) using the IOj3 (j = C or D) bit in the TRDIORCi register.
- Set the TRDBFji bit in the TRDMR register to 0 (general register).
- Set different values in registers TRDGRCi and TRDGRAi. Also, set different values in registers TRDGRDi and TRDGRBi.

Figure 8-51 shows an Operation Example When TRDGRCi Register is Used for Output Control of TRDIOAi Pin and TRDGRDi Register is Used for Output Control of TRDIOBi Pin.

**Figure 8-51. Operation Example When TRDGRCi Register is Used for Output Control of TRDIOAi Pin and TRDGRDi Register is Used for Output Control of TRDIOBi Pin**




Remark
i = 0 or 1
m: Value set in TRDGRAi register
n: Value set in TRDGRCi register
p: Value set in TRDGRBi register
q: Value set in TRDGRDi register

The above diagram applies under the following conditions :
The CSELi bit in the TRDSTR register is set to 1 (TRDi register is not stopped by compare match).
Bits TRDBFCi and TRDBFDi in the TRDMR register are set to 0 (TRDGRCi and TRDGRDi do not operate as buffers).
Bits EAi and EBi in the TRDOER1 register are set to 0 (TRDIOAi and TRDIOBi output enabled).
Bits CCLR2 to CCLR0 in the TRDCRi register are set to 001B (TRDi is set to 0000H by compare match with TRDGRAi).
Bits TOAi and TOBi in the TRDOCR register are set to 0 (initial output is low until compare match).
Bits IOA2 to IOA0 in the TRDIORAi register are set to 011B (TRDIOAi output inverted at TRDGRAi compare match).
Bits IOB2 to IOB0 in the TRDIORAi register are set to 011B (TRDIOBi output inverted at TRDGRBi compare match).
Bits IOC3 to IOC0 in the TRDIORCi register are set to 0011B (TRDIOAi output inverted at TRDGRCi compare match).
Bits IOD3 to IOD0 in the TRDIORCi register are set to 0011B (TRDIOBi output inverted at TRDGRDi compare match).

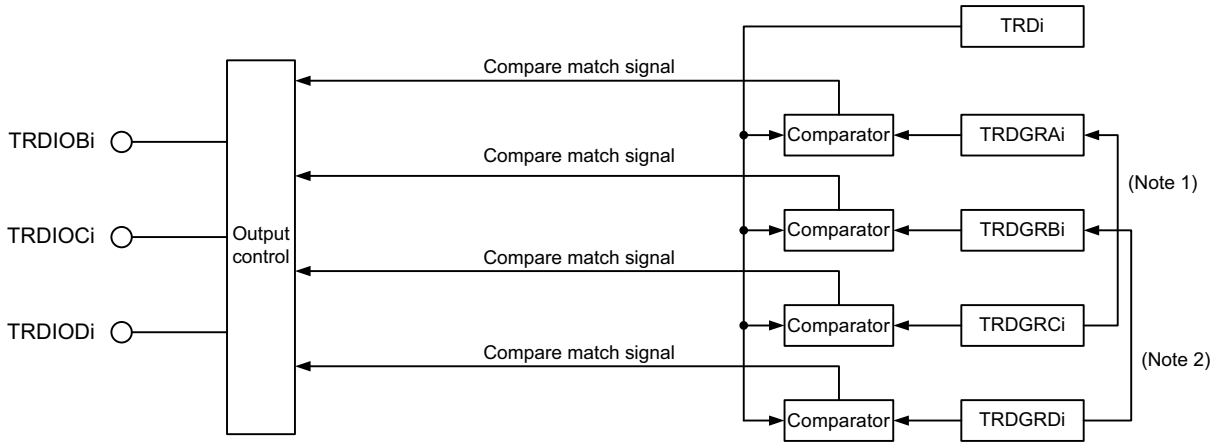
### 8.3.4 PWM Function

In PWM function, a PWM waveform is output. Up to three PWM waveforms with the same period can be output by timer RDi (i = 0 or 1). Also, up to six PWM waveforms with the same period can be output by synchronizing timer RD0 and timer RD1.

Since this mode functions by a combination of the TRDIOji pin (i = 0 or 1, j = B, C, or D) and TRDGRji register, PWM function, or any other mode or function, can be selected for each individual pin. (However, since the TRDGRAi register is used when using any pin for PWM function, the TRDGRAi register cannot be used for other modes.)

Figure 8-52 shows the Block Diagram of PWM function, Table 8-16 lists the PWM Function Specifications, and Figure 8-53 and Figure 8-54 show Operation Examples in PWM Function.

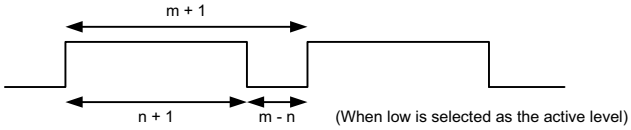
Figure **8-52. Block Diagram of PWM Function**




Remark
i = 0 or 1

Notes:
1. When the TRDBFCi bit in the TRDMR register is set to 1 (TRDGRCi register is buffer register for TRDGRAi register).
2. When the TRDBFDi bit in the TRDMR register is set to 1 (TRDGRDi register is buffer register for TRDGRBi register).

**Table 8-16. PWM Mode Specifications**

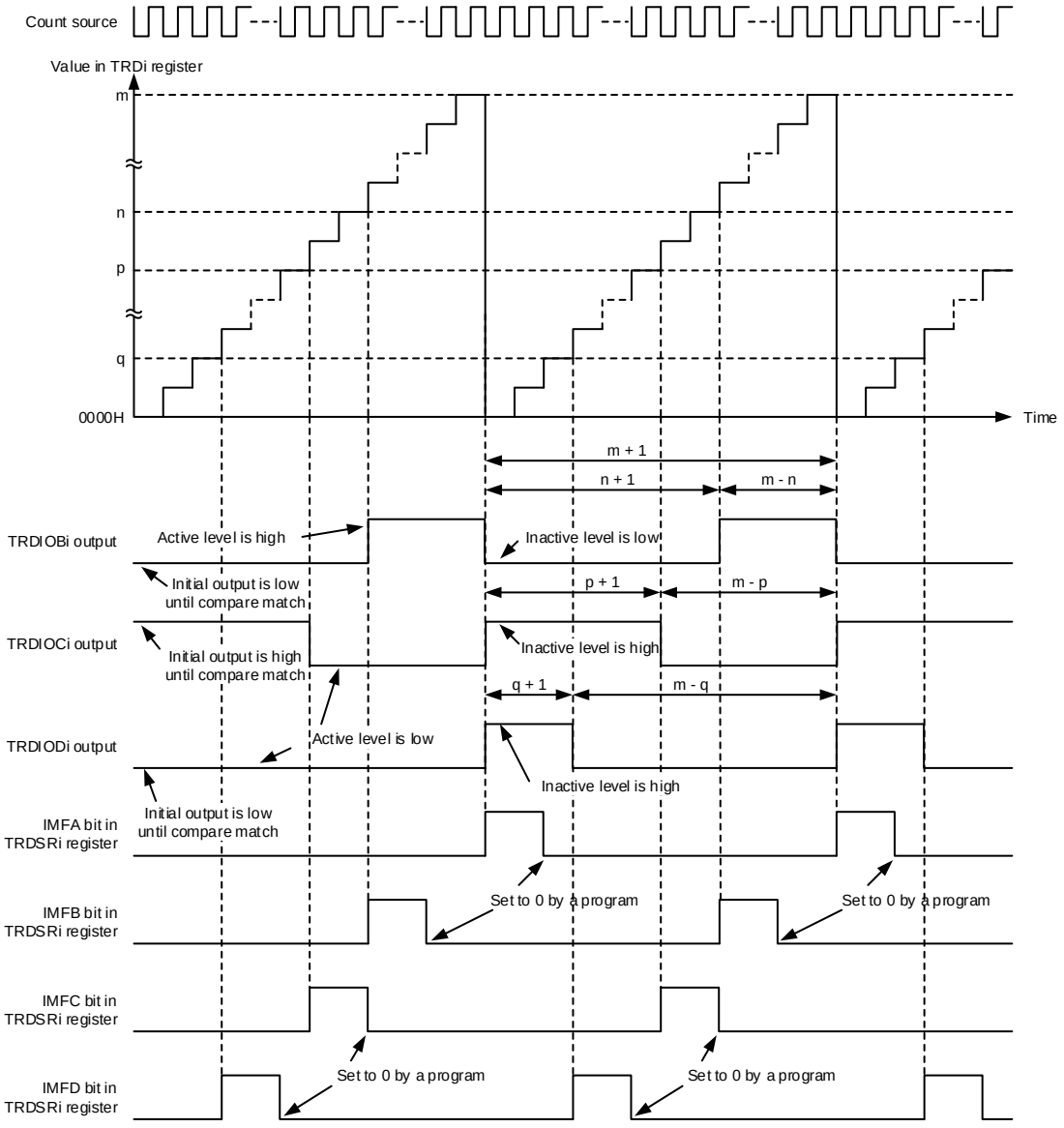
| Item | Specification |
|---|---|
| Count sources Note | $f_{CLK}$, $f_{PLL}$, $f_{IH}$, $f_{SUB}$, $f_{IL}$<br>External signal input to the TRDCLK0 pin (active edge selected by a program) |
| Count operations | Increment |
| PWM waveform | PWM period: 1/fk x (m + 1)<br>Active level width: 1/fk x (m - n)<br>Inactive level width: 1/fk x (n + 1)<br>fk: Frequency of count source<br>m: Value set in the TRDGRAi register<br>n: Value set in the TRDGRji register<br>  |
| Count start condition | 1 (count starts) is written to the TSTARTi bit in the TRDSTR register. |
| Count stop conditions | • 0 (count stops) is written to the TSTARTi bit in the TRDSTR register when the CSELi bit in the TRDSTR register is set to 1.<br>The PWM output pin holds the output level before the count stops.<br>• When the CSELi bit in the TRDSTR register is set to 0, the count stops at the compare match with the TRDGRAi register.<br>The PWM output pin holds the level after output change by compare match. |
| Interrupt request generation timing | • Compare match (content of the TRDi register matches content of the TRDGRhi register)<br>• TRDi register overflow |
| TRDIOA0 pin function | I/O port or TRDCLK (external clock) input |
| TRDIOA1 pin function | I/O port |
| TRDIOB0, TRDIOC0, TRDIOD0, TRDIOB1, TRDIOC1, TRDIOD1 pin function | I/O port or pulse output (selectable for each pin) |
| INTP0 pin function | Pulse output forced cutoff signal input (port or INTP0 interrupt input) |
| Read from timer | The count value can be read by reading the TRDi register. |
| Write to timer | The value can be written to the TRDi register. |
| Selectable functions | • One to three PWM output pins selectable with timer RDi<br>Either one pin or multiple pins of TRDIOBi, TRDIOCi, and TRDIODi.<br>• Active level selectable for each pin.<br>• Initial output level selectable for each pin.<br>• Synchronous operation (see **8. 3. 1 (3) Synchronous Operation**)<br>• Buffer operation (see **8. 3. 1 (2) Buffer Operation**)<br>• Pulse output forced cutoff signal input (see **8. 3. 1 (4) Pulse Output Forced Cutoff**) |

**Note** When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

**Remark** i = 0 or 1, j = B, C, or D, h = A, B, C, or D

(1) Operation Example

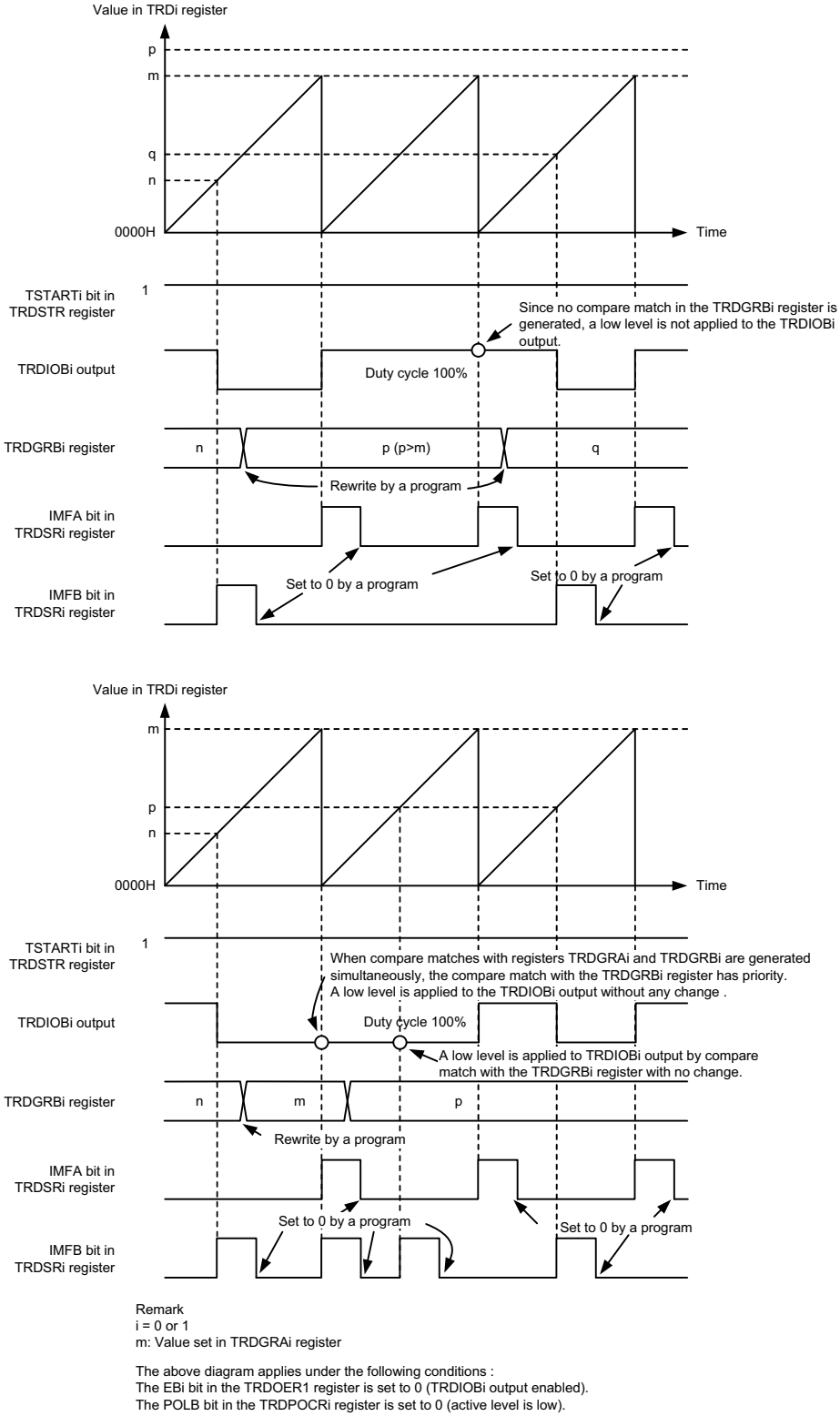
**Figure 8-53. Operation Example in PWM Function**




Remark
i = 0 or 1
m: Value set in TRDGRAi register
n: Value set in TRDGRBi register
p: Value set in TRDGRCi register
q: Value set in TRDGRDi register

The above diagram applies under the following conditions:
Bits TRDBFCi and TRDBFDi in the TRDMR register are set to 0 (TRDGRCi and TRDGRDi do not operate as buffers).
Bits EBi, ECi and EDi in the TRDOER1 register are set to 0 (TRDIOBi, TRDIOCi and TRDIODi output enabled).
Bits TOBi and TOCi in the TRDOCR register are set to 0 (inactive level), the TODi bit is set to 1 (active level).
The POLB bit in the TRDPOCRi register is set to 1 (active level is high), bits POLC and POLD are set to 0 (active level is low).

**Figure 8-54. Operation Example in PWM Function (Duty Cycle 0%, Duty Cycle 100%)**




Remark
i = 0 or 1
m: Value set in TRDGRAi register

The above diagram applies under the following conditions :
The EBi bit in the TRDOER1 register is set to 0 (TRDIOBi output enabled).
The POLB bit in the TRDPOCRi register is set to 0 (active level is low).

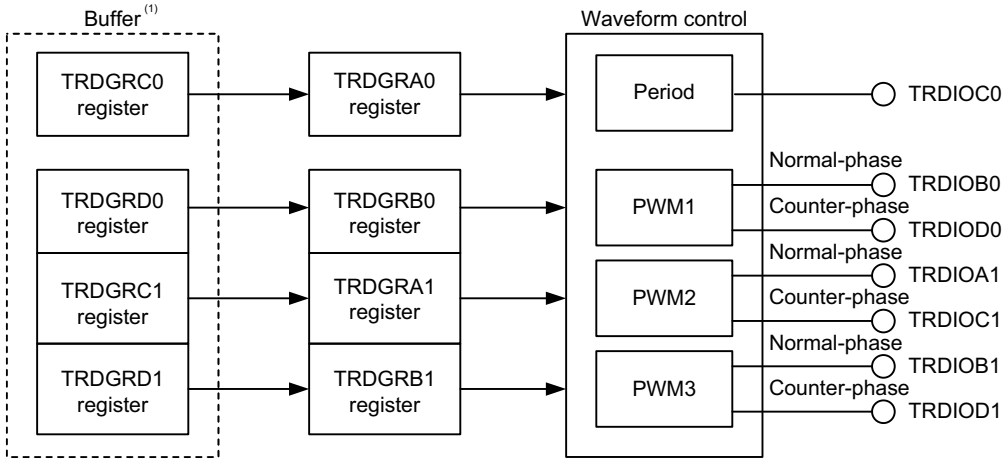
### 8.3.5 Reset Synchronous PWM Mode

In this mode, three normal-phases and three counter-phases of the PWM waveform are output with the same period (three-phase, sawtooth wave modulation, and no dead time).

Figure 8-55 shows the Block Diagram of Reset Synchronous PWM Mode, Table 8-17 lists the Reset Synchronous PWM Mode Specifications, Figure 8-56 shows an Operation Example in Reset Synchronous PWM Mode.

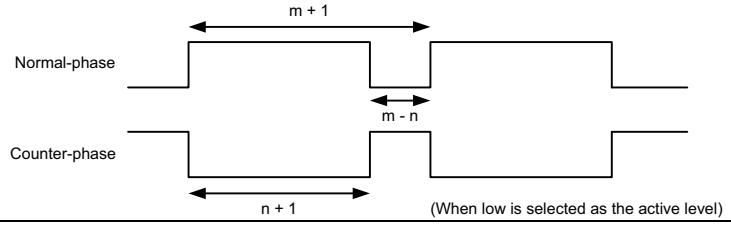
See **Figure 8-54 Operation Example in PWM Function (Duty Cycle 0%, Duty Cycle 100%)** for an operation example in PWM Mode with duty cycle 0% and duty cycle 100%.

**Figure 8-55. Block Diagram of Reset Synchronous PWM Mode**




Note:
1. When bits TRDBFC0, TRDBFD0, TRDBFC1, and TRDBFD1 in the TRDMR register are set to 1 (buffer register).

**Table 8-17. Reset Synchronous PWM Mode Specifications**

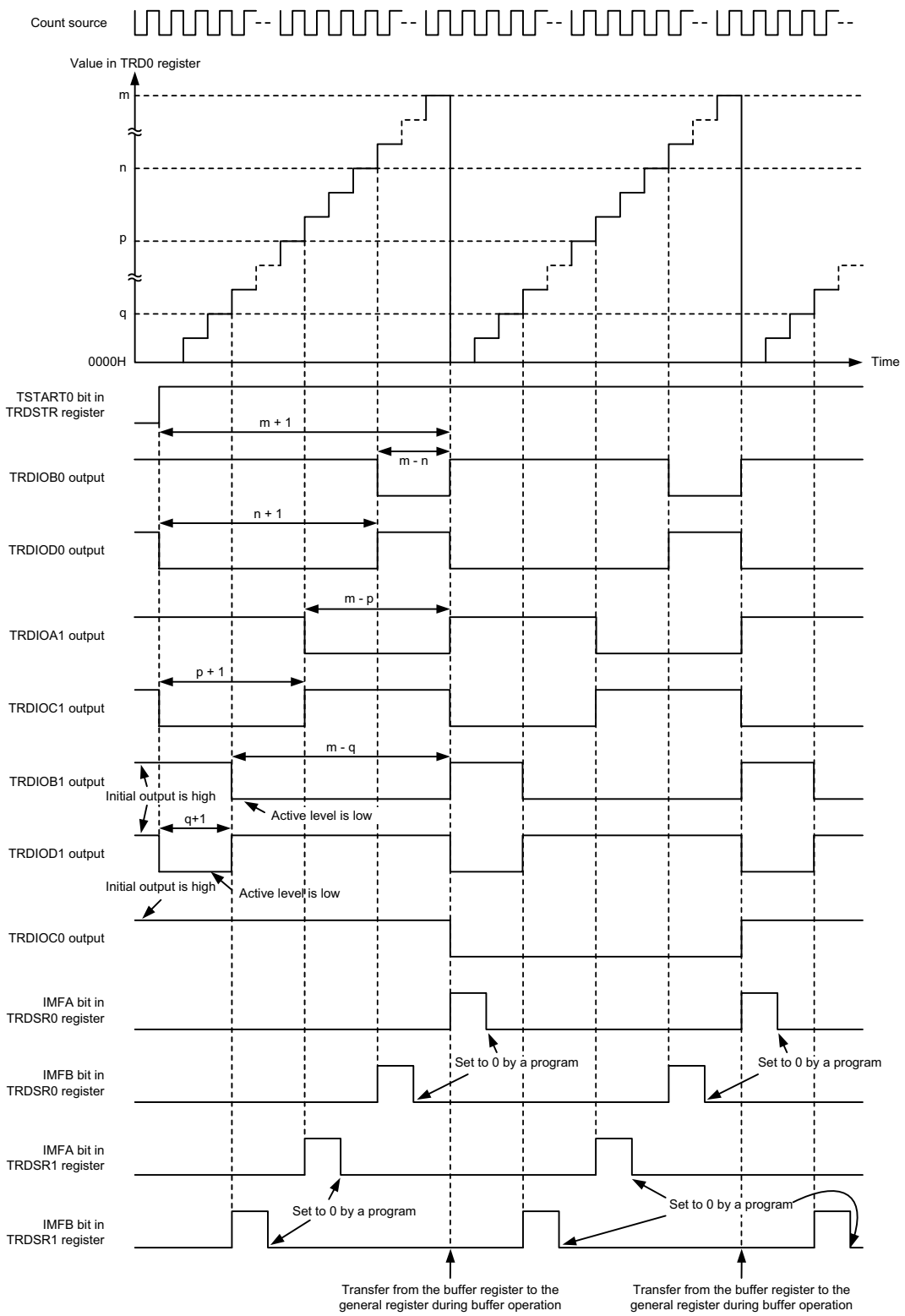
| Item | Specification |
|---|---|
| Count sources Note | $f_{CLK}$, $f_{PLL}$, $f_{IH}$, $f_{SUB}$, $f_{IL}$<br>External signal input to the TRDCLK0 pin (active edge selected by a program) |
| Count operations | The TRD0 register is incremented (the TRD1 register is not used). |
| PWM waveform | PWM period: 1/fk x (m + 1)<br>Active level of normal-phase: 1/fk x (m - n)<br>Inactive level of counter-phase: 1/fk x (n + 1)<br>fk: Frequency of count source<br>m: Value set in the TRDGRA0 register<br>n: Value set in the TRDGRB0 register (PWM1 output)<br>Value set in the TRDGRA1 register (PWM2 output)<br>Value set in the TRDGRB1 register (PWM3 output)<br>  |
| Count start condition | 1 (count starts) is written to the TSTART0 bit in the TRDSTR register. |
| Count stop conditions | • 0 (count stops) is written to the TSTART0 bit when the CSEL0 bit in the TRDSTR register is set to 1.<br>The PWM output pin outputs the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register.<br>• When the CSEL0 bit in the TRDSTR register is set to 0, the count stops at the compare match with the TRDGRA0 register.<br>The PWM output pin outputs the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register. |
| Interrupt request generation timing | • Compare match (content of the TRD0 register matches content of registers TRDGRj0, TRDGRA1, and TRDGRB1)<br>• TRD0 register overflow |
| TRDIOA0 pin function | I/O port or TRDCLK (external clock) input |
| TRDIOB0 pin function | PWM1 output normal-phase output |
| TRDIOD0 pin function | PWM1 output counter-phase output |
| TRDIOA1 pin function | PWM2 output normal-phase output |
| TRDIOC1 pin function | PWM2 output counter-phase output |
| TRDIOB1 pin function | PWM3 output normal-phase output |
| TRDIOD1 pin function | PWM3 output counter-phase output |
| TRDIOC0 pin function | Output inverted every PWM period |
| INTP0 pin function | Pulse output forced cutoff signal input (port or INTP0 interrupt input) |
| Read from timer | The count value can be read by reading the TRD0 register. |
| Write to timer | The value can be written to the TRD0 register. |
| Selectable functions | •The normal-phase and counter-phase active level and initial output level are selected individually.<br>• Buffer operation (see **8. 3. 1 (2) Buffer Operation**)<br>• Pulse output forced cutoff signal input (see **8. 3. 1 (4) Pulse Output Forced Cutoff**) |

**Note** When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

**Remark** j = A, B, C, or D

(1) Operation Example

**Figure 8-56. Operation Example in Reset Synchronous PWM Mode**




Remark
i = 0 or 1
m: Value set in TRDGRA0 register
n: Value set in TRDGRB0 register
p: Value set in TRDGRA1 register
q: Value set in TRDGRB1 register

The above diagram applies under the following condition :
Bits OLS1 and OLS0 in the TRDFCR register are set to 0 (initial output level is high, active level is low).

### 8.3.6 Complementary PWM Mode

In this mode, three normal-phases and three counter-phases of the PWM waveform are output with the same period (three-phase, triangular wave modulation, and with dead time).

Figure 8-57 shows the Block Diagram of Complementary PWM Mode, Table 8-18 lists the Complementary PWM Mode Specifications, and Figure 8-58 shows the Output Model of Complementary PWM Mode, and Figure 8-59 shows an Operation Example in Complementary PWM Mode.

**Figure 8-57. Block Diagram of Complementary PWM Mode**

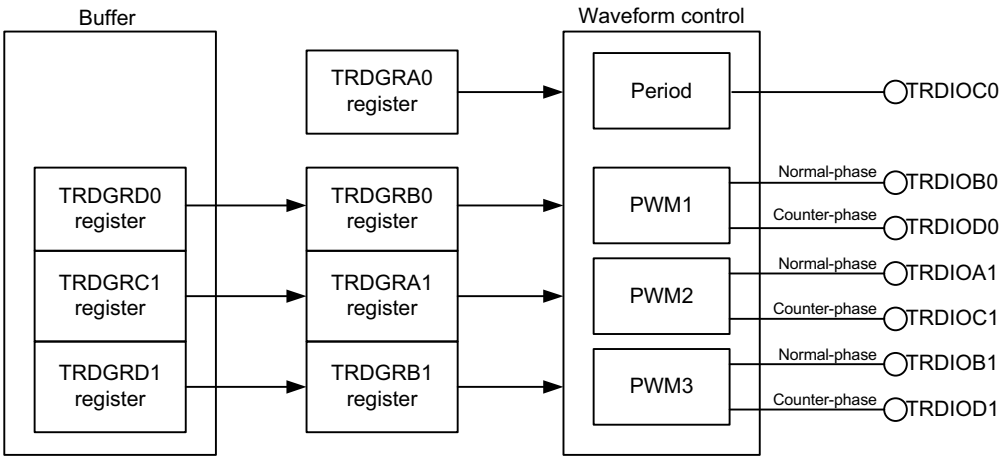
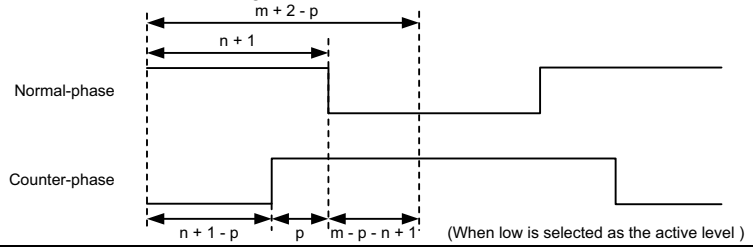


**Table 8-18. Complementary PWM Mode Specifications**

| Item | Specification |
|---|---|
| Count sources Note 1 | $f_{CLK}$, $f_{PLL}$, $f_{IH}$, $f_{SUB}$, $f_{IL}$<br>External signal input to the TRDCLK0 pin (active edge selected by a program)<br>Set bits TCK2 to TCK0 in the TRDCR1 register to the same value (same count source) as bits TCK2 to TCK0 in the TRDCR0 register. |
| Count operations | Increment or decrement.<br>Registers TRD0 and TRD1 are decremented with the compare match with registers TRD0 and TRDGRA0 during increment operation. When the TRD1 register changes from 0000H to FFFFH during decrement operation, and registers TRD0 and TRD1 are incremented. |
| PWM waveform | PWM period: 1/fk × (m + 2 - p) × 2 Note 2<br>Dead time: p<br>Active level width of normal-phase: 1/fk × (m - n - p + 1) × 2<br>Active level width of counter-phase: 1/fk × (n + 1 - p) × 2<br>fk: Frequency of count source<br>m: Value set in the TRDGRA0 register<br>n: Value set in the TRDGRB0 register (PWM1 output)<br>Value set in the TRDGRA1 register (PWM2 output)<br>Value set in the TRDGRB1 register (PWM3 output)<br>p: Value set in the TRD0 register<br>  m + 2 - p<br>n + 1<br>Normal-phase<br>Counter-phase<br>n + 1 - p<br>p<br>m - p - n + 1<br>(When low is selected as the active level ) |
| Count start condition | 1 (count starts) is written to bits TSTART0 and TSTART1 in the TRDSTR register. |
| Count stop condition | 0 (count stops) is written to bits TSTART0 and TSTART1 in the TRDSTR register when the CSEL0 bit in the TRDSTR register is set to 1. (The PWM output pin outputs the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register.) |
| Interrupt request generation timing | • Compare match (content of the TRDi register matches content of the TRDGRji register)<br>• TRD1 register underflow |
| TRDIOA0 pin function | I/O port or TRDCLK (external clock) input |
| TRDIOB0 pin function | PWM1 output normal-phase output |
| TRDIOD0 pin function | PWM1 output counter-phase output |
| TRDIOA1 pin function | PWM2 output normal-phase output |
| TRDIOC1 pin function | PWM2 output counter-phase output |
| TRDIOB1 pin function | PWM3 output normal-phase output |
| TRDIOD1 pin function | PWM3 output counter-phase output |
| TRDIOC0 pin function | Output inverted every 1/2 period of PWM |
| INTP0 pin function | Pulse output forced cutoff signal input (port or INTP0 interrupt input) |
| Read from timer | The count value can be read by reading the TRDi register. |
| Write to timer | The value can be written to the TRDi register. |
| Selectable functions | • Pulse output forced cutoff signal input (see **8. 3. 1 (4) Pulse Output Forced Cutoff)**<br>• The normal-phase and counter-phase active level and initial output level are selected individually.<br>• Transfer timing from the buffer register selection |

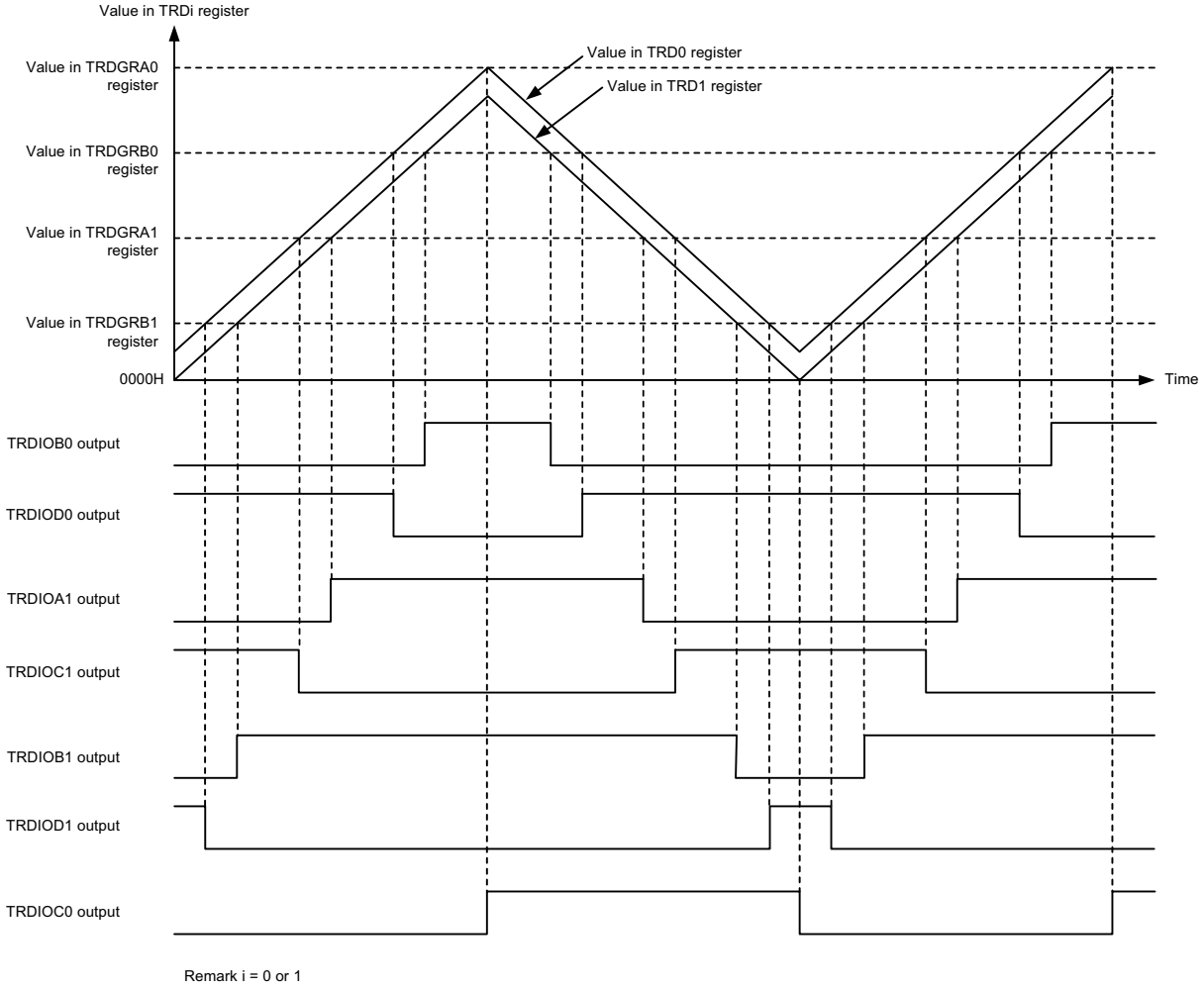
**Notes 1.** When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

**2.** After a count starts, the PWM period is fixed.

**Remark** i = 0 or 1, j = A, B, C, or D

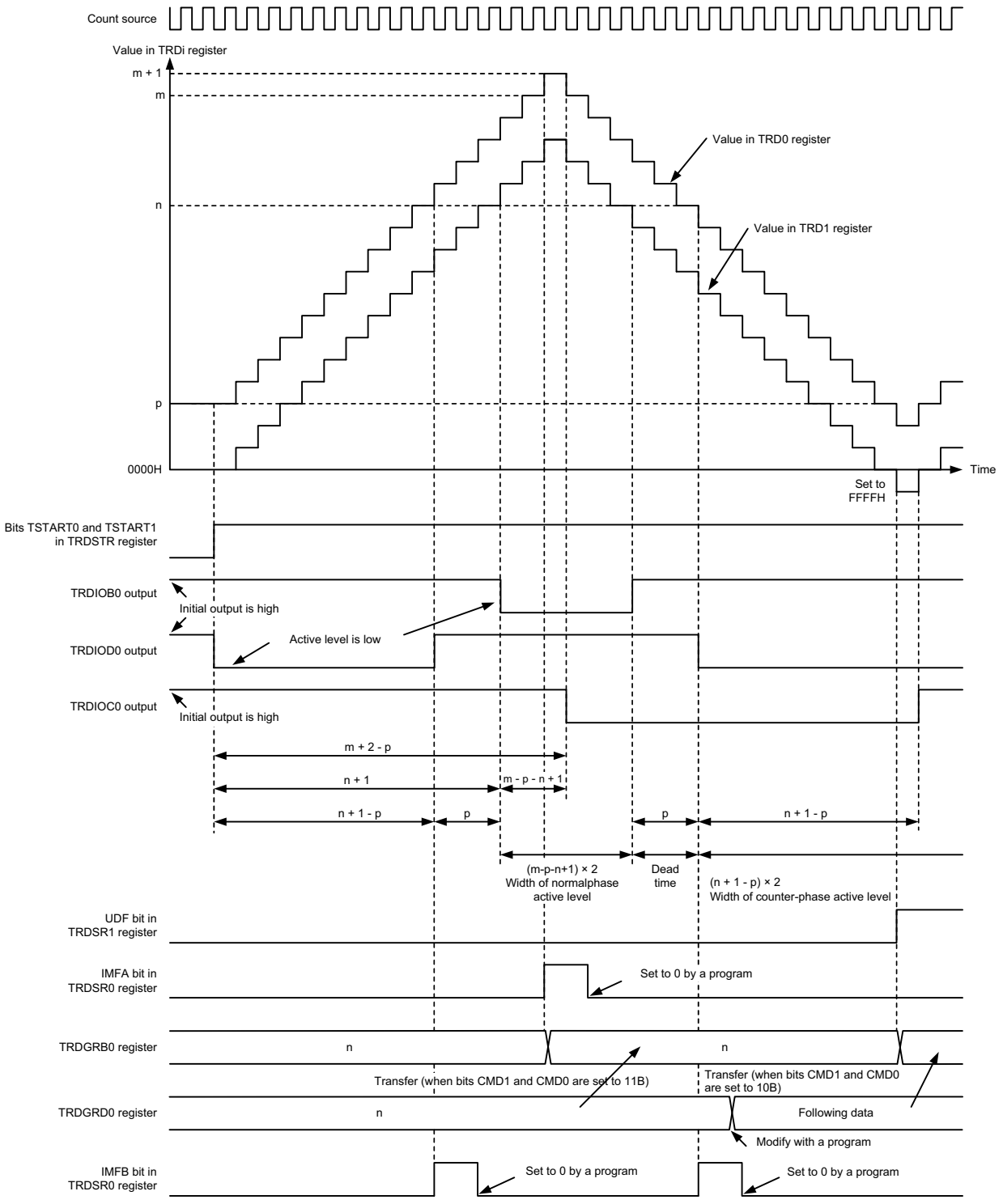
(1) Operation Example

**Figure 8-58. Output Model of Complementary PWM Mode**




Remark i = 0 or 1

**Figure 8-59. Operation Example in Complementary PWM Mode**




Remark
CMD0, CMD1: Bits in TRDFCR register
i = 0 or 1
m: Value set in TRDGRA0 register
n: Value set in TRDGRB0 register
p: Value set in TRD0 register

The above diagram applies under the following condition :
Bits OLS1 and OLS0 in TRDFCR are set to 0 (initial output level is high, active level is low for normal-phase and counter-phase).

(2) Transfer Timing from Buffer Register

- Transfer from the TRDGRD0, TRDGRC1, or TRDGRD1 register to the TRDGRB0, TRDGRA1, or TRDGRB1 register.

  When bits CMD1 and CMD0 in the TRDFCR register are set to 10B, the content is transferred when the TRD1 register underflows.

  When bits CMD1 and CMD0 are set to 11B, the content is transferred at compare match between registers TRD0 and TRDGRA0.

### 8.3.7 PWM3 Mode

In this mode, two PWM waveforms are output with the same period.

Figure 8-60 shows the Block Diagram of PWM3 Mode, Table 8-19 lists the PWM3 Mode Specifications, and Figure 8-61 shows an Operation Example in PWM3 Mode.

**Figure 8-60. Block Diagram of PWM3 Mode**

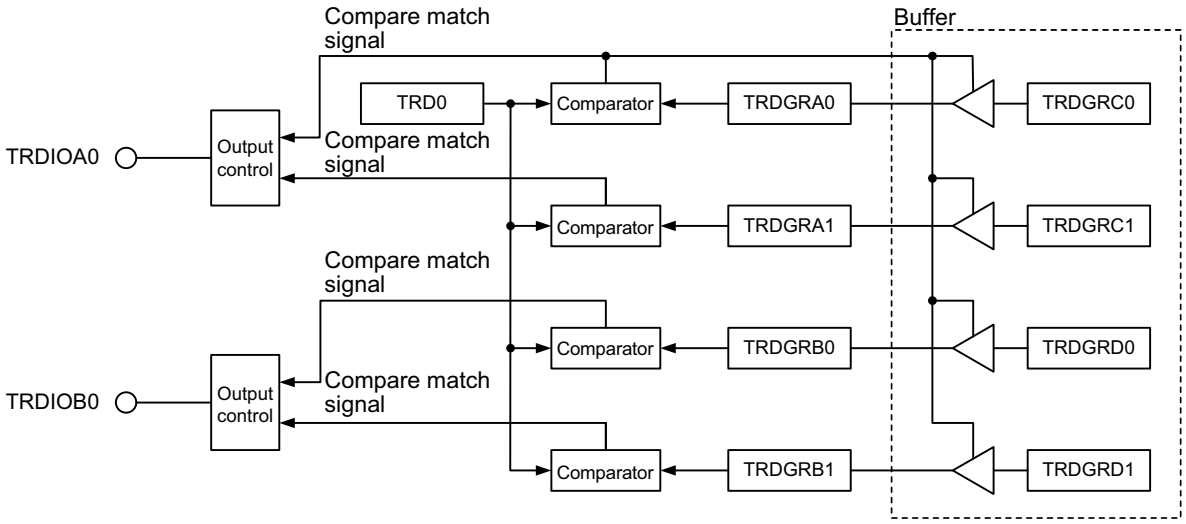
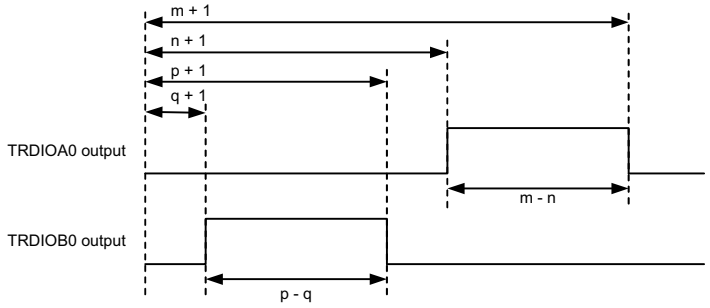


**Table 8-19. PWM3 Mode Specifications**

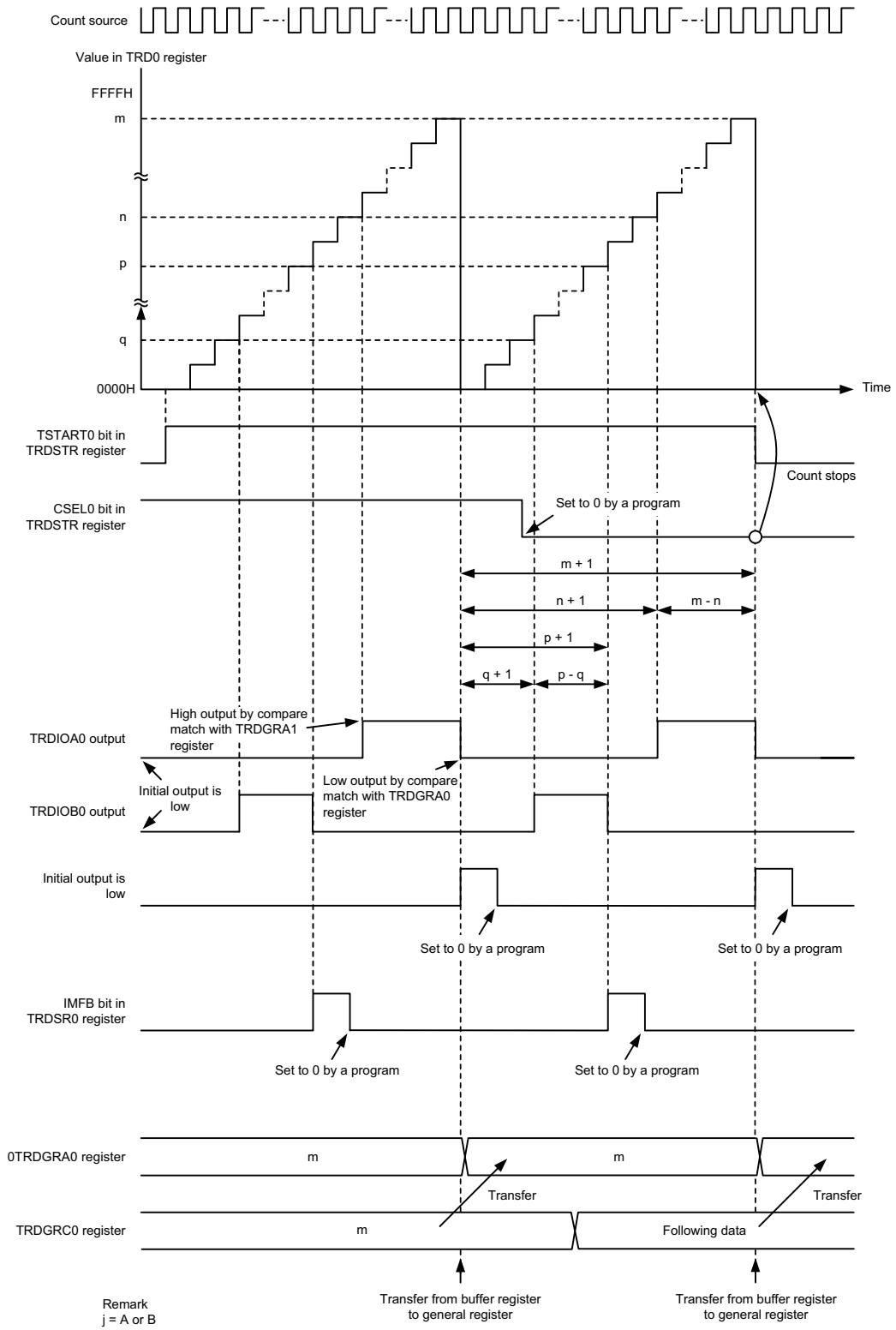
| Item | Specification |
|---|---|
| Count sources Note | $f_{CLK}$, $f_{PLL}$, $f_{IH}$, $f_{SUB}$, $f_{IL}$ |
| Count operations | The TRD0 register is incremented (the TRD1 register is not used). |
| PWM waveform | PWM period: 1/fk × (m + 1)<br>Active level width of TRDIOA0 output: 1/fk × (m - n)<br>Active level width of TRDIOB0 output: 1/fk × (p - q)<br>fk: Frequency of count source<br>m: Value set in the TRDGRA0 register<br>n: Value set in the TRDGRA1 register<br>p: Value set in the TRDGRB0 register<br>q: Value set in the TRDGRB1 register<br>  <br>(When high is selected as the active level ) |
| Count start condition | 1 (count starts) is written to the TSTART0 bit in the TRDSTR register. |
| Count stop conditions | • 0 (count stops) is written to the TSTART0 bit in the TRDSTR register when the CSEL0 bit in the TRDSTR register is set to 1.<br>The PWM output pin holds the output level before the count stops.<br>• When the CSEL0 bit in the TRDSTR register is set to 0, the count stops at compare match with the TRDGRA0 register.<br>The PWM output pin holds the level after output change by compare match. |
| Interrupt request generation timing | • Compare match (content of the TRDi register matches content of the TRDGRji register)<br>• TRD0 register overflow |
| TRDIOA0, TRDIOB0 pin function | PWM output |
| TRDIOA0, TRDIOD0, and TRDIOA1 to TRDIOD1 pin function | I/O port |
| INTP0 pin function | Pulse output forced cutoff signal input (port or INTP0 interrupt input) |
| Read from timer | The count value can be read by reading the TRD0 register. |
| Write to timer | The value can be written to the TRD0 register. |
| Selectable functions | • Pulse output forced cutoff signal input (see **8. 3. 1 (4) Pulse Output Forced Cutoff**)<br>• Active level selectable for each pin.<br>• Buffer operation (see **8. 3. 1 (2) Buffer Operation**) |

**Note** When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

**Remark** i = 0 or 1, j = A, B, C, or D

(1) Operation Example

**Figure 8-61. Operation Example in PWM3 Mode**




Remark
j = A or B
m: Value set in TRDGRA0 register
n: Value set in TRDGRA1 register
p: Value set in TRDGRB0 register
q: Value set in TRDGRB1 register

The above diagram applies under the following conditions :
The above diagram applies under the following conditions :
- Both the TOA0 and TOB0 bits in the TRDOCR register are set to 0 (initial output is low, high output by compare match with TRDGRj1 register, low output by compare match with TRDGRj0 register).
- The TRDBFC0 bit in the TRDMR register is set to 1 (TRDGRC0 register is buffer register for TRDGRA0 register).

## 8.4 Timer RD Interrupt

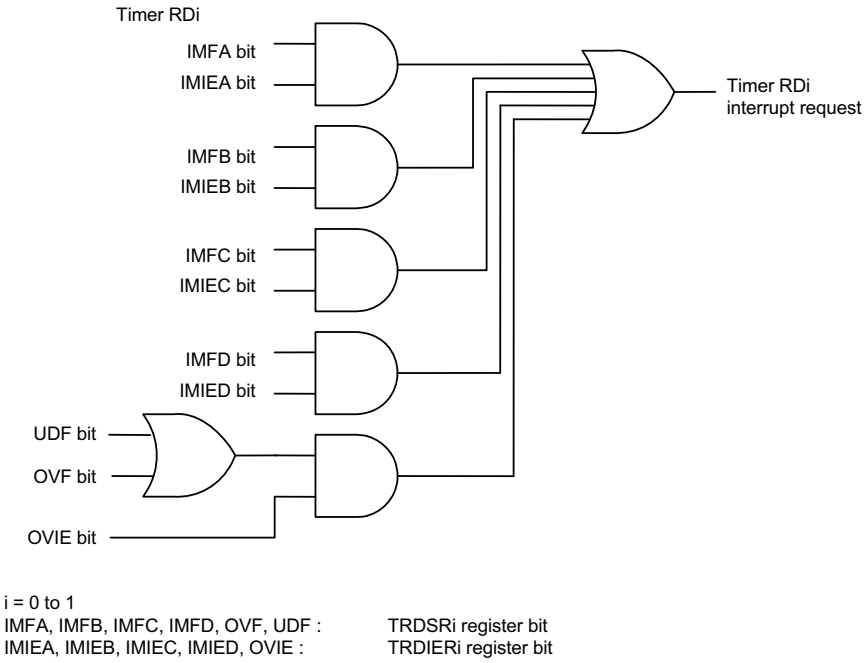
Timer RD generates the timer RDi (i = 0 or 1) interrupt request from six sources for each timer RD0 and timer RD1.

Table 8-20 lists the Registers Associated with Timer RD Interrupt and Figure 8-62 shows the Timer RD Interrupt Block Diagram.

**Table 8-20. Registers Associated with Timer RD Interrupt**

| | Timer RD Status Register | Timer RD Interrupt Enable Register | Interrupt Request Flag (Register) | Interrupt Mask Flag (Register) | Priority Specification Flag (Register) |
|---|---|---|---|---|---|
| Timer RD0 | TRDSR0 | TRDIER0 | TRDIF0 (IF0H) | TRDMK0 (MK0H) | TRDPR00 (PR00H)<br>TRDPR10 (PR10H) |
| Timer RD1 | TRDSR1 | TRDIER1 | TRDIF1 (IF0H) | TRDMK1 (MK0H) | TRDPR01 (PR00H)<br>TRDPR11 (PR10H) |

**Figure 8-62. Timer RD Interrupt Block Diagram**




i = 0 to 1
IMFA, IMFB, IMFC, IMFD, OVF, UDF : TRDSRi register bit
IMIEA, IMIEB, IMIEC, IMIED, OVIE : TRDIERi register bit

Since the interrupt source (timer RD interrupt) is generated by a combination of multiple interrupt request sources for timer RD, the following differences from other maskable interrupts apply:

- When a bit in the TRDSRi register is 1 and the corresponding bit in the TRDIERi register is 1 (interrupt enabled), the TRDIFi bit in the IF0H register is set to 1 (interrupt requested).
- If multiple bits in the TRDIERi register are set to 1, use the TRDSRi register to determine the source of the interrupt request.
- Since the bits in the TRDSRi register are not automatically set to 0 even if the interrupt is acknowledged, set the corresponding bit to 0 in the interrupt routine.
  Use either (a) or (b) described below to clear each bit of the TRDSRi register.
  (a) Set the TRDIERi register to 00H (disabling all interrupts) and then write 0 to all of the status flags.
  (b) When at least one bit in the TRDIERi register has the setting 1 and the status flag of an interrupt source enabled by the corresponding bit is 1, write 0 to all of the status flag bits whose settings are 1 in the TRDSRi register.

- While multiple bits in the TRDIERi register are set to 1, if the first request source is met and the TRDIFi bit is set to 1, and then the next request source is met, the TRDIFi bit is cleared to 0 when the interrupt is acknowledged. However, if the previously met request source is cleared, the TRDIFi bit is set to 1 by the next generated request source.

## 8.5 Notes on Timer RD

### 8.5.1 SFR Read/Write Access

The timer RD SFRs are undefined when FRQSEL4 = 1 in the user option byte (000C2H/020C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

When setting timer RD, set the TRD0EN bit in the PER1 register to 1 first. If the TRD0EN bit is 0, writes to the timer RD control registers are ignored and all the read values are the initial values (except for the port registers and the port mode registers).

The following registers must not be rewritten during count operation:

TRDELC, TRDMR, TRDPMR, TRDFCR, TRDOER1, TRDPTO bit in TRDOER2, TRDDFi, TRDCRi, TRDIORAi, TRDIORCi, TRDPOCRi

(1) TRDSTR Register

- Set the TRDSTR register by an 8-bit memory manipulation instruction.
- When the CSELi bit (i = 0 or 1) in the TRDSTR register is set to 0 (count stops at compare match between registers TRDi and TRDGRAi), the count does not stop and the TSTARTi bit remains unchanged even if 0 (count stops) is written to the TSTARTi bit.
  The TSTARTi bit is set to 0 (count stops) only by a compare match with the TRDGRAi register.
  If the CSELi bit is 0 when rewriting the TRDSTR register, write 0 to the TSTARTi bit to change the CSELi bit to 1 without affecting count operation.
  If 1 is written to the TSTARTi bit while the counter is stopped, count may be started.
  To stop counting by a program, set the TSTARTi bit after setting the CSELi bit to 1. Even if 1 is written to the CSELi bit and 0 is written to the TSTARTi bit at the same time (using one instruction), the count cannot be stopped.
- Table 8-21 lists the TRDIOji (j = A, B, C, or D) Pin Output Level When Count Stops while using the TRDIOji (j = A, B, C, or D) pin for timer RD output.

**Table 8-21. TRDIOji (j = A, B, C, or D) Pin Output Level When Count Stops**

| Count Stop | TRDIOji Pin Output When Count Stops |
|---|---|
| When the CSELi bit is set to 1, write 0 to the TSTARTi bit and the count stops. | The pin holds the output level immediately before the count stops. (The pin outputs the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register in timer RD complementary and reset synchronous PWM modes.) |
| When the CSELi bit is set to 0, the count stops at compare match with registers TRDi and TRDGRAi. | The pin holds the output level after the output changes by compare match. (The pin outputs the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register in timer RD complementary and reset synchronous PWM modes.) |

**Remark** i = 0 or 1, j = A, B, C, or D

(2) TRDDFi Register (i = 0 or 1)

Set bits DFCK0 and DFCK1 in the TRDDFi register before starting count operation.

### 8.5.2 Mode Switching

- Set the count to stopped (set bits TSTART0 and TSTART1 to 0) before switching modes during operation.
- Set bits TRDIF0 and TRDIF1 to 0 before changing bits TSTART0 and TSTART1 from 0 to 1. Refer to **CHAPTER 21 INTERRUPT FUNCTIONS** for details.

### 8.5.3 Count Source

- Switch the count source after the count stops.

[Changing procedure]

(1) Set the TSTARTi bit (i = 0 or 1) in the TRDSTR register to 0 (count stops).

(2) Change bits TCK0 to TCK2 in the TRDCRi register.

- When selecting the count source for the timer RD, set the same clock source as the count source for $f_{CLK}$ before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

### 8.5.4 Input Capture Function

- Set the pulse width of the input capture signal to three or more cycles of the timer RD operating clock ($f_{TRD}$).
- The value of the TRDi register is transferred to the TRDGRji register two to three cycles of the timer RD operating clock ($f_{TRD}$) after the input capture signal is applied to the TRDIOji pin (i = 0 or 1, j = A, B, C, or D) (when no digital filter is used).
- In input capture mode, an input capture interrupt request for the active edge of the TRDIOji input is also generated when the TRDTSTARTi bit in the TRDSTR register is 0 (count stops) if the edge selected by bits TRDIOj0 and TRDIOj1 in the TRDIORji register is input to the TRDIOji pin (i = 0 or 1; j = A, B, C, or D). Set the pulse width of the input capture signal to three or more cycles of the timer RD operating clock ($f_{TRD}$).

### 8.5.5 Procedure for Setting Pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi (i = 0 or 1)

After a reset, the I/O ports multiplexed with pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi function as input ports.

To output from pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi, use the following setting procedure:

Changing procedure

(1) Set the mode and the initial value.

(2) Enable output from pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi (TRDOER1 register).

(3) Set the port register bits corresponding to pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi to 0.

(4) Set the port mode register bits corresponding to pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi to output mode. (Output is started from pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi)

(5) Start the count (set bits TSTART0 and TSTART1 to 1).

To change the port mode register bits corresponding to pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi from output mode to input mode, use the following setting procedure:

(1) Set the port mode register bits corresponding to pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi to input mode (input is started from pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi).

(2) Set to the input capture function.

(3) Start the count (set bits TSTART0 and TSTART1 to 1).

When switching pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi from output mode to input mode, input capture operation may be performed depending on the pin states. When the digital filter is not used, edge detection is performed after two or more cycles of the timer RD operating clock ($f_{TRD}$) have elapsed. When the digital filter is used, edge detection is performed after five or more cycles of the sampling clock have elapsed.

### 8.5.6 External clock TRDCLK0

Set the pulse width of the external clock applied to the TRDCLK0 pin to three or more cycles of the timer RD operating clock ($f_{TRD}$).

### 8.5.7 Reset Synchronous PWM Mode

- When reset synchronous PWM mode is used for motor control, make sure OLS0 = OLS1.
- Set to reset synchronous PWM mode by the following procedure:

[Changing procedure]

(1) Set the TSTART0 bit in the TRDSTR register to 0 (count stops).
(2) Set bits CMD1 and CMD0 in the TRDFCR register to 00B (timer mode, PWM mode, and PWM3 mode).
(3) Set bits CMD1 and CMD0 to 01B (reset synchronous PWM mode).
(4) Set the other registers associated with timer RD again.

### 8.5.8 Complementary PWM Mode

- When complementary PWM mode is used for motor control, make sure OLS0 = OLS1.
- Change bits CMD0 and CMD1 in the TRDFCR register in the following procedure.

Changing procedure: When setting to complementary PWM mode (including re-set), or changing the transfer timing from the buffer register to the general register in complementary PWM mode.

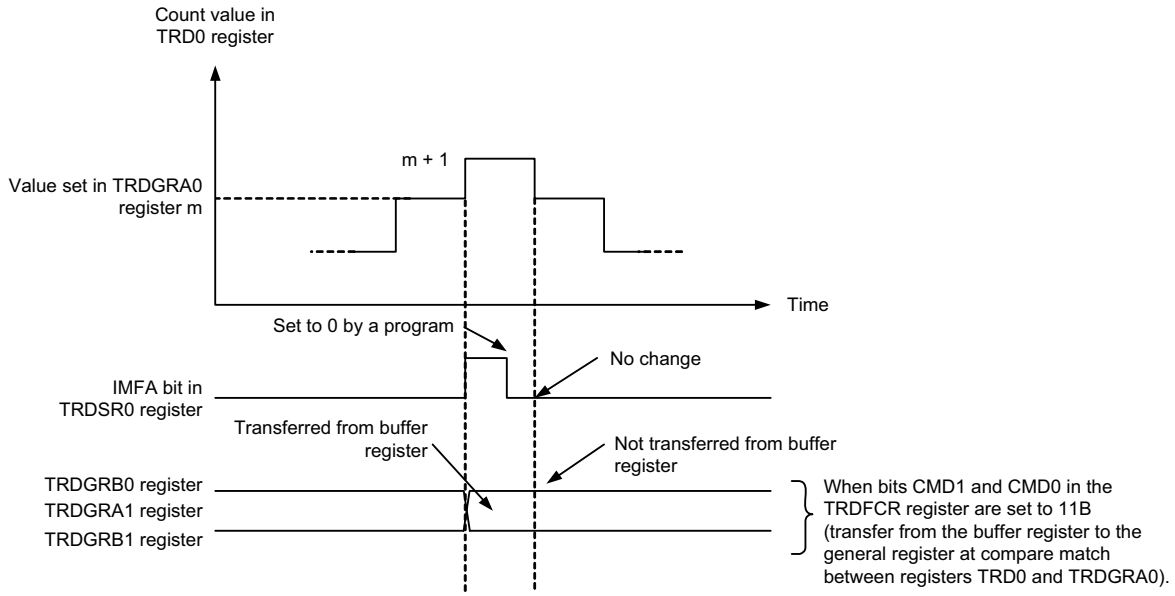
(1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
(2) Set bits CMD1 and CMD0 in the TRDFCR register to 00B (timer mode, PWM mode, and PWM3 mode).
(3) Set bits CMD1 and CMD0 to 10B or 11B (complementary PWM mode).
(4) Set the registers associated with other timer RD again.

Changing procedure: When stopping complementary PWM mode

(1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
(2) Set bits CMD1 to CMD0 to 00B (timer mode, PWM mode, and PWM3 mode).

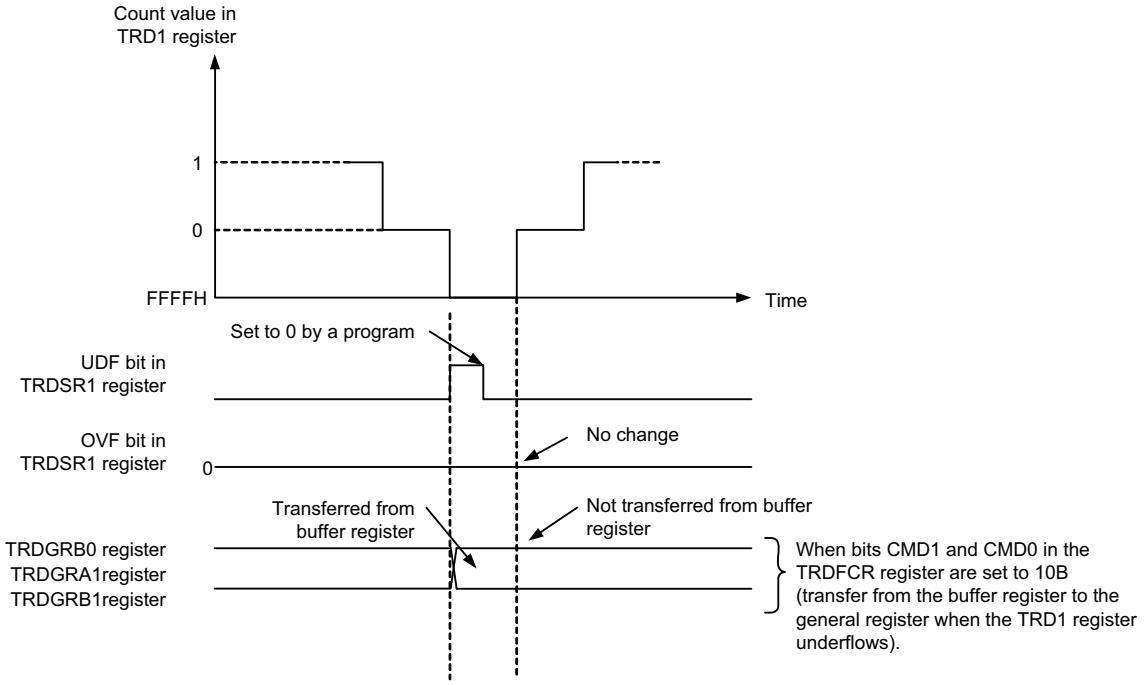
- Do not write to the TRDGRA0, TRDGRB0, TRDGRA1, or TRDGRB1 register during operation.
  When changing the PWM waveform, transfer the values written to registers TRDGRD0, TRDGRC1, and TRDGRD1 to registers TRDGRB0, TRDGRA1, and TRDGRB1 using the buffer operation.
  However, to write data to the TRDGRD0, TRDGRC1, or TRDGRD1 register, set bits TRDBFD0, TRDBFC1, and TRDBFD1 to 0 (general register). After this, bits TRDBFD0, TRDBFC1, and TRDBFD1 may be set to 1 (buffer register).
  The PWM period cannot be changed.
- If the value set in the TRDGRA0 register is assumed to be m, the TRD0 register counts m - 1, m, m + 1, m, m - 1, in that order, when changing from increment to decrement operation.
  When changing from m to m + 1, the IMFA bit in the TRDSRi register is set to 1. Also, bits CMD1 and CMD0 in the TRDFCR register are set to 11B (complementary PWM mode, buffer data transferred at compare match between registers TRD0 and TRDGRA0), the content of the buffer registers (TRDGRD0, TRDGRC1, and TRDGRD1) is transferred to the general registers (TRDGRB0, TRDGRA1, and TRDGRB1).
  During operation of m + 1, m, and m - 1, the IMFA bit remains unchanged and data is not transferred to registers such as the TRDGRA0 register.

**Figure 8-63. Operation at Compare Match between Registers TRD0 and TRDGRA0 in Complementary PWM Mode**



The TRD1 register counts 1, 0, FFFFH, 0, 1, in that order, when changing from decrement to increment operation.
Counting from 1, to 0, to FFFFH causes the UDF bit in the TRDSRi register to be set to 1. Also, when bits CMD1 and CMD0 in the TRDFCR register are set to 10B (complementary PWM mode, buffer data transferred at underflow of the TRD1 register), the content of the buffer registers (TRDGRD0, TRDGRC1, and TRDGRD1) is transferred to the general registers (TRDGRB0, TRDGRA1, and TRDGRB1).
During operation of FFFFH, 0, and 1, data is not transferred to registers such as the TRDGRB0 register.
Also, at this time, the OVF bit in the TRDSRi register remains unchanged.

**Figure 8-64. Operation When TRD1 Register Underflows in Complementary PWM Mode**

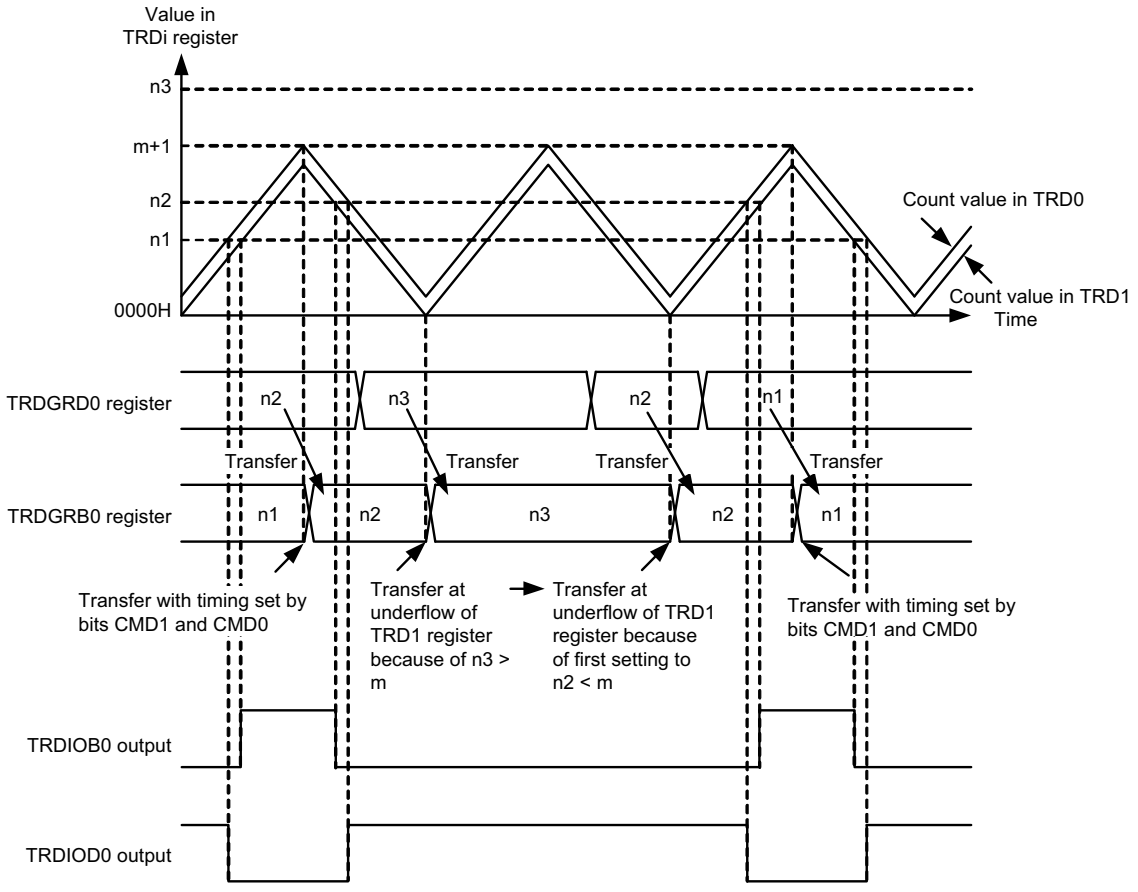


- The timing of data transfer from the buffer register to the general register should be selected using bits CMD0 and CMD1 in the TRDFCR register. However, regardless of the values of bits CMD0 and CMD1, transfer takes place with the following timing when duty cycle is 0% and duty cycle is 100%.

  Value in buffer register ≥ value in TRDGRA0 register (duty cycle is 0%):
  Transfer take place at underflow of the TRD1 register.
  After this, when the buffer register is set to 0001H or above and a smaller value than the value of the TRDGRA0 register, and the TRD1 register underflows for the first time after setting, the value is transferred to the general register. After that, the value is transferred with the timing selected by bits CMD1 and CMD0. A direct change of the duty from 0% to 100% is not possible.

  However, no waveform with duty cycle 0% can be generated while the initial value of the buffer register is FFFFH. To generate a waveform with duty cycle 0%, set the value of the buffer register ≥ TRDGRA0 by writing to the buffer register.

**Figure 8-65. Operation When Value in Buffer Register ≥ Value in TRDGRA0 Register in Complementary PWM Mode**

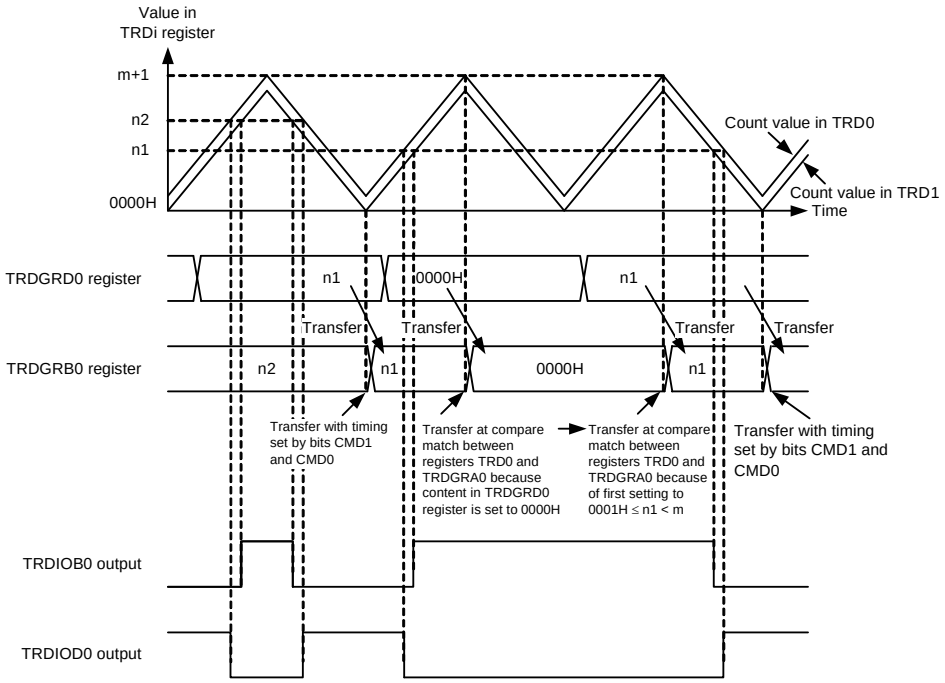



Remark
m: Value set in TRDGRA0 register

The above diagram applies under the following conditions :
- Bits CMD1 and CMD0 in the TRDFCR register are set to 11B
  (data in the buffer register is transferred at compare match between registers TRD 0 and TRDGRA0 in complementary PWM mode).
- Both the OSL0 and OLS1 bits in the TRDFCR register are set to 1 (active high for normal-phase and counter-phase).

When the value in the buffer register is set to 0000H (duty cycle is 100%):
Transfer takes place at compare match between registers TRD0 and TRDGRA0.
After this, when the buffer register is set to 0001H or above and a smaller value than the value of the TRDGRA0 register, and a compare match occurs between registers TRD0 and TRDGRA0 for the first time after setting, the value is transferred to the general register. After that, the value is transferred with the timing selected by bits CMD0 and CMD1. A direct change of the duty from 100% to 0% is not possible.

**Figure 8-66. Operation When Value in Buffer Register is Set to 0000H in Complementary PWM Mode**




Remark
m: Value set in TRDGRA0 register

The above diagram applies under the following conditions:
- Bits CMD1 and CMD0 in the TRDFCR register are set to 10B (data in the buffer register is transferred at underflow of the TRD1 register in PWM mode).
- Both the OLS0 and OLS1 bits in the TRDFCR register are set to 1 (active high for normal-phase and counter-phase).

# CHAPTER 9 REAL-TIME CLOCK

## 9.1 Functions of Real-time Clock

The real-time clock has the following features.

- Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
- Constant-period interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm interrupt function (alarm: week, hour, minute)
- Pin output function of 1 Hz
- Watch error correction register

**Caution The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock ($f_{SUB}$ = 32.768 kHz), high-speed on-chip oscillator ($f_{IH}$ = 4 MHz or 8 MHz), or high-speed system clock ($f_{MX}$ = 4 MHz, 8 MHz, 4.19 MHz, 8.38 MHz) is selected as the operation clock of the real-time clock. When selecting the high-speed on-chip oscillator or high-speed system clock, use the RTC clock select register (RTCCL) to select the clock and the frequency divisor.**

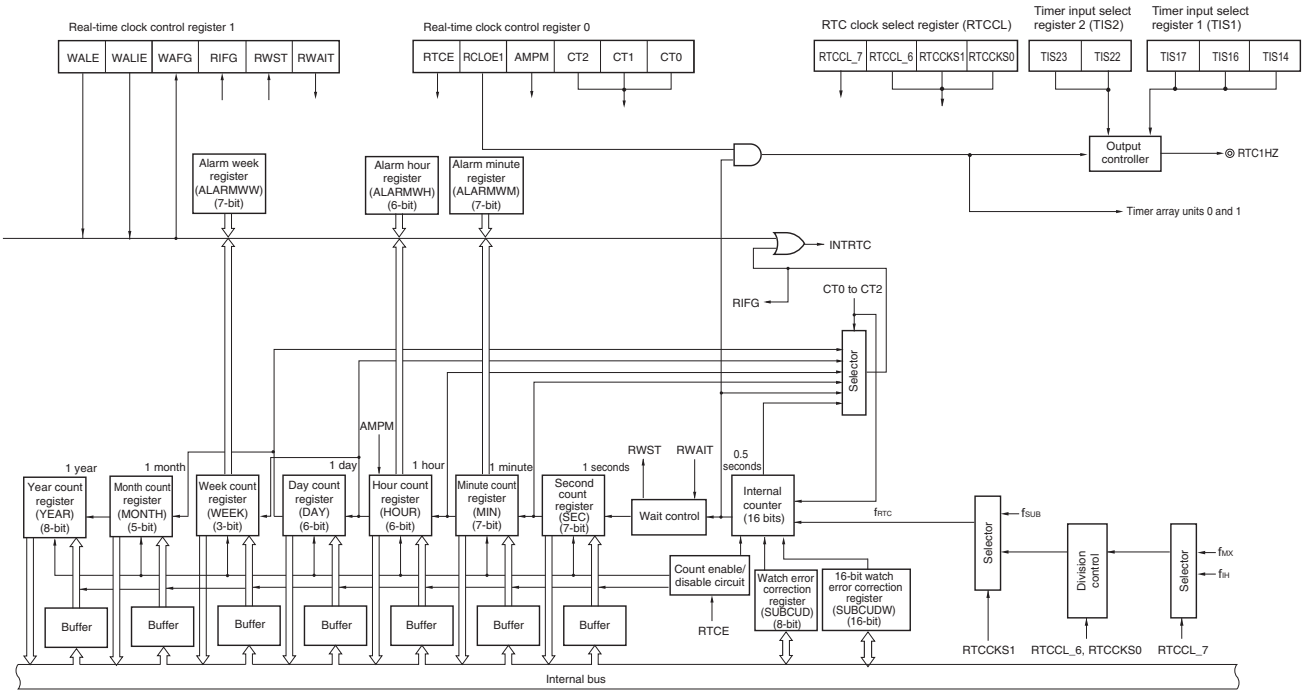
## 9.2 Configuration of Real-time Clock

The real-time clock includes the following hardware.

**Table 9-1. Configuration of Real-time Clock**

| Item | Configuration |
|---|---|
| Counter | Internal counter (16 bits) |
| Control registers | Peripheral enable register 0 (PER0) |
| | Operation speed mode control register (OSMC) |
| | Timer input select register 1 (TIS1) |
| | Timer input select register 2 (TIS2) |
| | RTC clock select register (RTCCL) |
| | Real-time clock control register 0 (RTCC0) |
| | Real-time clock control register 1 (RTCC1) |
| | Second count register (SEC) |
| | Minute count register (MIN) |
| | Hour count register (HOUR) |
| | Day count register (DAY) |
| | Week count register (WEEK) |
| | Month count register (MONTH) |
| | Year count register (YEAR) |
| | Watch error correction register (SUBCUD) |
| | 16-bit watch error correction register (SUBCUDW) |
| | Alarm minute register (ALARMWM) |
| | Alarm hour register (ALARMWH) |
| | Alarm week register (ALARMWW) |

**Figure 9-1. Block Diagram of Real-time Clock**



## 9.3 Registers Controlling Real-time Clock

The real-time clock is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Operation speed mode control register (OSMC)
- Timer input select register 1 (TIS1)
- Timer input select register 2 (TIS2)
- RTC clock select register (RTCCL)
- Real-time clock control register 0 (RTCC0)
- Real-time clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Watch error correction register (SUBCUD)
- 16-bit watch error correction register (SUBCUDW)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm week register (ALARMWW)

### 9.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the real-time clock is used, be sure to set the operation clock of the real-time clock by the RTCCL register before setting bit 7 (RTCEN) of this register to 1.

Set the PER0 register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Writing to the PER0 register is disabled when the GCSC bit of the IAWCTL register is set to 1.

**Figure 9-2. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

| Symbol | <7> | 6 | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PER0 | RTCEN | 0 | ADCEN | IICA0EN Notes 1, 2 | SAU1EN Note 1 | SAU0EN | TAU1EN Note 1 | TAU0EN |

| RTCEN | Control of input clock supply to real-time clock (RTC) |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by the real-time clock (RTC) cannot be written.<br>• The real-time clock (RTC) is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by the real-time clock (RTC) can be read/written. |

**Notes 1.** Not provided in the RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory.

**2.** Not provided in 30-pin products of the RL78/F13 (CAN and LIN incorporated) and 30-pin products of the RL78/F14.

**Cautions 1. When using the real-time clock, first set the RTCEN bit to 1, while oscillation of the input clock ($f_{RTC}$) is stable. If RTCEN = 0, writing to a control register of the real-time clock is ignored, and, even if the register is read, only the default value is read (except for the operation speed mode control register (OSMC), timer input select registers 1 and 2 (TIS1 and TIS2), and RTC clock select register (RTCCL).**

**2. Clock supply to peripheral functions other than the real-time clock can be stopped in HALT mode when the subsystem/low-speed on-chip oscillator select clock is used, by setting the RTCLPC bit of the operation speed mode control register (OSMC) to 1. In this case, set the RTCEN bit of the PER0 register to 1 and the other bits (bits 0 to 6) to 0.**

**3. Be sure to set the operating clock of the real-time clock by RTCCL register before setting bit 7 (RTCEN) of this register to 1.**

**4. Be sure to clear the following bits to 0.**

- **Bits 1, 3, 4, and 6 in the RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory**
- **Bits 4 and 6 in 30-pin products of the RL78/F13 (CAN and LIN incorporated) and in 30-pin products of the RL78/F14**
- **Bit 6 in the products other than above**

### 9.3.2 Operation speed mode control register (OSMC)

The RTCLPC bit can be used to reduce power consumption by stopping clock functions that are unnecessary. For details about setting the RTCLPC bit, see **CHAPTER 5 CLOCK GENERATOR**.

Set the OSMC register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-3. Format of Operation Speed Mode Control Register (OSMC)**

Address: F00F3H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSMC | RTCLPC | 0 | 0 | WUTMMCK0 | 0 | 0 | 0 | 0 |

| RTCLPC | Setting in STOP mode or HALT mode while subsystem/low-speed on-chip oscillator select clock is selected as CPU clock |
|---|---|
| 0 | Enables supply of subsystem/low-speed on-chip oscillator select clock to peripheral functions<br>(See **Table 23-1** for peripheral functions whose operations are enabled.) |
| 1 | Stops supply of subsystem/low-speed on-chip oscillator select clock to peripheral functions other than real-time clock. |

### 9.3.3 Timer input select register 1 (TIS1)

The TIS1 register selects an input source of the timer array unit 0.

The TIS17, TIS16, and TIS14 bits in the TIS1 register are used in conjunction with the real time clock to implement the watch error correction in channels 7 and 6. When the TIS17 and TIS16 bits are set to 0 and 1 respectively, the RTC1HZ output signal is selected for the timer input of channel 7.

When the TIS14 bit is set to 1, the RTC1HZ output signal is selected for the timer input of channel 6.

Set the TIS1 register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-4. Format of Timer Input Select Register 1 (TIS1)**

Address: F0075H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TIS1 | TIS17 | TIS16 | 0 | TIS14 | 0 | TIS12 | 0 | TIS0 |

| TIS17 | TIS16 | Selection of timer input used with channel 7 of timer array unit 0 |
|---|---|---|
| 0 | 0 | Input signal of timer input pin (TI07) |
| 0 | 1 | RTC1HZ output signal |
| 1 | 0 | RxD0 pin (detection of the wake-up signal and measurement of the low-level width of the sync break field and the pulse width of the sync field) |
| 1 | 1 | Setting prohibited |

| TIS14 | Selection of timer input used with channel 6 of timer array unit 0 |
|---|---|
| 0 | Input signal of timer input pin (TI06) |
| 1 | RTC1HZ output signal |

| TIS12 | Selection of timer input used with channel 5 of timer array unit 0 |
|---|---|
| 0 | Input signal of timer input pin (TI05) |
| 1 | Input signal of timer input pin (TI03) |

| TIS10 | Selection of timer input used with channel 4 of timer array unit 0 |
|---|---|
| 0 | Input signal of timer input pin (TI04) |
| 1 | Input signal of timer input pin (TI03) |

**Cautions 1. Do not change the select bit of the timer input while inputting data to the TImn pin (m = 0, 1; n = 0 to 7).**

**2. When selecting the RTC1HZ output signal for the clock source of the timer input used in channels 7 and 6 in the TAU, set the TIS17, TIS16, and TIS14 bits to 0, 1, and 1 respectively and select the RTC1HZ output signal for the timer input of channels 7 and 6.**

**Remark** Set the TIS17 and TIS16 bits to 1 and 0 respectively and select the input signal of the RxD0 pin before using the LIN-bus communication.

### 9.3.4 Timer input select register 2 (TIS2)

The TIS2 register selects an input source of the timer array unit 1.

The TIS23 and TIS22 bits in the TIS2 register are used in conjunction with the real time clock to implement the watch error correction in channels 7 and 6. When the TIS23 bit is set to 1, the RTC1HZ output signal is selected for the timer input of channel 7. When the TIS22 bit is set to 1, the RTC1HZ output signal is selected for the timer input of channel 6.

Set the TIS2 register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

This function is valid only for the Group E products.

**Figure 9-5. Format of Timer Input Select Register 2 (TIS2)**

Address: F007AH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TIS2 | 0 | 0 | 0 | 0 | TIS23 | TIS22 | 0 | 0 |

| TIS22 | Selection of timer input used with channel 6 of timer array unit 1 |
|---|---|
| 0 | Input signal of timer input pin (TI16) |
| 1 | RTC1HZ output signal |

| TIS23 | Selection of timer input used with channel 7 of timer array unit 1 |
|---|---|
| 0 | Input signal of timer input pin (TI17) |
| 1 | RTC1HZ output signal |

**Cautions 1. Do not change the select bit of the timer input while inputting data to the TImn pin (m = 0, 1; n = 0 to 7).**

**2. When selecting the RTC1HZ output signal for the clock source of the timer input used in channels 7 and 6 in the TAU, set the TIS23 and TIS22 bits to 1 and select the RTC1HZ output signal for the timer input of channels 7 and 6.**

### 9.3.5 RTC clock select register (RTCCL)

The RTCCL register is used to select the operation clock of the real-time clock.

Set the RTCCL register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-6. Format of RTC clock select register (RTCCL)**

Address: F02C8H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| RTCCL | RTCCL_7 | RTCCL_6 | - | - | - | - | RTCCKS1 | RTCCKS0 |

| RTCCL_7 | Control over operation of the low-speed on-chip oscillator |
|---|---|
| 0 | High-speed system clock ($f_{MX}$) |
| 1 | High-speed on-chip oscillator clock ($f_{IH}$) |

| RTCCKS1 Note 3 | RTCCKS0 | RTCCL_6 | Control of RTC operating clock selection |
|---|---|---|---|
| 0 | 0 | × | Subsystem clock ($f_{SUB}$) Note 1 |
| 0 | 1 | × | |
| 1 | 0 | 0 | $f_{MX}$ or $f_{IH}$ / 128 Note 2 |
| 1 | 0 | 1 | $f_{MX}$ or $f_{IH}$ / 122 Note 2 |
| 1 | 1 | 0 | $f_{MX}$ or $f_{IH}$ / 256 Note 2 |
| 1 | 1 | 1 | $f_{MX}$ or $f_{IH}$ / 244 Note 2 |

**Notes 1.** When the SELLOSC bit in the CKSEL register is 1, the subsystem clock ($f_{SUB}$) cannot be supplied to the input clock ($f_{RTC}$) of the real time clock.

**2.** Switch after selecting RTCCL_7.

**3.** When setting the RTCCKS1 bit to 1, first set the CSS bit in the CKC register to 0 to select the main system/PLL select clock ($f_{MP}$) as the CPU/peripheral hardware clock ($f_{CLK}$).

**Cautions 1. 20, 30, or 32-pin products do not have a subsystem clock ($f_{SUB}$), so it should not be selected.**

**2. Set bits 2 to 5 to 0.**

**Remark** ×: don't care

### 9.3.6 Real-time clock control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time clock operation, control the RTC1HZ pin, and set a 12- or 24-hour system and the constant-period interrupt function.

Set the RTCC0 register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-7. Format of Real-time Clock Control Register 0 (RTCC0)**

Address: FFF9DH After reset: 00H R/W

| Symbol | <7> | 6 | <5> | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| RTCC0 | RTCE | 0 | RCLOE1 | 0 | AMPM | CT2 | CT1 | CT0 |

| RTCE | Real-time clock operation control |
|---|---|
| 0 | Stops counter operation. |
| 1 | Starts counter operation. |

| RCLOE1 | RTC1HZ pin output control |
|---|---|
| 0 | Disables output of the RTC1HZ pin (1 Hz). |
| 1 | Enables output of the RTC1HZ pin (1 Hz). |

| AMPM | Selection of 12-/24-hour system |
|---|---|
| 0 | 12-hour system (a.m. and p.m. are displayed.) |
| 1 | 24-hour system |
| • Rewrite the AMPM bit value after setting the RWAIT bit (bit 0 of real-time clock control register 1 (RTCC1)) to 1. If the AMPM bit value is changed, the values of the hour count register (HOUR) change according to the specified time system.<br>• Table 9-2 shows the displayed time digits that are displayed. | |

| CT2 | CT1 | CT0 | Constant-period interrupt (INTRTC) selection |
|---|---|---|---|
| 0 | 0 | 0 | Does not use fixed-cycle interrupt function. |
| 0 | 0 | 1 | Once per 0.5 s (synchronized with second count up) |
| 0 | 1 | 0 | Once per 1 s (same time as second count up) |
| 0 | 1 | 1 | Once per 1 m (second 00 of every minute) |
| 1 | 0 | 0 | Once per 1 hour (minute 00 and second 00 of every hour) |
| 1 | 0 | 1 | Once per 1 day (hour 00, minute 00, and second 00 of every day) |
| 1 | 1 | × | Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month) |
| When changing the values of the CT2 to CT0 bits while the counter operates (RTCE = 1), rewrite the values of the CT2 to CT0 bits after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of the CT2 to CT0 bits, enable interrupt servicing after clearing the RIFG and RTCIF flags. | | | |

**Caution Do not change the value of the RTCLOE1 bit when RTCE = 1.**

**Remark** ×: don't care

### 9.3.7 Real-time clock control register 1 (RTCC1)

The RTCC1 register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.

Set the RTCC1 register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-8. Format of Real-time Clock Control Register 1 (RTCC1) (1/2)**

Address: FFF9EH After reset: 00H R/W Note

| Symbol | <7> | <6> | 5 | <4> | <3> | 2 | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| RTCC1 | WALE | WALIE | 0 | WAFG | RIFG | 0 | RWST | RWAIT |

| WALE | Alarm operation control |
|---|---|
| 0 | Match operation is invalid. |
| 1 | Match operation is valid. |
| When setting a value to the WALE bit while the counter operates (RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting each alarm register (WALIE flag of real-time clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm week register (ALARMWW)), set match operation to be invalid (“0”) for the WALE bit. | |

| WALIE | Control of alarm interrupt (INTRTC) function operation |
|---|---|
| 0 | Does not generate interrupt on matching of alarm. |
| 1 | Generates interrupt on matching of alarm. |

| WAFG | Alarm detection status flag |
|---|---|
| 0 | Alarm mismatch |
| 1 | Detection of matching of alarm |
| This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to “1”, one operating clock ($f_{RTC}$) after matching of the alarm is detected. This flag is cleared when “0” is written to it. Writing “1” to it is invalid. | |

**Note** Bit 1 is read-only.

**Figure 9-8. Format of Real-time Clock Control Register 1 (RTCC1) (2/2)**

| RIFG | Constant-period interrupt status flag |
|---|---|
| 0 | Fixed-cycle interrupt is not generated. |
| 1 | Fixed-cycle interrupt is generated. |
| This flag indicates the status of generation of the fixed-cycle interrupt. When the fixed-cycle interrupt is generated, it is set to “1”.<br>This flag is cleared when “0” is written to it. Writing “1” to it is invalid. | |

| RWST | Wait status flag of real-time clock |
|---|---|
| 0 | Counter is operating. |
| 1 | Mode to read or write counter value |
| This status flag indicates whether the setting of the RWAIT bit is valid.<br>Before reading or writing the counter value, confirm that the value of this flag is 1. | |

| RWAIT | Wait control of real-time clock |
|---|---|
| 0 | Sets counter operation. |
| 1 | Stops SEC to YEAR counters. Mode to read or write counter value |
| This bit controls the operation of the counter.<br>Be sure to write “1” to it to read or write the counter value.<br>As the internal counter (16 bits) is continuing to run, complete reading or writing within one second and turn back to 0.<br>When RWAIT = 1, it takes up to 1 operating clock ($f_{RTC}$) until the counter value can be read or written (RWST = 1). **Notes 1, 2**<br>When the internal counter (16 bits) overflowed while RWAIT = 1, it keeps the event of overflow until RWAIT = 0, then counts up.<br>However, when it wrote a value to second count register, it will not keep the overflow event. | |

<R> **Notes 1.** When setting RWAIT=1 during 1 operating clock ($f_{RTC}$) after setting RTCE=1, it may take two clock time of the operation clock($f_{RTC}$) until RWST bit becomes “1”.

<R> **Notes 2.** When setting RWAIT=1 during 1 operating clock ($f_{RTC}$) after returning from a stand-by (HALT mode, STOP mode, SNOOZE mode), it may take two clock time of the operation clock($f_{RTC}$) until RWST bit becomes “1”.

**Caution If writing is performed to the RTCC1 register with a 1-bit manipulation instruction, the RIFG flag and WAFG flag may be cleared. Therefore, to perform writing to the RTCC1 register, be sure to use an 8-bit manipulation instruction. To prevent the RIFG flag and WAFG flag from being cleared during writing, disable writing by setting 1 to the corresponding bit. If the RIFG flag and WAFG flag are not used and the value may be changed, the RTCC1 register may be written by using a 1-bit manipulation instruction.**

**Remarks 1.** Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

**2.** If writing is performed to the second count register (SEC), the internal counter (16 bits) is cleared.

### 9.3.8 Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the internal counter (16 bits) overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 operating clocks ($f_{RTC}$) later. Set a decimal value of 00 to 59 to this register in BCD code.

Set the SEC register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-9. Format of Second Count Register (SEC)**

Address: FFF92H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| SEC | 0 | SEC40 | SEC20 | SEC10 | SEC8 | SEC4 | SEC2 | SEC1 |

**Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), see 9.4.3 Reading/writing real-time clock and follow the described procedures.**

**Remark** If writing is performed to the second count register (SEC), the internal counter (16 bits) is cleared.

### 9.3.9 Minute count register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 operating clocks ($f_{RTC}$) later. Even if the second count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 59 to this register in BCD code.

Set the MIN register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-10. Format of Minute Count Register (MIN)**

Address: FFF93H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| MIN | 0 | MIN40 | MIN20 | MIN10 | MIN8 | MIN4 | MIN2 | MIN1 |

**Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), see 9.4.3 Reading/writing real-time clock and follow the described procedures.**

### 9.3.10 Hour count register (HOUR)

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 operating clocks ($f_{RTC}$) later. Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using bit 3 (AMPM) of real-time clock control register 0 (RTCC0).

If the AMPM bit value is changed, the values of the HOUR register change according to the specified time system.

Set the HOUR register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

**Figure 9-11. Format of Hour Count Register (HOUR)**

Address: FFF94H After reset: 12H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| HOUR | 0 | 0 | HOUR20 | HOUR10 | HOUR8 | HOUR4 | HOUR2 | HOUR1 |

**Cautions 1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).**

**2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.**

Table 9-2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

**Table 9-2. Displayed Time Digits**

| 24-Hour Display (AMPM = 1) | | 12-Hour Display (AMPM = 1) | |
|---|---|---|---|
| Time | HOUR Register | Time | HOUR Register |
| 0 | 00H | 12 a.m. | 12H |
| 1 | 01H | 1 a.m. | 01H |
| 2 | 02H | 2 a.m. | 02H |
| 3 | 03H | 3 a.m. | 03H |
| 4 | 04H | 4 a.m. | 04H |
| 5 | 05H | 5 a.m. | 05H |
| 6 | 06H | 6 a.m. | 06H |
| 7 | 07H | 7 a.m. | 07H |
| 8 | 08H | 8 a.m. | 08H |
| 9 | 09H | 9 a.m. | 09H |
| 10 | 10H | 10 a.m. | 10H |
| 11 | 11H | 11 a.m. | 11H |
| 12 | 12H | 12 p.m. | 32H |
| 13 | 13H | 1 p.m. | 21H |
| 14 | 14H | 2 p.m. | 22H |
| 15 | 15H | 3 p.m. | 23H |
| 16 | 16H | 4 p.m. | 24H |
| 17 | 17H | 5 p.m. | 25H |
| 18 | 18H | 6 p.m. | 26H |
| 19 | 19H | 7 p.m. | 27H |
| 20 | 20H | 8 p.m. | 28H |
| 21 | 21H | 9 p.m. | 29H |
| 22 | 22H | 10 p.m. | 30H |
| 23 | 23H | 11 p.m. | 31H |

The HOUR register value is set to 12-hour display when the AMPM bit is “0” and to 24-hour display when the AMPM bit is “1”.

In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

### 9.3.11 Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.
It counts up when the hour counter overflows.
This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to 2 operating clocks ($f_{RTC}$) later. Even if the hour count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 31 to this register in BCD code.
Set the DAY register by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 01H.

**Figure 9-12. Format of Day Count Register (DAY)**

Address: FFF96H After reset: 01H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DAY | 0 | 0 | DAY20 | DAY10 | DAY8 | DAY4 | DAY2 | DAY1 |

**Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), see 9.4.3 Reading/writing real-time clock and follow the described procedures.**

### 9.3.12 Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.

It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to 2 operating clocks ($f_{RTC}$) later. Set a decimal value of 00 to 06 to this register in BCD code.

Set the WEEK register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-13. Format of Week Count Register (WEEK)**

Address: FFF95H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| WEEK | 0 | 0 | 0 | 0 | 0 | WEEK4 | WEEK2 | WEEK1 |

**Cautions 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically. After reset release, set the week count register as follow.**

| Day | WEEK |
|---|---|
| Sunday | 00H |
| Monday | 01H |
| Tuesday | 02H |
| Wednesday | 03H |
| Thursday | 04H |
| Friday | 05H |
| Saturday | 06H |

**2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), see 9.4.3 Reading/writing real-time clock and follow the described procedures.**

### 9.3.13 Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 operating clocks ($f_{RTC}$) later. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code.

Set the MONTH register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 01H.

**Figure 9-14. Format of Month Count Register (MONTH)**

Address: FFF97H After reset: 01H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| MONTH | 0 | 0 | 0 | MONTH10 | MONTH8 | MONTH4 | MONTH2 | MONTH1 |

**Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), see 9.4.3 Reading/writing real-time clock and follow the described procedures.**

### 9.3.14 Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month count register (MONTH) overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to 2 operating clocks ($f_{RTC}$) later. Even if the MONTH register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code.

Set the YEAR register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-15. Format of Year Count Register (YEAR)**

Address: FFF98H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| YEAR | YEAR80 | YEAR40 | YEAR20 | YEAR10 | YEAR8 | YEAR4 | YEAR2 | YEAR1 |

**Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), see 9.4.3 Reading/writing real-time clock and follow the described procedures.**

### 9.3.15 Watch error correction register (SUBCUD)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value that overflows from the internal counter (16 bits) to the second count register (SEC) (reference value: 7FFFH).

Set the SUBCUD register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-16. Format of Watch Error Correction Register (SUBCUD)**

Address: FFF99H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| SUBCUD | DEV | F12 | F5 | F4 | F3 | F2 | F1 | F0 |

| DEV | Setting of watch error correction timing |
|---|---|
| 0 | Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds). |
| 1 | Corrects watch error only when the second digits are at 00 (every 60 seconds). |
| Writing to the SUBCUD register at the following timing is prohibited.<br>• When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H<br>• When DEV = 1 is set: For a period of SEC = 00H | |

| F12 | Setting of watch error correction value |
|---|---|
| 0 | Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2. |
| 1 | Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2. |
| When (F12, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected.<br>Range of correction value: (when F12 = 0) 2, 4, 6, 8, … , 120, 122, 124<br>(when F12 = 1) –2, –4, –6, –8, … , –120, –122, –124 | |

**Cautions 1. / of /Fn (n = 0 to 5) means invert.**

**2. * means 0 or 1.**

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

| | DEV = 0 (correction every 20 seconds) | DEV = 1 (correction every 60 seconds) |
|---|---|---|
| Correctable range | –189.2 ppm to 189.2 ppm | –63.1 ppm to 63.1 ppm |
| Maximum excludes quantization error | ± 1.53 ppm | ± 0.51 ppm |
| Minimum resolution | ± 3.05 ppm | ± 1.02 ppm |

**Remark** If a correctable range is –63.1 ppm or lower and 63.1 ppm or higher, set 0 to DEV.

### 9.3.16 16-bit watch error correction register (SUBCUDW)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value that overflows from the internal counter (16 bits) to the second count register (SEC) (reference value: 7FFFH).

Set the SUBCUDW register by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

**Figure 9-17. Format of 16-Bit Watch Error Correction Register (SUBCUDW)**

Address: FFF54H After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| SUBCUDW | DEV | - | - | F12 | F11 | F10 | F9 | F8 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| F7 | F6 | F5 | F4 | F3 | F2 | F1 | F0 |

| DEV | Setting of watch error correction timing |
|---|---|
| 0 | Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds). |
| 1 | Corrects watch error only when the second digits are at 00 (every 60 seconds). |
| Writing to the SUBCUDW register at the following timing is prohibited.<br>• When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H<br>• When DEV = 1 is set: For a period of SEC = 00H | |

| F12 | Setting of watch error correction value |
|---|---|
| 0 | Increases by {(F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) – 1} × 2. |
| 1 | Decreases by {(/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2. |
| When (F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *), the watch error is not corrected.<br>Range of correction value: (when F12 = 0) 2, 4, 6, 8, … , 8184, 8186, 8188<br>(when F12 = 1) –2, –4, –6, –8, … , –8184, –8186, –8188 | |

**Cautions 1. / of /Fn (n = 0 to 11) means invert.**

**2. * means 0 or 1.**

The range of value that can be corrected by using the 16-bit watch error correction register (SUBCUDW) is shown below.

| | DEV = 0 (correction every 20 seconds) | DEV = 1 (correction every 60 seconds) |
|---|---|---|
| Correctable range | –12496.9 ppm to 12496.9 ppm | –4165.6 ppm to 4165.6 ppm |
| Maximum excludes quantization error | ± 1.53 ppm | ± 0.51 ppm |
| Minimum resolution | ± 3.05 ppm | ± 1.02 ppm |

**Remark** If a correctable range is −4165.6 ppm or lower and 4165.6 ppm or higher, set DEV to 0.

### 9.3.17 Alarm minute register (ALARMWM)

This register is used to set minutes of alarm.

Set the ALARMWM register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.**

**Figure 9-18. Format of Alarm Minute Register (ALARMWM)**

Address: FFF9AH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ALARMWM | 0 | WM40 | WM20 | WM10 | WM8 | WM4 | WM2 | WM1 |

### 9.3.18 Alarm hour register (ALARMWH)

This register is used to set hours of alarm.

Set the ALARMWH register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

**Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.**

**Figure 9-19. Format of Alarm Hour Register (ALARMWH)**

Address: FFF9BH After reset: 12H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ALARMWH | 0 | 0 | WH20 | WH10 | WH8 | WH4 | WH2 | WH1 |

**Caution Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).**

### 9.3.19 Alarm week register (ALARMWW)

This register is used to set date of alarm.

Set the ALARMWW register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 9-20. Format of Alarm Week Register (ALARMWW)**

Address: FFF9CH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ALARMWW | 0 | WW6 | WW5 | WW4 | WW3 | WW2 | WW1 | WW0 |

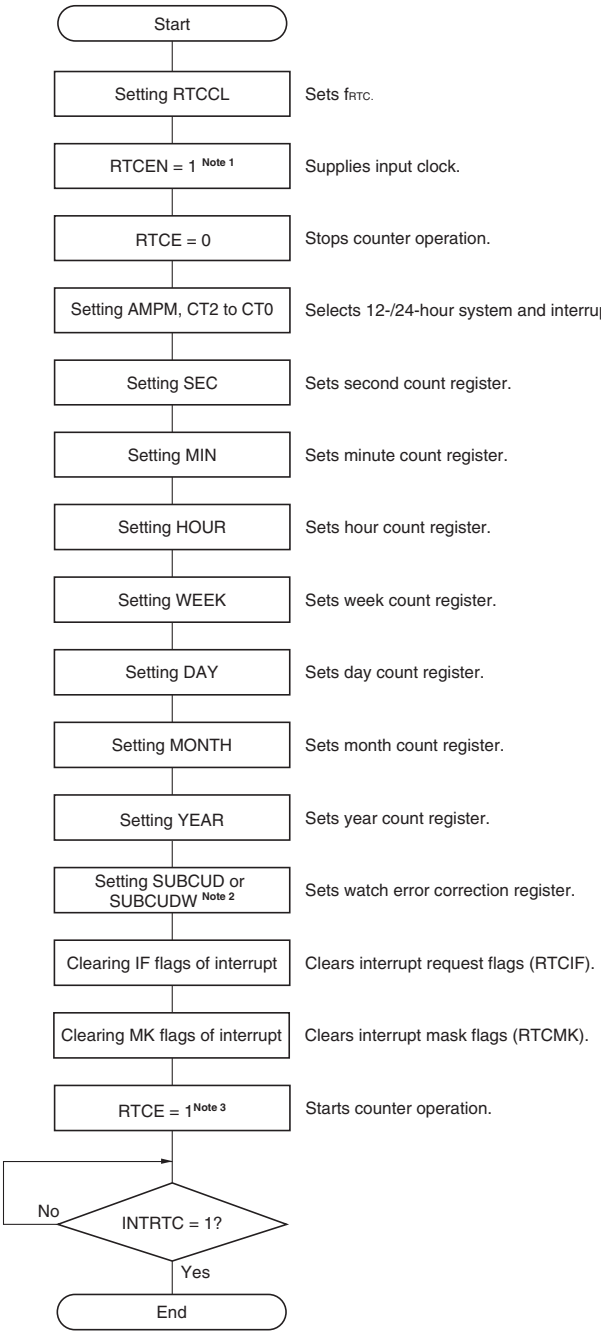
Here is an example of setting the alarm.

| Time of Alarm | Day | | | | | | | 12-Hour Display | | | | 24-Hour Display | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sunday<br>WW0 | Monday<br>WW1 | Tuesday<br>WW2 | Wednesday<br>WW3 | Thursday<br>WW4 | Friday<br>WW5 | Saturday<br>WW6 | Hour 10 | Hour 1 | Minute 10 | Minute 1 | Hour 10 | Hour 1 | Minute 10 | Minute 1 |
| Every day, 0:00 a.m. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| Every day, 1:30 a.m. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 3 | 0 | 0 | 1 | 3 | 0 |
| Every day, 11:59 a.m. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 5 | 9 | 1 | 1 | 5 | 9 |
| Monday through Friday, 0:00 p.m. | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 3 | 2 | 0 | 0 | 1 | 2 | 0 | 0 |
| Sunday, 1:30 p.m. | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 3 | 0 | 1 | 3 | 3 | 0 |
| Monday, Wednesday, Friday, 11:59 p.m. | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 3 | 1 | 5 | 9 | 2 | 3 | 5 | 9 |

## 9.4 Real-time Clock Operation

### 9.4.1 Starting operation of real-time clock

**Figure 9-21. Procedure for Starting Operation of Real-time Clock**




**Notes 1.** First set the RTCEN bit to 1, while oscillation of the input clock ($f_{RTC}$) is stable.

**2.** Set up the SUBCUD register only if the watch error must be corrected. Set up the SUBCUDW register if the watch must be corrected with high accuracy. For details about how to calculate the correction value, see **9.4.6 Example of watch error correction of real-time clock**.

**3.** Confirm the procedure described in **9.4.2 Shifting to HALT/STOP mode after starting operation** when shifting to HALT/STOP mode without waiting for INTRTC = 1 after RTCE = 1.

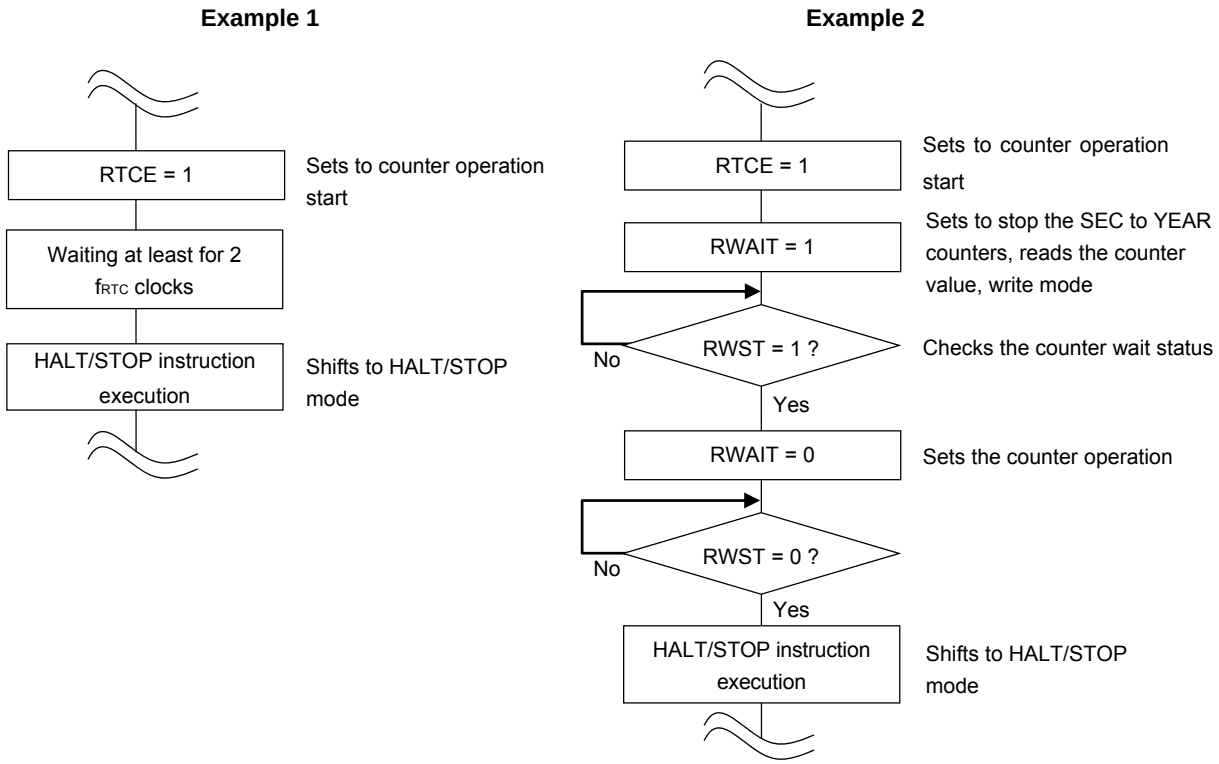
### 9.4.2 Shifting to HALT/STOP mode after starting operation

Perform one of the following processing when shifting to HALT/STOP mode immediately after setting the RTCE bit to 1.

However, after setting the RTCE bit to 1, this processing is not required when shifting to HALT/STOP mode after INTRTC interrupt has occurred.

- Shifting to HALT/STOP mode when at least two operating clocks ($f_{RTC}$) have elapsed after setting the RTCE bit to 1 (see **Figure 9-20**, **Example 1**).
- Checking by polling the RWST bit to become 1, after setting the RTCE bit to 1 and then setting the RWAIT bit to 1. Afterward, setting the RWAIT bit to 0 and shifting to HALT/STOP mode after checking again by polling that the RWST bit has become 0 (see **Figure 9-20**, **Example 2**).

**Figure 9-22. Procedure for Shifting to HALT/STOP Mode After Setting RTCE bit to 1**

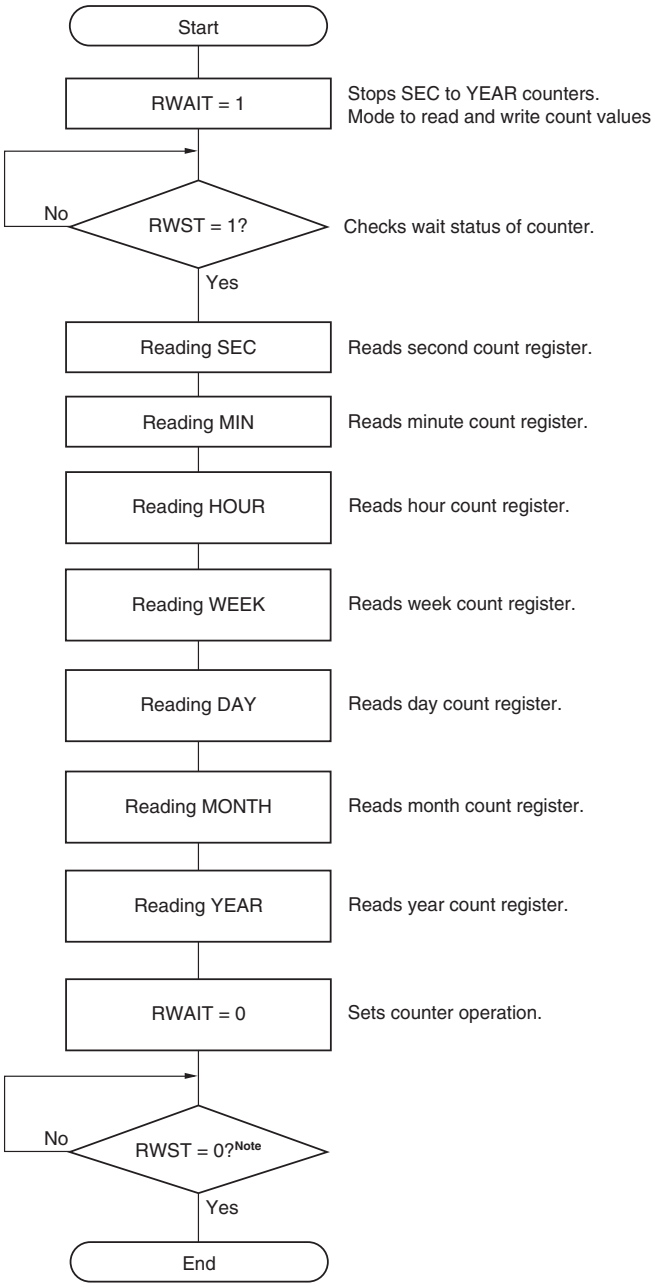


### 9.4.3 Reading/writing real-time clock

Read or write the counter after setting 1 to RWAIT first.

Set RWAIT to 0 after completion of reading or writing the counter.

**Figure 9-23. Procedure for Reading Real-time Clock**



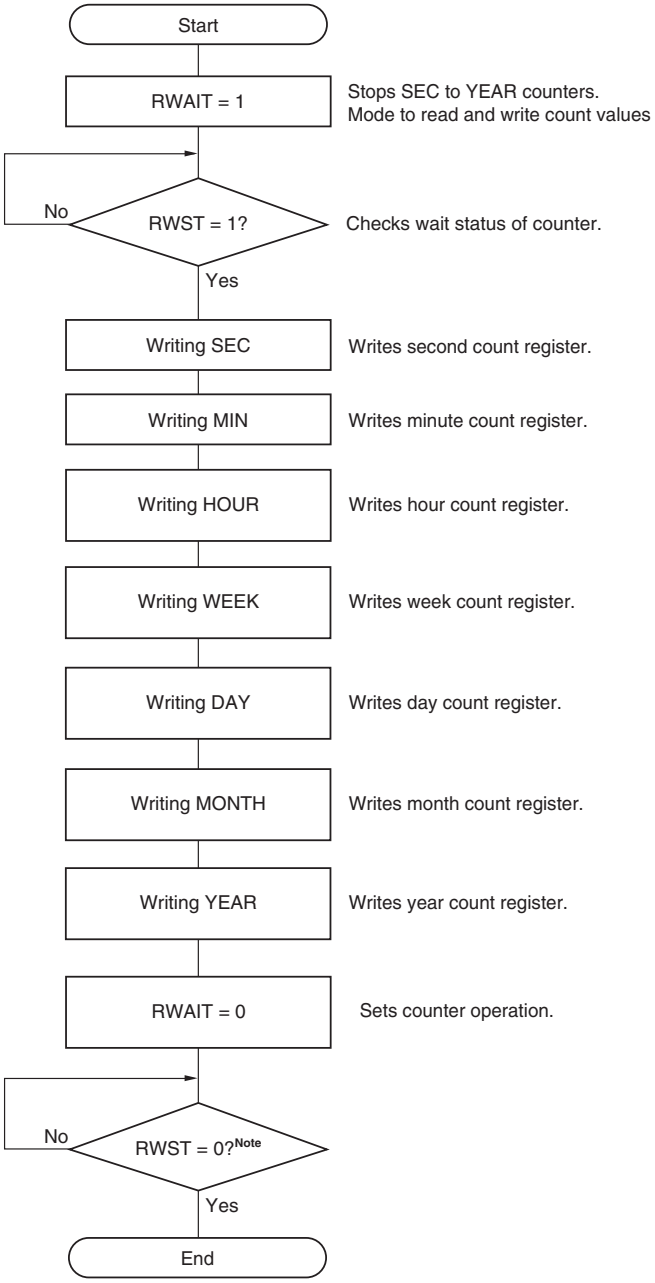

**Note** Be sure to confirm that RWST = 0 before setting HALT/STOP mode.

**Caution Complete the series of process of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.**

**Remark** The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence.

All the registers do not have to read and only some registers may be read.

**Figure 9-24. Procedure for Writing Real-time Clock**




**Note** Be sure to confirm that RWST = 0 before setting HALT/STOP mode.

**Cautions 1. Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.**

**2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counter operates (RTCE = 1), rewrite the values of the MIN register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the MIN register.**

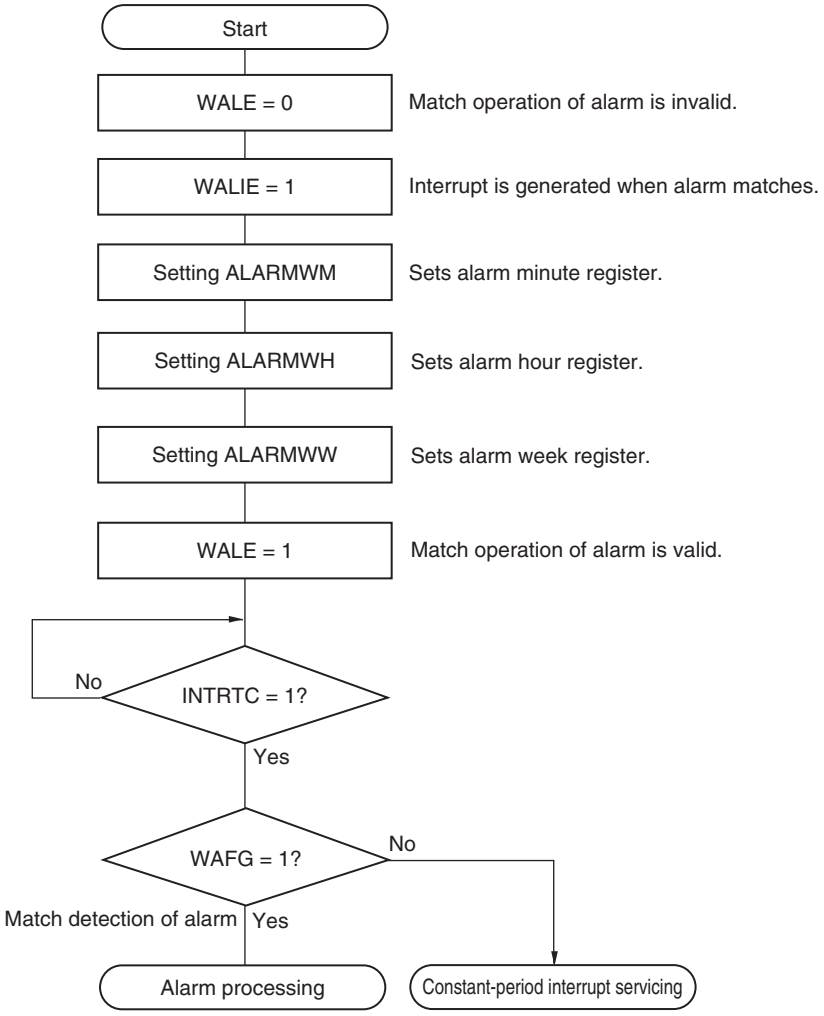
**Remark** The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be written in any sequence.

All the registers do not have to be set and only some registers may be written.

### 9.4.4 Setting alarm of real-time clock

Set time of alarm after setting 0 to WALE first.

**Figure 9-25. Alarm Setting Procedure**

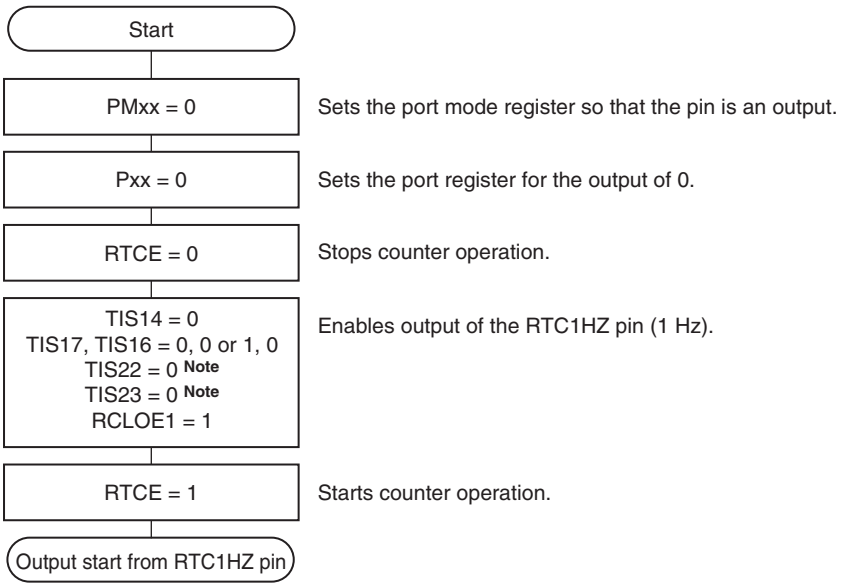



**Remarks 1.** The alarm week register (ALARMWW), alarm hour register (ALARMWH), and alarm week register (ALARMWW) may be written in any sequence.

**2.** Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

### 9.4.5 1 Hz output of real-time clock

**Figure 9-26. 1 Hz Output Setting Procedure**




**Note** The timer input select register 2 (TIS2) is only available in the RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory.

**Caution First set the RTCEN bit to 1 while oscillation of the input clock ($f_{RTC}$) is stable.**

### 9.4.6 Example of watch error correction of real-time clock

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the 16-bit watch error correction register (SUBCUDW).

#### (1) Example of calculating the correction value

The correction value used when correcting the count value of the internal counter (16 bits) is calculated by using the following expression.

Set the DEV bit to 0 when the correction range is –4165.6 ppm or less, or 4165.6 ppm or more.

(When DEV = 0)

Correction value[Note] = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60 ÷ 3

(When DEV = 1)

Correction value[Note] = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60

**Note** The correction value is the watch error correction value calculated by using bits 12 to 0 of the 16-bit watch error correction register (SUBCUDW).

(When F12 = 0) Correction value = {(F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) – 1} × 2

(When F12 = 1) Correction value = – {(/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *), watch error correction is not performed. "*" is 0 or 1.

/F11 to /F0 are bit-inverted values (000000000011 when 111111111100).

**Remarks 1.** The correction value is 2, 4, 6, 8, … 8186, 8188 or –2, –4, –6, –8, … –8186, –8188.

**2.** The oscillation frequency is the input clock ($f_{RTC}$).

It can be calculated from the output frequency of the RTC1HZ pin × 32768 when the 16-bit watch error correction register (SUBCUDW) is set to its initial value (0000H).

**3.** The target frequency is the frequency resulting after correction performed by using the 16-bit watch error correction register (SUBCUDW).

**(2) Correction example**

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

The oscillation frequency Note 1 of each product is measured by outputting about 1 Hz from the RTC1HZ pin when the watch error correction register (SUBCUD, SUBCUDW) is set to its initial value (0000H). The frequency can also be measured by selecting Note 2 RTC1HZ for the input of timer array unit.

**Notes 1.** See **9.4.5 1 Hz output of real-time clock** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin. For input selection of timer array unit, see **6.3.9 Timer input select register 1 (TIS1)** and **6.3.10 Timer input select register 2 (TIS2).**

**2.** The RTC1HZ signal is not output from the RTC1HZ pin when the RTC1HZ output signal is selected for the input to timer array unit by the setting of the timer input select register 1 or 2 (TIS1 or TIS2).

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 0.9999817 Hz)

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

Correction value = Number of correction counts in 1 minute
= (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60
= (32767.4 ÷ 32768 − 1) × 32768 × 60
= −36

[Calculating the values to be set to (F12 to F0)]

(When the correction value is −36)

If the correction value is 0 or less (when quickening), assume F12 to be 1.

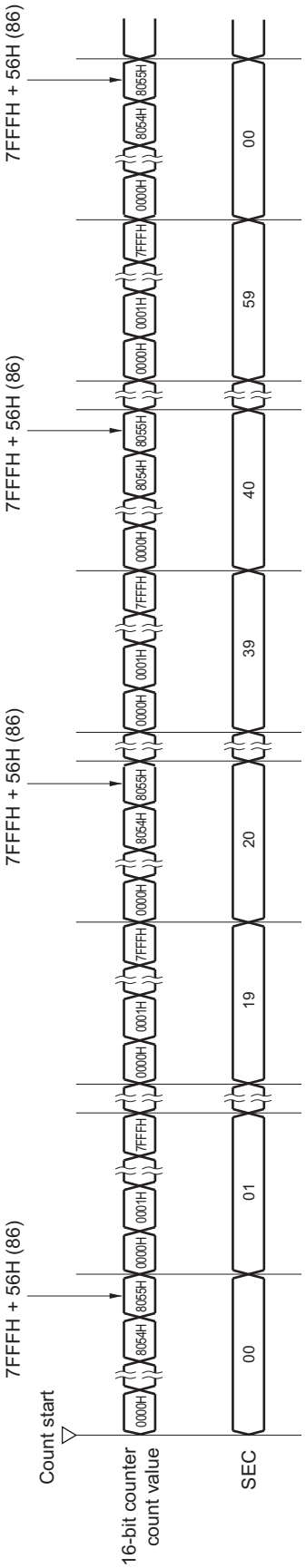
Calculate (F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) from the correction value.

− {(/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0) − 1} × 2 = −36
(/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0) = 17
(/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0) = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1)
(F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0)

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is −36 (bits 12 to 0 of the SUBCUDW register: 1111111101110) results in 32768 Hz (0 ppm).

Figure 9-25 shows the operation when (DEV, F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0).

**Figure 9-27. Operation when (DEV, F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0)**



# CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

Whether the output pin for the clock and buzzer output controller is present depends on the product.

| Output pin | 20, 30, and 32-pin products | 48, 64, 80, and 100-pin products |
|---|---|---|
| PCLBUZ0 | — | √ |

**Caution Most of the following descriptions in this chapter use the 80-pin as an example.**

## 10.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral ICs.

Buzzer output is a function to output a square wave of buzzer frequency.

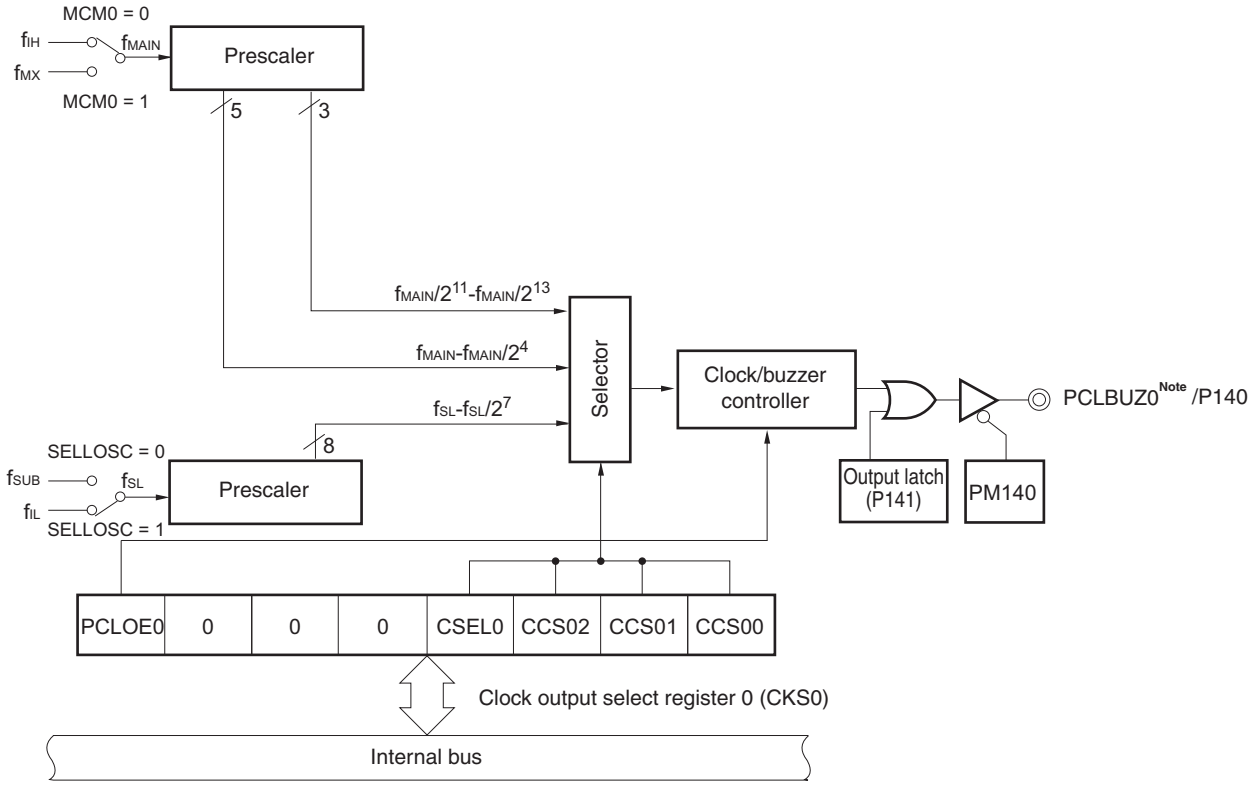
One pin can be used to output a clock or buzzer sound.

The PCLBUZ0 pin outputs a clock selected by clock output select register 0 (CKS0).

Figure 10-1 shows the block diagram of clock output/buzzer output controller.

**Caution In the low-consumption RTC mode (when the RTCLPC bit of the operation speed mode control register (OSMC) = 1), it is not possible to output the sub/low-speed on-chip oscillator select clock ($f_{SL}$) from the PCLBUZ0 pin.**

**Figure 10-1. Block Diagram of Clock Output/Buzzer Output Controller**




**Note** For output frequencies available from PCLBUZ0, refer to the AC characteristics in **CHAPTER 34** to **CHAPTER 36 ELECTRICAL SPECIFICATIONS**.

## 10.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

**Table 10-1. Configuration of Clock Output/Buzzer Output Controller**

| Item | Configuration |
|---|---|
| Control registers | Clock output select registers 0 (CKS0)<br>Port mode register 14 (PM14)<br>Port register 14 (P14) |

## 10.3 Registers Controlling Clock Output/Buzzer Output Controller

The following registers are used to control the clock output/buzzer output controller.

- Clock output select registers 0 (CKS0)
- Port mode register 14 (PM14)
- Port register 14 (P14)

### 10.3.1 Clock output select register 0 (CKS0)

These registers set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZ0), and set the output clock.

Select the clock to be output from the PCLBUZ0 pin by using the CKS0 register.

The CKS0 register are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 10-2. Format of Clock Output Select Register 0 (CKS0)**

Address: FFFA5H (CKS0), FFFA6H (CKS1) After reset: 00H R/W

| Symbol | <7> | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CKS0 | PCLOE0 | 0 | 0 | 0 | CSEL0 | CCS02 | CCS01 | CCS00 |

| PCLOE0 | PCLBUZ0 pin output enable/disable specification |
|---|---|
| 0 | Output disable (default) |
| 1 | Output enable |

| CSEL0 | CCS02 | CCS01 | CCS00 | PCLBUZ0 pin output clock selection | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | $f_{MAIN}$ = 5 MHz | $f_{MAIN}$ = 10 MHz | $f_{MAIN}$ = 20 MHz | $f_{MAIN}$ = 32 MHz | $f_{MAIN}$ = 48 MHz | $f_{MAIN}$ = 64 MHz |
| 0 | 0 | 0 | 0 | $f_{MAIN}$ | 5 MHz | 10 MHz Note | Setting prohibited Note | Setting prohibited Note | Setting prohibited Note | Setting prohibited Note |
| 0 | 0 | 0 | 1 | $f_{MAIN}/2$ | 2.5 MHz | 5 MHz | 10 MHz Note | 16 MHz Note | Setting prohibited Note | Setting prohibited Note |
| 0 | 0 | 1 | 0 | $f_{MAIN}/2^2$ | 1.25 MHz | 2.5 MHz | 5 MHz | 8 MHz | 12 MHz Note | 16 MHz Note |
| 0 | 0 | 1 | 1 | $f_{MAIN}/2^3$ | 625 kHz | 1.25 MHz | 2.5 MHz | 4 MHz | 6 MHz | 8 MHz |
| 0 | 1 | 0 | 0 | $f_{MAIN}/2^4$ | 312.5 kHz | 625 kHz | 1.25 MHz | 2 MHz | 3 MHz | 4 MHz |
| 0 | 1 | 0 | 1 | $f_{MAIN}/2^{11}$ | 2.44 kHz | 4.88 kHz | 9.76 kHz | 15.62 kHz | 23.43 kHz | 31.25 kHz |
| 0 | 1 | 1 | 0 | $f_{MAIN}/2^{12}$ | 1.22 kHz | 2.44 kHz | 4.88 kHz | 7.81 kHz | 11.71 kHz | 15.12 kHz |
| 0 | 1 | 1 | 1 | $f_{MAIN}/2^{13}$ | 610 Hz | 1.22 kHz | 2.44 kHz | 3.90 kHz | 5.85 kHz | 7.81 kHz |
| 1 | 0 | 0 | 0 | $f_{SL}$ | 32.768 kHz ($f_{SUB}$) or 15 kHz ($f_{IL}$) | | | | | |
| 1 | 0 | 0 | 1 | $f_{SL}/2$ | 16.384 kHz ($f_{SUB}$) or 7.5 kHz ($f_{IL}$) | | | | | |
| 1 | 0 | 1 | 0 | $f_{SL}/2^2$ | 8.192 kHz ($f_{SUB}$) or 3.75 kHz ($f_{IL}$) | | | | | |
| 1 | 0 | 1 | 1 | $f_{SL}/2^3$ | 4.096 kHz ($f_{SUB}$) or 1.87 kHz ($f_{IL}$) | | | | | |
| 1 | 1 | 0 | 0 | $f_{SL}/2^4$ | 2.048 kHz ($f_{SUB}$) or 937.5 Hz ($f_{IL}$) | | | | | |
| 1 | 1 | 0 | 1 | $f_{SL}/2^5$ | 1.024 kHz ($f_{SUB}$) or 468.75 Hz ($f_{IL}$) | | | | | |
| 1 | 1 | 1 | 0 | $f_{SL}/2^6$ | 512 Hz ($f_{SUB}$) or 234.37 Hz ($f_{IL}$) | | | | | |
| 1 | 1 | 1 | 1 | $f_{SL}/2^7$ | 256 Hz ($f_{SUB}$) or 117.18 Hz ($f_{IL}$) | | | | | |

**Note** Use the output clock within a range of 16 MHz. See the AC characteristics in **CHAPTER 34** to **CHAPTER 36 ELECTRICAL SPECIFICATIONS** for details.

**Cautions 1. Change the output clock and the CSEL0 and CCS02 to CCS00 bits after disabling clock output (PCLOE0 = 0).**

**2. To shift to STOP mode when the main system clock is selected (CSEL0 = 0), set PCLOE0 = 0 before executing the STOP instruction. When the subsystem clock is selected (CSEL0 = 1), PCLOE0 = 1 can be set because the clock can be output in STOP mode.**

**3. In the low-consumption RTC mode (when the RTCLPC bit of the operation speed mode control register (OSMC) = 1), it is not possible to output the subsystem/low-speed on-chip oscillator select clock ($f_{SL}$) from the PCLBUZ0 pin.**

4. **The high-speed on-chip oscillator clock ($f_{IH}$) and the high-speed system clock ($f_{MX}$) can be selected as the main system clock ($f_{MAIN}$) by the setting of the MCM0 bit (bit 4 of the system clock control register (CKC)). For details, refer to CHAPTER 5 CLOCK GENERATOR.**
5. **The subsystem clock ($f_{SUB}$) and the low-speed on-chip oscillator clock ($f_{IL}$) can be selected as the sub/low-speed on-chip oscillator select clock by the setting of the SELLOSC bit (bit 0 of the clock select register (CKSEL)). For details, refer to CHAPTER 5 CLOCK GENERATOR.**

### 10.3.2 Clock Select Register (CKSEL)

This register is used to select the CPU clock ($f_{SUB}$/$f_{IL}$) the clocks for the timer RJ, timer RD, and clock output/buzzer output. Together with the CMC register, the SELLOSC bit is used to set the operation mode of the subsystem clock. For details, see **Figure 5-3 Format of Clock Operation Mode Control Register (CMC)**.

Set the CKSEL register by a 1-bit or 8-bit memory manipulation instruction.

Writing to the CKSEL register is disabled when the GCSC bit of the IAWCTL register is set to 1.

**Figure 10-3. Format of Clock Select Register (CKSEL)**

Address: F02C4H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| CKSEL | 0 | 0 | 0 | 0 | 0 | TRD_CKSEL | 0 | SELLOSC Notes 3, 4 |

| SELLOSC Notes 3, 4 | Control of sub/low-speed on-chip oscillator selection clock ($f_{SL}$) selection |
|---|---|
| 0 | Selects $f_{SUB}$ Note 1 |
| 1 | Selects $f_{IL}$ Note 2 |

**Notes 1.** When setting $f_{SUB}$ as the CPU/peripheral hardware clock, first set the SELLOSC bit to 0 and then set the CSS bit in the CKC register to 1.

**2.** When setting $f_{IL}$ as the CPU/peripheral hardware clock, first set the SELLOSC bit to 1 and then set the CSS bit in the CKC register to 1.

**3.** When the SELLOSC bit is set to 1, the low-speed on-chip oscillator operates.

**4.** When setting the CKSEL register in the 20-, 30-, or 32-pin products, set the SELLOSC bit to 1.

### 10.3.3 Port mode register 14 (PM14)

These registers set input/output of port in 1-bit units.

When using the P140/PCLBUZ0 pins for clock output and buzzer output, clear PM140 bit and the output latch of P140 to 0.

The PM14 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

**Figure 10-4. Format of Port Mode Register 14 (PM14)**

Address: FFF2EH After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM14 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | PM140 |

| PMmn | Pmn pin I/O mode selection (mn = 140) |
|---|---|
| 0 | Output mode (output buffer on) |
| 1 | Input mode (output buffer off) |

## 10.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.

The PCLBUZ0 pin outputs a clock/buzzer selected by the clock output select register 0 (CKS0).

### 10.4.1 Operation as output pin

The PCLBUZ0 pin is output as the following procedures.

<1> Set the bit in the port mode register 14 (PM14) and the bit in the port register 14 (P14) corresponding to the port used as a PCLBUZ0 pin.

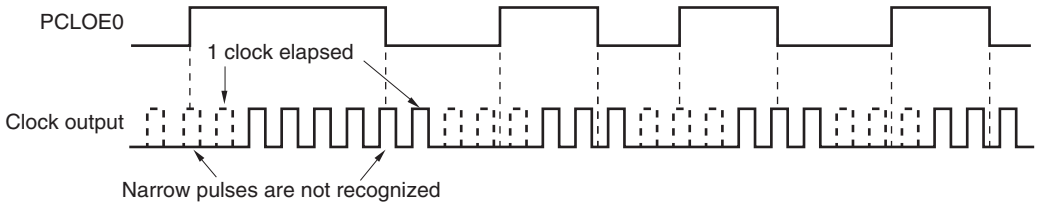
<2> Clear the output latches of PM140 and P140 to 0.

<3> Select the output clock with bits 0 to 3 (CCS00 to CCS02, CSEL0) of the clock output select register 0 (CKS0) of the PCLBUZ0 pin (output in disabled status).

<4> Set bit 7 (PCLOE0) of the CKS0 register to 1 to enable clock/buzzer output.

**Remark** The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOE0 bit) is switched. At this time, pulses with a narrow width are not output. Figure 10-5 shows enabling or stopping output using the PCLOE0 bit and the timing of outputting the clock.

**Figure 10-5. Remote Control Output Application Example**




## 10.5 Notes on Clock Output/Buzzer Output Controller

When the CPU enters STOP mode within 1.5 clock cycles of main system clock after the setting to disable output is made (PCLOE0 = 0) while the main system clock is selected for PCLBUZ0 output (CSEL0 = 0), the pulse width of the PCLBUZ0 output is narrowed.

# CHAPTER 11 WATCHDOG TIMER

## 11.1 Functions of Watchdog Timer

The watchdog timer operates on the WDT-dedicated low-speed on-chip oscillator clock ($f_{WDT}$).

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register
- If data is written to the WDTE register during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDCLRF) of the reset control flag register (RESF) is set to 1. For details of the RESF register, see **CHAPTER 24 RESET FUNCTION**.

When 75% of the overflow time + 1/2 $f_{WDT}$ is reached, an interval interrupt can be generated.

## 11.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

**Table 11-1. Configuration of Watchdog Timer**

| Item | Configuration |
|---|---|
| Counter | Internal counter (17 bits) |
| Control register | Watchdog timer enable register (WDTE) |

How the internal counter (17 bits) operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

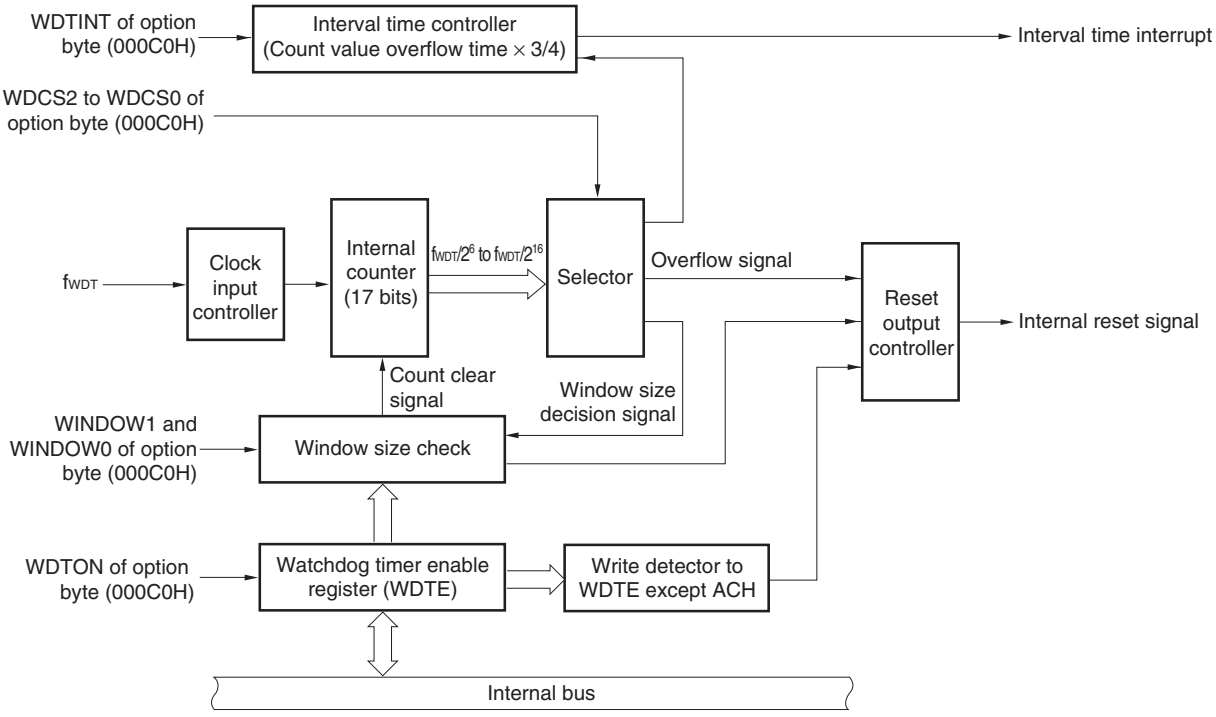
**Caution Set the same value as 000C0H to 020C0H when the boot swap operation is used because 000C0H is replaced by 020C0H.**

**Table 11-2. Setting of Option Bytes and Watchdog Timer**

| Setting of Watchdog Timer | Option Byte (000C0H) |
|---|---|
| Watchdog timer interval interrupt | Bit 7 (WDTINT) |
| Window open period | Bits 6 and 5 (WINDOW1, WINDOW0) |
| Controlling counter operation of watchdog timer | Bit 4 (WDTON) |
| Overflow time of watchdog timer | Bits 3 to 1 (WDCS2 to WDCS0) |
| Controlling counter operation of watchdog timer (in HALT/STOP/SNOOZE mode) | Bit 0 (WDSTBYON) |

**Remark** For the option byte, see **CHAPTER 29 OPTION BYTE**.

**Figure 11-1. Block Diagram of Watchdog Timer**



## 11.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

### 11.3.1 Watchdog timer enable register (WDTE)

When the WDTON bit in the option byte (000C0H) is 1, writing ACH to the WDTE register clears the watchdog timer counter and starts counting again.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 1AH or 9AH[Note].

**Figure 11-2. Format of Watchdog Timer Enable Register (WDTE)**

Address: FFFABH After reset: 1AH/9AH[Note] R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| WDTE | | | | | | | | |

**Note** The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

| WDTON Bit Setting Value | WDTE Register Reset Value |
|---|---|
| 0 (watchdog timer count operation disabled) | 1AH |
| 1 (watchdog timer count operation enabled) | 9AH |

**Cautions 1. When the WDTON bit in the option byte (000C0H) is 1, if a value other than "ACH" is written to the WDTE register, an internal reset signal is generated.**

**2. When the WDTON bit in the option byte (000C0H) is 1, if a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.**

**3. The value read from the WDTE register is 1AH/9AH (this differs from the written value (ACH) as specified in the WDTON bit of the option byte (000C0H)).**

## 11.4 Operation of Watchdog Timer

### 11.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 29 OPTION BYTE**).

| WDTON | Watchdog Timer Counter |
|---|---|
| 0 | Counter operation disabled (counting stopped after reset) |
| 1 | Counter operation enabled (counting started after reset) |

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **11.4.2 Setting overflow time of watchdog timer** and **CHAPTER 29 OPTION BYTE**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **11.4.3 Setting window open period of watchdog timer** and **CHAPTER 29 OPTION BYTE**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer counter is cleared and starts counting again.
4. After that, write the WDTE register the second time or later after a reset release during the window open period. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to the WDTE register, an internal reset signal is generated.
   An internal reset signal is generated in the following cases.

   - If a 1-bit manipulation instruction is executed on the WDTE register
   - If data other than “ACH” is written to the WDTE register

**Cautions 1. When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer counter is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer counter starts counting again.**

**2. If the watchdog timer counter is cleared by writing “ACH” to the WDTE register, the actual overflow time may be different from the overflow time set by the option byte by up to $2/f_{IL}$ seconds.**

**3. The watchdog timer counter can be cleared immediately before the count value overflows.**

4. **The operation of the watchdog timer in the HALT, STOP, and SNOOZE modes differs as follows depending on the set value of bit 0 (WDSTBYON) and bit 4 (WDTON) of the option byte (000C0H).**

| | WDTON = 1 and WDSTBYON = 0 | WDTON = 1 and WDSTBYON = 1 |
|---|---|---|
| In HALT mode | Watchdog timer operation stops. | Watchdog timer operation continues. |
| In STOP mode | | |
| In SNOOZE mode | | |

**If WDSTBYON = 0, the watchdog timer resumes counting after the HALT, STOP, or SNOOZE modes is released. At this time, the counter is cleared to 0 and counting starts.**
**When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.**
**Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.**
**Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer counter is to be cleared after the STOP mode release by an interval interrupt.**

### 11.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to the watchdog timer enable register (WDTE) during the window open period before the overflow time.

The following overflow times can be set.

**Table 11-3. Setting of Overflow Time of Watchdog Timer**

| WDCS2 | WDCS1 | WDCS0 | Overflow Time of Watchdog Timer ($f_{WDT}$ = 17.25 kHz (MAX.)) |
|---|---|---|---|
| 0 | 0 | 0 | $2^6/f_{WDT}$ (3.71 ms) |
| 0 | 0 | 1 | $2^7/f_{WDT}$ (7.42 ms) |
| 0 | 1 | 0 | $2^8/f_{WDT}$ (14.84 ms) |
| 0 | 1 | 1 | $2^9/f_{WDT}$ (29.68 ms) |
| 1 | 0 | 0 | $2^{11}/f_{WDT}$ (118.72 ms) |
| 1 | 0 | 1 | $2^{13}/f_{WDT}$ Note (474.89 ms) |
| 1 | 1 | 0 | $2^{14}/f_{WDT}$ Note (949.79 ms) |
| 1 | 1 | 1 | $2^{16}/f_{WDT}$ Note (3799.18 ms) |

<R> **Note** When the interval interrupt of watchdog timer is used, do not set the overflow time to $2^{13}/f_{WDT}$, $2^{14}/f_{WDT}$ or $2^{16}/f_{WDT}$.

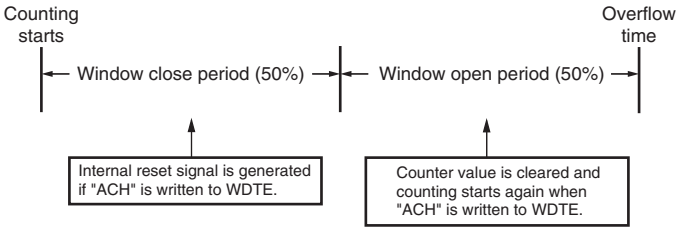
**Remark** $f_{WDT}$: WDT-dedicated low-speed on-chip oscillator clock frequency

### 11.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If “ACH” is written to the watchdog timer enable register (WDTE) during the window open period, the watchdog timer counter is cleared and starts counting again.
- Even if “ACH” is written to the WDTE register during the window close period, an abnormality is detected and an internal reset signal is generated.

**Example**: If the window open period is 50%




**Caution When data is written to the WDTE register for the first time after reset release, the watchdog timer counter is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer counter starts counting again.**

The window open period can be set is as follows.

**Table 11-4. Setting Window Open Period of Watchdog Timer**

| WINDOW1 | WINDOW0 | Window Open Period of Watchdog Timer |
|---|---|---|
| 0 | 0 | Setting prohibited |
| 0 | 1 | 50% |
| 1 | 0 | 75% |
| 1 | 1 | 100% |

**Caution When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of the WINDOW1 and WINDOW0 bits.**

**Remark** If the overflow time is set to $2^9/f_{WDT}$, the window close time and open time are as follows.

| | Setting of Window Open Period | | |
|---|---|---|---|
| | 50% | 75% | 100% |
| Window close time | 0 to 20.08 ms | 0 to 10.04 ms | None |
| Window open time | 20.08 to 29.68 ms | 10.04 to 29.68 ms | 0 to 29.68 ms |

<When window open period is 50%>

- Overflow time:
  $2^9/f_{WDT}$ (MAX.) = $2^9$/17.25 kHz (MAX.) = 29.68 ms
- Window close time:
  0 to $2^9/f_{WDT}$ (MIN.) × (1 − 0.5) = 0 to $2^9$/12.75 kHz × 0.5 = 0 to 20.08 ms
- Window open time:
  $2^9/f_{WDT}$ (MIN.) × (1 − 0.5) to $2^9/f_{WDT}$ (MAX.) = $2^9$/12.75 kHz × 0.5 to $2^9$/17.25 kHz
  = 20.08 to 29.68 ms

### 11.4.4 Setting watchdog timer interval interrupt

Setting bit 7 (WDTINT) of an option byte (000C0H) can generate an interval interrupt (INTWDTI) when 75% of the overflow time is reached.

**Table 11-5. Setting of Watchdog Timer Interval Interrupt**

| WDTINT | Use of Watchdog Timer Interval Interrupt |
|---|---|
| 0 | Interval interrupt is not used. |
| 1 | Interval interrupt is generated when 75% of overflow time + 1/2 $f_{WDT}$ is reached. |

**Caution When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.**
**Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.**
**Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.**

**Remark** The watchdog timer continues counting even after INTWDTI is generated (until ACH is written to the watchdog timer enable register (WDTE)). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

# CHAPTER 12 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

| Pin Count | | 100 Pins | 80 Pins | | 64 Pins | | 48 Pins | | 32 Pins | 30 Pins | 20 Pins |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Analog input channel | RL78/F13 (LIN incorporated) | | | 20 ch ANI0 to ANI15, ANI24 to ANI27 | 19 ch ANI0 to ANI15, ANI24 to ANI26 | 12 ch ANI0 to ANI11 | 15 ch ANI0 to ANI12, ANI24, ANI25 | 12 ch ANI0 to ANI11 | 8 ch ANI0 to ANI7 | 10 ch ANI0 to ANI9 | 4 ch ANI0 to ANI3 |
| | Code flash | — | — | 64 KB or more | 96 KB or more | 64 KB or less | 96 KB or more | 64 KB or less | — | — | — |
| | RL78/F13 (CAN and LIN incorporated) | | | 20 ch ANI0 to ANI15, ANI24 to ANI27 | 19 ch ANI0 to ANI15, ANI24 to ANI26 | | 15 ch ANI0 to ANI12, ANI24, ANI25 | | 10 ch ANI0 to ANI7, ANI24, ANI25 | 12 ch ANI0 to ANI9, ANI24, ANI25 | |
| | Code flash | — | — | — | — | — | — | — | — | — | — |
| | RL78/F14 | 31 ch ANI0 to ANI23, ANI24 to ANI30 | 25 ch ANI0 to ANI17, ANI24 to ANI30 | 20 ch ANI0 to ANI15, ANI24 to ANI27 | 20 ch ANI0 to ANI16, ANI24 to ANI26 | 19 ch ANI0 to ANI15, ANI24 to ANI26 | 18 ch ANI0 to ANI12, ANI24 to ANI28 | 15 ch ANI0 to ANI12, ANI24, ANI25 | 10 ch ANI0 to ANI7, ANI24, ANI25 | 12 ch ANI0 to ANI9, ANI24, ANI25 | |
| | Code flash | — | 128 KB or more | 96 KB or less | 128 KB or more | 96 KB or less | 128 KB or more | 96 KB or less | — | — | — |

## 12.1 Function of A/D Converter

The A/D converter is a 10-bit resolution[Note] converter that converts analog input signals into digital values, and is configured to control analog inputs, including up to 31 channels of A/D converter analog inputs (ANI0 to ANI23 and ANI24 to ANI30).

The A/D converter has the following function.

- **10-bit resolution A/D conversion[Note]**

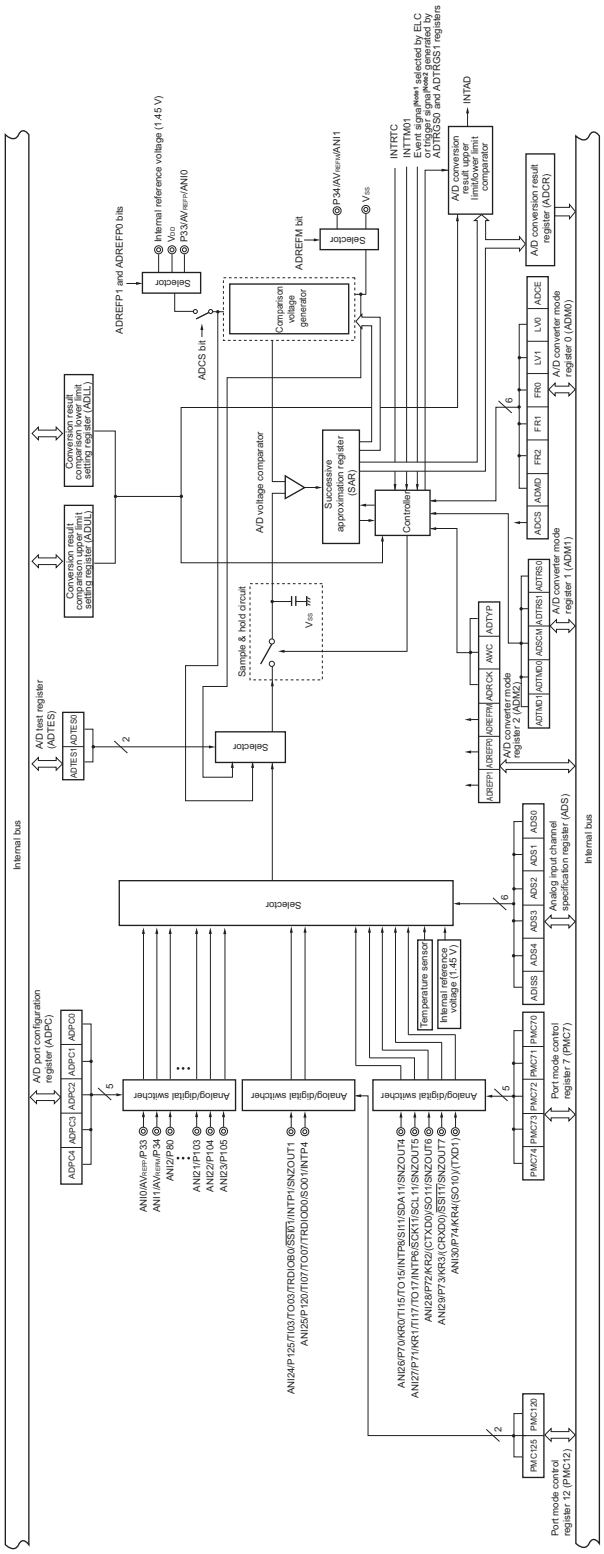
  10-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI23 and ANI24 to ANI30. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated (when in the select mode).

**Note** 8-bit resolution can also be selected by using the ADTYP bit of A/D converter mode register 2 (ADM2).

Various A/D conversion modes can be specified by using the mode combinations below.

| Trigger Mode | Channel Selection Mode | Conversion Operation Mode |
|---|---|---|
| • Software trigger<br>Conversion is started by specifying a software trigger.<br>• Hardware trigger no-wait mode<br>Conversion is started by detecting a hardware trigger.<br>• Hardware trigger wait mode<br>The power is turned on by detecting a hardware trigger while the system is off and in the conversion standby state, and conversion is then started automatically after the stabilization wait time passes. | • Select mode<br>A/D conversion is performed on the analog input of one channel.<br>• Scan mode<br>A/D conversion is performed on the analog input of four channels in order. | • One-shot conversion mode<br>A/D conversion is performed on the selected channel once.<br>• Sequential conversion mode<br>A/D conversion is sequentially performed on the selected channels until it is stopped by software. |

**Figure 12-1. Block Diagram of A/D Converter**




**Notes 1.** Only available in the RL78/F14.

**2.** Only available in the RL78/F13.

**Remark** Analog input pin for figure 12-1 when a 100-pin product of RL78/F14 is used.

## 12.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

**(1) ANI0 to ANI23 ($V_{DD}$) and ANI24 to ANI30 ($EV_{DD}$) pins**

These are the analog input pins of the twenty channels of the A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

<R> **Caution If the supply setting other than $AV_{REFP}$, $AV_{REFM}$ is used as the reference voltage of the A/D converter, the conversion accuracy decreases. In addition, since the $EV_{DD}$ system analog pins have lower accuracy than the $V_{DD}$ system analog pins, the $V_{DD}$ analog pins should be used for highly accurate conversion.**

**(2) Sample & hold circuit**

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

**(3) A/D voltage comparator**

This A/D voltage comparator compares the voltage generated from the voltage tap of the comparison voltage generator with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 $AV_{REF}$) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 $AV_{REF}$), the MSB bit of the SAR is reset.

After that, bit 8 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the comparison voltage generator is selected by the value of bit 9, to which the result has been already set.

Bit 9 = 0: (1/4 $AV_{REF}$)
Bit 9 = 1: (3/4 $AV_{REF}$)

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 8 = 1
Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.

When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 2 of the SAR register.

**Remark** $AV_{REF}$: The + side reference voltage of the A/D converter. This can be selected from $AV_{REFP}$, the internal reference voltage (1.45 V), and $V_{DD}$.

**(4) Comparison voltage generator**

The comparison voltage generator generates the comparison voltage input from an analog input pin.

**(5) Successive approximation register (SAR)**

The SAR register is a register that sets voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

**(6) 10-bit A/D conversion result register (ADCR)**

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

**(7) 8-bit A/D conversion result register (ADCRH)**

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

**(8) Controller**

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD.

**(9) $AV_{REFP}$ pin**

This pin inputs an external reference voltage ($AV_{REFP}$).

If using $AV_{REFP}$ as the + side reference voltage of the A/D converter, set the ADREFP1 bit of A/D converter mode register 2 (ADM2) to 0 and the ADREFP0 bit to 1.

The analog signals input to ANI0 to ANI23 and ANI24 to ANI30 are converted to digital signals based on the voltage applied between $AV_{REFP}$ and the – side reference voltage ($AV_{REFM}$/$V_{SS}$).

In addition to $AV_{REFP}$, it is possible to select $V_{DD}$ or the internal reference voltage (1.45 V) as the + side reference voltage of the A/D converter.

**(10) $AV_{REFM}$ pin**

This pin inputs an external reference voltage ($AV_{REFM}$). If using $AV_{REFM}$ as the – side reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.

In addition to $AV_{REFM}$, it is possible to select $V_{SS}$ as the – side reference voltage of the A/D converter.

## 12.3 Registers Used in A/D Converter

The A/D converter uses the following registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- A/D port configuration register (ADPC)
- A/D converter trigger select registers 0, 1 (ADTRGS0, ADTRGS1)
- Port mode control registers 7, 9, and 12 (PMC7, PMC9, PMC12)
- Port mode registers 3, 7 to 10, 12 (PM3, PM7 to PM10, PM12)

### 12.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the A/D converter is used, be sure to set bit 5 (ADCEN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 12-2. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

| Symbol | <7> | 6 | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PER0 | RTCEN | 0 | ADCEN | IICA0EN | SAU1EN | SAU0EN | TAU1EN | TAU0EN |

| ADCEN | Control of A/D converter input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by the A/D converter cannot be written.<br>• The A/D converter is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by the A/D converter can be read/written. |

**Cautions 1. When setting the A/D converter, be sure to set the ADCEN bit to 1 first. If ADCEN = 0, writing to a control register of the A/D converter is ignored, and, even if the register is read, only the default value is read (except for port mode registers 3, 7 to 10, 12 (PM3, PM7 to PM10, PM12), port mode control registers 7, 9, and 12 (PMC7, PMC9, PMC12), and A/D port configuration register (ADPC)).**

**2. Be sure to clear the following bits to 0.**
**Bits 1, 3, 4, and 6 in the RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory**
**Bits 4 and 6 in 30-pin products of the RL78/F13 (CAN and LIN incorporated) and in 30-pin products of the RL78/F14**
**Bit 6 in the products other than above**

### 12.3.2 A/D converter mode register 0 (ADM0)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 12-3. Format of A/D Converter Mode Register 0 (ADM0)**

Address: FFF30H After reset: 00H R/W

| Symbol | <7> | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| ADM0 | ADCS | ADMD | FR2 Note 1 | FR1 Note 1 | FR0 Note 1 | LV1 Note 1 | LV0 Note 1 | ADCE |

| ADCS | A/D conversion operation control Note 4 |
|---|---|
| 0 | Stops conversion operation<br>[When read]<br>Conversion stopped/standby status |
| 1 Note 3 | Enables conversion operation<br>[When read Note 2]<br>While in the software trigger mode: Conversion operation status<br>While in the hardware trigger wait mode: Stabilization wait status + conversion operation status |

| ADMD | Specification of the A/D conversion channel selection mode |
|---|---|
| 0 | Select mode |
| 1 | Scan mode |

| ADCE | A/D voltage comparator operation control Note 2 |
|---|---|
| 0 Note 3 | Stops A/D voltage comparator operation |
| 1 | Enables A/D voltage comparator operation |

**Notes 1.** For details of the FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Table 12-3 A/D Conversion Time Selection**.

**2.** While in the software trigger mode or hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 1 μs from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 1 μs or more has elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

**3.** The ADCS bit is not set when 1 is written to it and 0 is written to the ADCE bit. The ADCS bit is read as 0.

**4.** The ADCS bit cannot be used as the flag of conversion operation status in the hardware trigger no-wait mode.

**Cautions 1. Change the ADMD, FR2 to FR0, LV1, LV0, and ADCE bits in the conversion stopped/standby status (ADCS = 0).**

**2. Do not change the ADCS and ADCE bits from 0 to 1 by using an 8-bit manipulation instruction. Be sure to set these bits in the order described in 12.7 A/D Converter Setup Flowchart.**

**Table 12-1. Settings of ADCS and ADCE Bits**

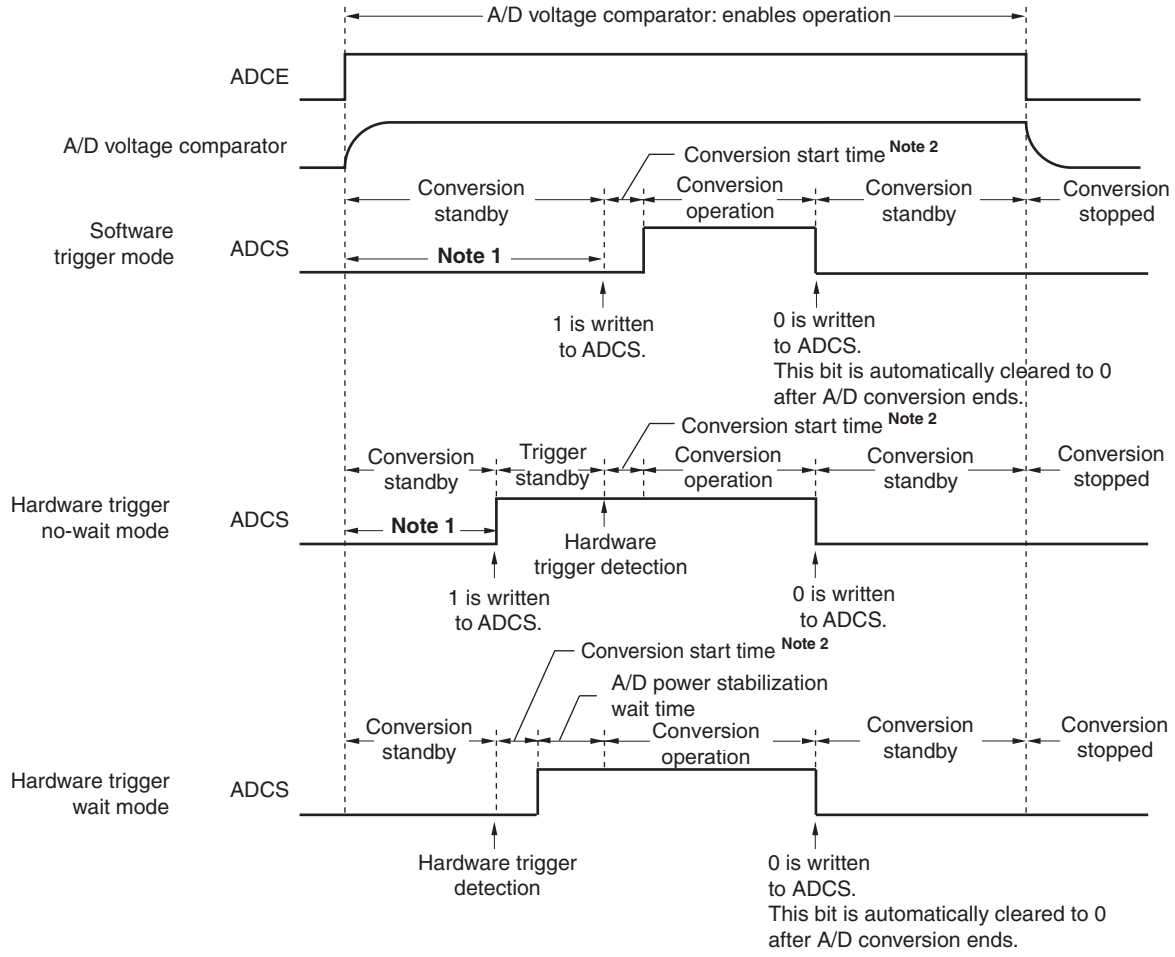
| ADCS | ADCE | A/D Conversion Operation |
|---|---|---|
| 0 | 0 | Stop status (DC power consumption path does not exist) |
| 0 | 1 | Conversion standby mode (only A/D voltage comparator consumes power Note) |
| 1 | 0 | Setting prohibited |
| 1 | 1 | Conversion mode (A/D voltage comparator: enables operation) |

**Note** In hardware trigger wait mode, there is no DC power consumption path even during conversion standby mode.

**Table 12-2. Setting and Clearing Conditions for ADCS Bit**

| A/D Conversion Mode | | | Set Conditions | Clear Conditions |
|---|---|---|---|---|
| Software trigger | Select mode | Sequential conversion mode | When 1 is written to ADCS | When 0 is written to ADCS |
| | | One-shot conversion mode | | • When 0 is written to ADCS<br>• The bit is automatically cleared to 0 when A/D conversion ends. |
| | Scan mode | Sequential conversion mode | | When 0 is written to ADCS |
| | | One-shot conversion mode | | • When 0 is written to ADCS<br>• The bit is automatically cleared to 0 when conversion ends on the specified four channels. |
| Hardware trigger no-wait mode | Select mode | Sequential conversion mode | | When 0 is written to ADCS |
| | | One-shot conversion mode | | When 0 is written to ADCS |
| | Scan mode | Sequential conversion mode | | When 0 is written to ADCS |
| | | One-shot conversion mode | | When 0 is written to ADCS |
| Hardware trigger wait mode | Select mode | Sequential conversion mode | When a hardware trigger is input | When 0 is written to ADCS |
| | | One-shot conversion mode | | • When 0 is written to ADCS<br>• The bit is automatically cleared to 0 when A/D conversion ends. |
| | Scan mode | Sequential conversion mode | | When 0 is written to ADCS |
| | | One-shot conversion mode | | • When 0 is written to ADCS<br>• The bit is automatically cleared to 0 when conversion ends on the specified four channels. |

**Figure 12-4. Timing Chart When A/D Voltage Comparator Is Used**




**Notes 1.** While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer to stabilize the internal circuit.

**2.** The following time is the maximum amount of time necessary to start conversion.

| ADM0 | | | Conversion Clock ($f_{AD}$) | Conversion Start Time (Number of $f_{CLK}$ Clocks) | |
|---|---|---|---|---|---|
| FR2 | FR1 | FR0 | | Software trigger mode/ Hardware trigger no wait mode | Hardware trigger wait mode |
| 0 | 0 | 0 | $f_{CLK}$/64 | 63 | 1 |
| 0 | 0 | 1 | $f_{CLK}$/32 | 31 | |
| 0 | 1 | 0 | $f_{CLK}$/16 | 15 | |
| 0 | 1 | 1 | $f_{CLK}$/8 | 7 | |
| 1 | 0 | 0 | $f_{CLK}$/6 | 5 | |
| 1 | 0 | 1 | $f_{CLK}$/5 | 4 | |
| 1 | 1 | 0 | $f_{CLK}$/4 | 3 | |
| 1 | 1 | 1 | $f_{CLK}$/2 | 1 | |

However, for the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected.

**Remark** $f_{CLK}$: CPU/peripheral hardware clock frequency

**Cautions 1. If using the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (but the bit is automatically switched to 1 when the hardware trigger signal is detected). However, it is possible to clear the ADCS bit to 0 to specify the A/D conversion standby status.**

**2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS flag is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.**

**3. Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby status).**

**4. To complete A/D conversion, specify at least the following time as the hardware trigger interval:**
**Hardware trigger no wait mode: 2 $f_{CLK}$ clocks + A/D conversion time**
**Hardware trigger wait mode: 2 $f_{CLK}$ clocks + stabilization wait time + A/D conversion time**

**Table 12-3. A/D Conversion Time Selection (1/4)**

**(1) 4.0 V ≤ $V_{DD}$ ≤ 5.5 V**

**When there is stabilization wait time (hardware trigger wait mode)**

| A/D Converter Mode Register 0 (ADM0) | | | | | Mode | Conversion Clock ($f_{AD}$) | Number of A/D Power Supply Stabilization Wait Clocks | Number of conversion clocks | A/D Power Supply Stabilization Wait Time + Conversion Time | Conversion Time Selection | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FR2 | FR1 | FR0 | LV1 | LV0 | | | | | | $f_{CLK}$ = 1 MHz | $f_{CLK}$ = 2 MHz | $f_{CLK}$ = 4 MHz | $f_{CLK}$ = 8 MHz | $f_{CLK}$ = 16 MHz | $f_{CLK}$ = 32 MHz |
| 0 | 0 | 0 | 0 | 0 | Normal 1 | $f_{CLK}$/64 | 8 $f_{AD}$ | 19 $f_{AD}$ (number of sampling clocks: 7 $f_{AD}$) | 1728/$f_{CLK}$ | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited |
| 0 | 0 | 1 | | | | $f_{CLK}$/32 | | | 864/$f_{CLK}$ | | | | | | 27 μs |
| 0 | 1 | 0 | | | | $f_{CLK}$/16 | | | 432/$f_{CLK}$ | | | | | 27 μs | 13.5 μs |
| 0 | 1 | 1 | | | | $f_{CLK}$/8 | | | 216/$f_{CLK}$ | | | | 27 μs | 13.5 μs | 6.75 μs |
| 1 | 0 | 0 | | | | $f_{CLK}$/6 | | | 162/$f_{CLK}$ | | | | 20.25 μs | 10.125 μs | 5.0625 μs |
| 1 | 0 | 1 | | | | $f_{CLK}$/5 | | | 135/$f_{CLK}$ | | | 33.75 μs | 16.875 μs | 8.4375 μs | 4.21875 μs |
| 1 | 1 | 0 | | | | $f_{CLK}$/4 | | | 108/$f_{CLK}$ | | | 27 μs | 13.5 μs | 6.75 μs | 3.375 μs |
| 1 | 1 | 1 | | | | $f_{CLK}$/2 | | | 54/$f_{CLK}$ | | 27 μs | 13.5 μs | 6.75 μs | 3.375 μs | Setting prohibited |
| 0 | 0 | 0 | 0 | 1 | Normal 2 | $f_{CLK}$/64 | 8 $f_{AD}$ | 17 $f_{AD}$ (number of sampling clocks: 5 $f_{AD}$) | 1600/$f_{CLK}$ | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited |
| 0 | 0 | 1 | | | | $f_{CLK}$/32 | | | 800/$f_{CLK}$ | | | | | | 25 μs |
| 0 | 1 | 0 | | | | $f_{CLK}$/16 | | | 400/$f_{CLK}$ | | | | | 25 μs | 12.5 μs |
| 0 | 1 | 1 | | | | $f_{CLK}$/8 | | | 200/$f_{CLK}$ | | | | 25 μs | 12.5 μs | 6.25 μs |
| 1 | 0 | 0 | | | | $f_{CLK}$/6 | | | 150/$f_{CLK}$ | | | 37.5 μs | 18.75 μs | 9.375 μs | 4.6875 μs |
| 1 | 0 | 1 | | | | $f_{CLK}$/5 | | | 125/$f_{CLK}$ | | | 31.25 μs | 15.625 μs | 7.8125 μs | 3.90625 μs |
| 1 | 1 | 0 | | | | $f_{CLK}$/4 | | | 100/$f_{CLK}$ | | | 25 μs | 12.5 μs | 6.25 μs | 3.125 μs |
| 1 | 1 | 1 | | | | $f_{CLK}$/2 | | | 50/$f_{CLK}$ | | 25 μs | 12.5 μs | 6.25 μs | 3.125 μs | Setting prohibited |
| Other than above | | | | | | Setting prohibited | | | | | | | | | |

**Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.**

**2. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.**

**3. In the hardware trigger wait mode, the conversion time includes the time spent waiting for stabilization after the hardware trigger is detected.**

**4. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock ($f_{AD}$).**

**Remark** $f_{CLK}$: CPU/peripheral hardware clock frequency

**Table 12-3. A/D Conversion Time Selection (2/4)**

**(2) 2.7 V ≤ $V_{DD}$ < 4.0 V**
**When there is stabilization wait time (hardware trigger wait mode)**

| A/D Converter Mode Register 0 (ADM0) | | | | | Mode | Conversion Clock ($f_{AD}$) | Number of A/D Power Supply Stabilization Wait Clocks | Number of conversion clocks | A/D Power Supply Stabilization Wait Time + Conversion Time | Conversion Time Selection | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FR2 | FR1 | FR0 | LV1 | LV0 | | | | | | $f_{CLK}$ = 1 MHz | $f_{CLK}$ = 2 MHz | $f_{CLK}$ = 4 MHz | $f_{CLK}$ = 8 MHz | $f_{CLK}$ = 16 MHz | $f_{CLK}$ = 32 MHz |
| 0 | 0 | 0 | 0 | 0 | Normal 1 | $f_{CLK}$/64 | 8 $f_{AD}$ | 19 $f_{AD}$ (number of sampling clocks: 7 $f_{AD}$) | 1728/$f_{CLK}$ | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited |
| 0 | 0 | 1 | | | | $f_{CLK}$/32 | | | 864/$f_{CLK}$ | | | | | | 27 μs |
| 0 | 1 | 0 | | | | $f_{CLK}$/16 | | | 432/$f_{CLK}$ | | | | | 27 μs | 13.5 μs |
| 0 | 1 | 1 | | | | $f_{CLK}$/8 | | | 216/$f_{CLK}$ | | | | 27 μs | 13.5 μs | 6.75 μs |
| 1 | 0 | 0 | | | | $f_{CLK}$/6 | | | 162/$f_{CLK}$ | | | | 20.25 μs | 10.125 μs | 5.0625 μs |
| 1 | 0 | 1 | | | | $f_{CLK}$/5 | | | 135/$f_{CLK}$ | | | 33.75 μs | 16.875 μs | 8.4375 μs | 4.2188 μs |
| 1 | 1 | 0 | | | | $f_{CLK}$/4 | | | 108/$f_{CLK}$ | | | 27 μs | 13.5 μs | 6.75 μs | Setting prohibited |
| 1 | 1 | 1 | | | | $f_{CLK}$/2 | | | 54/$f_{CLK}$ | | 27 μs | 13.5 μs | 6.75 μs | Setting prohibited | |
| 0 | 0 | 0 | 0 | 1 | Normal 2 | $f_{CLK}$/64 | 8 $f_{AD}$ | 17 $f_{AD}$ (number of sampling clocks: 5 $f_{AD}$) | 1600/$f_{CLK}$ | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited |
| 0 | 0 | 1 | | | | $f_{CLK}$/32 | | | 800/$f_{CLK}$ | | | | | | 25 μs |
| 0 | 1 | 0 | | | | $f_{CLK}$/16 | | | 400/$f_{CLK}$ | | | | | 25 μs | 12.5 μs |
| 0 | 1 | 1 | | | | $f_{CLK}$/8 | | | 200/$f_{CLK}$ | | | | 25 μs | 12.5 μs | 6.25 μs |
| 1 | 0 | 0 | | | | $f_{CLK}$/6 | | | 150/$f_{CLK}$ | | | 37.5 μs | 18.75 μs | 9.375 μs | 4.6875 μs |
| 1 | 0 | 1 | | | | $f_{CLK}$/5 | | | 125/$f_{CLK}$ | | | 31.25 μs | 15.625 μs | 7.8125 μs | 3.9063 μs |
| 1 | 1 | 0 | | | | $f_{CLK}$/4 | | | 100/$f_{CLK}$ | | | 25 μs | 12.5 μs | 6.25 μs | Setting prohibited |
| 1 | 1 | 1 | | | | $f_{CLK}$/2 | | | 50/$f_{CLK}$ | | 25 μs | 12.5 μs | 6.25 μs | Setting prohibited | |
| Other than above | | | | | | Setting prohibited | | | | | | | | | |

**Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.**

**2. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.**

**3. While in the hardware trigger wait mode, the conversion time includes the time spent waiting for stabilization after the hardware trigger is detected.**

**4. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock ($f_{AD}$).**

**Remark** $f_{CLK}$: CPU/peripheral hardware clock frequency

**Table 12-3. A/D Conversion Time Selection (3/4)**

**(3) 4.0 V ≤ $V_{DD}$ ≤ 5.5 V**

**When there is no stabilization wait time (software trigger mode/hardware trigger no-wait mode)**

| A/D Converter Mode Register 0 (ADM0) | | | | | Mode | Conversion Clock ($f_{AD}$) | Number of conversion clocks | Conversion Time | Conversion Time Selection | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FR2 | FR1 | FR0 | LV1 | LV0 | | | | | $f_{CLK}$ = 1 MHz | $f_{CLK}$ = 2 MHz | $f_{CLK}$ = 4 MHz | $f_{CLK}$ = 8 MHz | $f_{CLK}$ = 16 MHz | $f_{CLK}$ = 32 MHz |
| 0 | 0 | 0 | 0 | 0 | Normal 1 | $f_{CLK}$/64 | 19 $f_{AD}$ (number of sampling clocks: 7 $f_{AD}$) | 1216/$f_{CLK}$ | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | 38 μs |
| 0 | 0 | 1 | | | | $f_{CLK}$/32 | | 608/$f_{CLK}$ | | | | | 38 μs | 19 μs |
| 0 | 1 | 0 | | | | $f_{CLK}$/16 | | 304/$f_{CLK}$ | | | | 38 μs | 19 μs | 9.5 μs |
| 0 | 1 | 1 | | | | $f_{CLK}$/8 | | 152/$f_{CLK}$ | | | 38 μs | 19 μs | 9.5 μs | 4.75 μs |
| 1 | 0 | 0 | | | | $f_{CLK}$/6 | | 114/$f_{CLK}$ | | | 28.5 μs | 14.25 μs | 7.125 μs | 3.5625 μs |
| 1 | 0 | 1 | | | | $f_{CLK}$/5 | | 95/$f_{CLK}$ | | | 23.75 μs | 11.875 μs | 5.938 μs | 2.9688 μs |
| 1 | 1 | 0 | | | | $f_{CLK}$/4 | | 76/$f_{CLK}$ | | 38 μs | 19 μs | 9.5 μs | 4.75 μs | 2.375 μs |
| 1 | 1 | 1 | | | | $f_{CLK}$/2 | | 38/$f_{CLK}$ | 38 μs | 19 μs | 9.5 μs | 4.75 μs | 2.375 μs | Setting prohibited |
| 0 | 0 | 0 | 0 | 1 | Normal 2 | $f_{CLK}$/64 | 17 $f_{AD}$ (number of sampling clocks: 5 $f_{AD}$) | 1088/$f_{CLK}$ | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | 34 μs |
| 0 | 0 | 1 | | | | $f_{CLK}$/32 | | 544/$f_{CLK}$ | | | | | 34 μs | 17 μs |
| 0 | 1 | 0 | | | | $f_{CLK}$/16 | | 272/$f_{CLK}$ | | | | 34 μs | 17 μs | 8.5 μs |
| 0 | 1 | 1 | | | | $f_{CLK}$/8 | | 136/$f_{CLK}$ | | | 34 μs | 17 μs | 8.5 μs | 4.25 μs |
| 1 | 0 | 0 | | | | $f_{CLK}$/6 | | 102/$f_{CLK}$ | | | 25.5 μs | 12.75 μs | 6.375 μs | 3.1875 μs |
| 1 | 0 | 1 | | | | $f_{CLK}$/5 | | 85/$f_{CLK}$ | | | 21.25 μs | 10.625 μs | 5.3125 μs | 2.6563 μs |
| 1 | 1 | 0 | | | | $f_{CLK}$/4 | | 68/$f_{CLK}$ | | 34 μs | 17 μs | 8.5 μs | 4.25 μs | 2.125 μs Note |
| 1 | 1 | 1 | | | | $f_{CLK}$/2 | | 34/$f_{CLK}$ | 34 μs | 17 μs | 8.5 μs | 4.25 μs | 2.125 μs Note | Setting prohibited |
| Other than above | | | | | | | Setting prohibited | | | | | | | |

**Note** This value is prohibited when using the temperature sensor.

**Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.**

**2. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.**

**3. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock ($f_{AD}$).**

**Remark** $f_{CLK}$: CPU/peripheral hardware clock frequency

**Table 12-3. A/D Conversion Time Selection (4/4)**

**(4) 2.7 V ≤ $V_{DD}$ < 4.0 V**

**When there is no stabilization wait time (software trigger mode/hardware trigger no-wait mode)**

| A/D Converter Mode Register 0 (ADM0) | | | | | Mode | Conversion Clock ($f_{AD}$) | Number of conversion clocks | Conversion Time | Conversion Time Selection | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FR2 | FR1 | FR0 | LV1 | LV0 | | | | | $f_{CLK}$ = 1 MHz | $f_{CLK}$ = 2 MHz | $f_{CLK}$ = 4 MHz | $f_{CLK}$ = 8 MHz | $f_{CLK}$ = 16 MHz | $f_{CLK}$ = 32 MHz |
| 0 | 0 | 0 | 0 | 0 | Normal 1 | $f_{CLK}$/64 | 19 $f_{AD}$ (number of sampling clocks: 7 $f_{AD}$) | 1216/$f_{CLK}$ | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | 38 μs |
| 0 | 0 | 1 | | | | $f_{CLK}$/32 | | 608/$f_{CLK}$ | | | | | 38 μs | 19 μs |
| 0 | 1 | 0 | | | | $f_{CLK}$/16 | | 304/$f_{CLK}$ | | | | 38 μs | 19 μs | 9.5 μs |
| 0 | 1 | 1 | | | | $f_{CLK}$/8 | | 152/$f_{CLK}$ | | | 38 μs | 19 μs | 9.5 μs | 4.75 μs |
| 1 | 0 | 0 | | | | $f_{CLK}$/6 | | 114/$f_{CLK}$ | | | 28.5 μs | 14.25 μs | 7.125 μs | 3.5625 μs |
| 1 | 0 | 1 | | | | $f_{CLK}$/5 | | 95/$f_{CLK}$ | | | 23.75 μs | 11.875 μs | 5.938 μs | Setting prohibited |
| 1 | 1 | 0 | | | | $f_{CLK}$/4 | | 76/$f_{CLK}$ | | 38 μs | 19 μs | 9.5 μs | 4.75 μs | |
| 1 | 1 | 1 | | | | $f_{CLK}$/2 | | 38/$f_{CLK}$ | 38 μs | 19 μs | 9.5 μs | 4.75 μs | Setting prohibited | |
| 0 | 0 | 0 | 0 | 1 | Normal 2 | $f_{CLK}$/64 | 17 $f_{AD}$ (number of sampling clocks: 5 $f_{AD}$) | 1088/$f_{CLK}$ | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | Setting prohibited | 34 μs |
| 0 | 0 | 1 | | | | $f_{CLK}$/32 | | 544/$f_{CLK}$ | | | | | 34 μs | 17 μs |
| 0 | 1 | 0 | | | | $f_{CLK}$/16 | | 272/$f_{CLK}$ | | | | 34 μs | 17 μs | 8.5 μs |
| 0 | 1 | 1 | | | | $f_{CLK}$/8 | | 136/$f_{CLK}$ | | | 34 μs | 17 μs | 8.5 μs | 4.25 μs |
| 1 | 0 | 0 | | | | $f_{CLK}$/6 | | 102/$f_{CLK}$ | | | 25.5 μs | 12.75 μs | 6.375 μs | 3.1875 μs Note |
| 1 | 0 | 1 | | | | $f_{CLK}$/5 | | 85/$f_{CLK}$ | | | 21.25 μs | 10.625 μs | 5.3125 μs | Setting prohibited |
| 1 | 1 | 0 | | | | $f_{CLK}$/4 | | 68/$f_{CLK}$ | | 34 μs | 17 μs | 8.5 μs | 4.25 μs | |
| 1 | 1 | 1 | | | | $f_{CLK}$/2 | | 34/$f_{CLK}$ | 34 μs | 17 μs | 8.5 μs | 4.25 μs | Setting prohibited | |
| Other than above | | | | | | Setting prohibited | | | | | | | | |

**Note** This value is prohibited when using the temperature sensor.

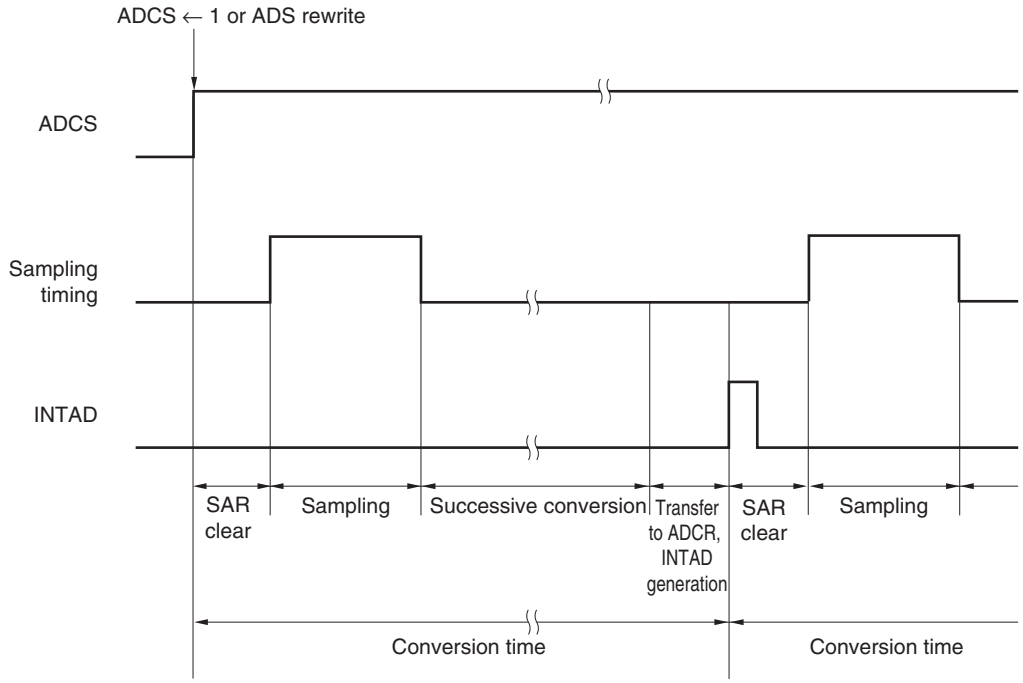
**Cautions 1. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.**

**2. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.**

**3. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock ($f_{AD}$).**

**Remark** $f_{CLK}$: CPU/peripheral hardware clock frequency

**Figure 12-5. A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)**



### 12.3.3 A/D converter mode register 1 (ADM1)

This register is used to specify the A/D conversion trigger, conversion mode, and hardware trigger signal.

The ADM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 12-6. Format of A/D Converter Mode Register 1 (ADM1)**

Address: FFF32H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADM1 | ADTMD1 | ADTMD0 | ADSCM | 0 | 0 | 0 | ADTRS1 | ADTRS0 |

| ADTMD1 | ADTMD0 | Selection of the A/D conversion trigger mode |
|---|---|---|
| 0 | × | Software trigger mode |
| 1 | 0 | Hardware trigger no-wait mode |
| 1 | 1 | Hardware trigger wait mode |

| ADSCM | Specification of the A/D conversion mode |
|---|---|
| 0 | Sequential conversion mode |
| 1 | One-shot conversion mode |

| ADTRS1 | ADTRS0 | Selection of the hardware trigger signal |
|---|---|---|
| 0 | 0 | End of TAU0 channel 1 count or capture interrupt signal (INTTM01) |
| 0 | 1 | Trigger signal generated by the ADTRGS0 and ADTRGS1 registers Note 1<br>Event signal selected by ELC Note 2 |
| 1 | 0 | Pretimed signal/alarm interrupt signal (INTRTC) |
| 1 | 1 | Setting prohibited |

**Notes 1.** Only in the RL78/F13.

**2.** Only in the RL78/F14.

**Cautions 1. Only rewrite the value of the ADM1 register while conversion operation is stopped (which is indicated by the ADCS bit of A/D converter mode register 0 (ADM0) being 0).**

**2. To complete A/D conversion, specify at least the following time as the hardware trigger interval:**
**Hardware trigger no wait mode: 2 $f_{CLK}$ clocks + A/D conversion time**
**Hardware trigger wait mode: 2 $f_{CLK}$ clocks + stabilization wait time + A/D conversion time**

**3. In modes other than SNOOZE mode, input of the next INTRTC will not be recognized as a valid hardware trigger for up to four $f_{CLK}$ cycles after the first INTRTC is input.**

**Remarks 1.** ×: don't care

**2.** $f_{CLK}$: CPU/peripheral hardware clock frequency

### 12.3.4 A/D converter mode register 2 (ADM2)

This register is used to select the A/D converter reference voltage, check the upper limit and lower limit A/D conversion result values, select the resolution, and specify whether to use the SNOOZE mode.

The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 12-7. Format of A/D Converter Mode Register 2 (ADM2) (1/2)**

Address: F0010H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | <3> | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| ADM2 | ADREFP1 | ADREFP0 | ADREFM | 0 | ADRCK | AWC | 0 | ADTYP |

| ADREFP1 | ADREFP0 | Selection of the + side reference voltage source of the A/D converter |
|---|---|---|
| 0 | 0 | Supplied from $V_{DD}$ |
| 0 | 1 | Supplied from P33/$AV_{REFP}$/ANI0 |
| 1 | 0 | Supplied from the internal reference voltage (1.45 V) |
| 1 | 1 | Setting prohibited |

- When ADREFP1 or ADREFP0 bit is rewritten, this must be configured in accordance with the following procedures.
  (1) Set ADCE = 0
  (2) Change the values of ADREFP1 and ADREFP0
  (3) Stabilization wait time (A)
  (4) Set ADCE = 1
  (5) Stabilization wait time (B)
  When ADREFP1 and ADREFP0 are set to 1 and 0, the setting is changed to A = 5 μs, B = 1 μs.
  When ADREFP1 and ADREFP0 are set to 0 and 0 or 0 and 1, A needs no wait and B = 1 μs.
  Start A/D conversion after the wait time (5) specified above).
- When ADREFP1 and ADREFP0 are set to 1 and 0, respectively, A/D conversion cannot be performed on the temperature sensor output and internal reference voltage output.
  Be sure to perform A/D conversion while ADISS = 0.

| ADREFM | Selection of the – side reference voltage source of the A/D converter |
|---|---|
| 0 | Supplied from $V_{SS}$ |
| 1 | Supplied from P34/$AV_{REFM}$/ANI1 |

| ADRCK | Checking the upper limit and lower limit conversion result values |
|---|---|
| 0 | The interrupt signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (<1>). |
| 1 | The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (<2>) or the ADUL register < the ADCR register (<3>). |
| Figure 12-8 shows the generation range of the interrupt signal (INTAD) for <1> to <3>. | |

**Cautions 1. Only rewrite the value of the ADM2 register while conversion operation is stopped (which is indicated by the ADCS bit of A/D converter mode register 0 (ADM0) being 0).**

**2. When entering STOP mode or HALT mode while the CPU is operating on the subsystem/low-speed on-chip oscillator select clock, do not set ADREFP1 to 1. When the internal reference voltage is selected (ADREFP1, ADREFP0 = 1, 0), the current value defined in the supply current characteristics in CHAPTER 34 to CHAPTER 36 ELECTRICAL SPECIFICATIONS must be added.**

**3. When using $AV_{REFP}$ and $AV_{REFM}$, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.**

**Figure 12-7. Format of A/D Converter Mode Register 2 (ADM2) (2/2)**

Address: F0010H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | <3> | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| ADM2 | ADREFP1 | ADREFP0 | ADREFM | 0 | ADRCK | AWC | 0 | ADTYP |

| AWC | Specification of the SNOOZE mode |
|---|---|
| 0 | Do not use the SNOOZE mode function. |
| 1 | Use the SNOOZE mode function. |

When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode).

- The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock ($f_{CLK}$). If any other clock is selected, specifying this mode is prohibited.
- Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited.
- Using the SNOOZE mode function in the sequential conversion mode is prohibited.
- When using the SNOOZE mode function, specify a hardware trigger interval of at least “shift time to SNOOZE mode Note + A/D power supply stabilization wait time + A/D conversion time + 2 $f_{CLK}$ clocks”.

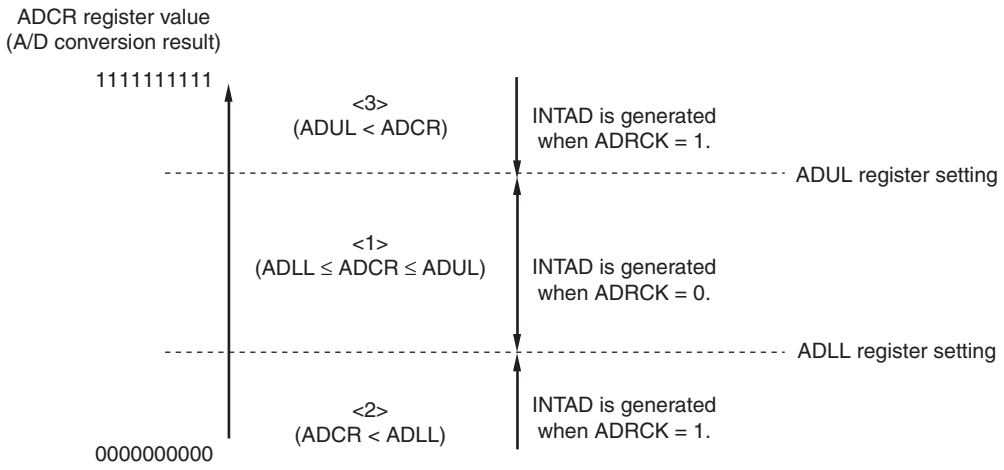
□ Even when using SNOOZE mode, be sure to set the AWC bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode.
Also, be sure to change the AWC bit to 0 after returning from STOP mode to normal operation mode.
If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE or normal operation mode.

| ADTYP | Selection of the A/D conversion resolution |
|---|---|
| 0 | 10-bit resolution |
| 1 | 8-bit resolution |

**Note** Refer to “From STOP to SNOOZE” in **23.3.3 SNOOZE mode**.

**Caution Only rewrite the value of the ADM2 register while conversion operation is stopped (which is indicated by the ADCS bit of A/D converter mode register 0 (ADM0) being 0).**

**Figure 12-8. ADRCK Bit Interrupt Signal Generation Range**




**Remark** If INTAD does not occur, the A/D conversion result is not stored in the ADCR or ADCRH register.

### 12.3.5 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result in the select mode. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH **Note**.

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

**Note** If the A/D conversion result is outside the range specified by using the A/D conversion result comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 12-8**), the result is not stored.

**Figure 12-9. Format of 10-bit A/D Conversion Result Register (ADCR)**

Address: FFF1FH, FFF1EH After reset: 0000H R

| Symbol | FFF1FH | | | | | | | | FFF1EH | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADCR | | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Cautions 1. When writing to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of the ADCR register may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, and ADPC registers. Using timing other than the above may cause an incorrect conversion result to be read.**

**2. When 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1) and the ADCR register is read, 0 is read from the lower two bits (ADCR1 and ADCR0).**

**3. When the ADCR register is accessed in 16-bit units, the higher 10 bits of the conversion result are read in order starting at bit 15.**

### 12.3.6 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored Note.

The ADCRH register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Note** If the A/D conversion result is outside the range specified by using the A/D conversion result comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 12-8**), the result is not stored.

**Figure 12-10. Format of 8-bit A/D Conversion Result Register (ADCRH)**

Address: FFF1FH After reset: 00H R

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADCRH | | | | | | | | |

**Caution When writing to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of the ADCRH register may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, and ADPC registers. Using timing other than the above may cause an incorrect conversion result to be read.**

### 12.3.7 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 12-11. Format of Analog Input Channel Specification Register (ADS) (1/2)**

Address: FFF31H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADS | ADISS | 0 | 0 | ADS4 | ADS3 | ADS2 | ADS1 | ADS0 |

○ Select mode (ADMD = 0)

| ADISS | ADS4 | ADS3 | ADS2 | ADS1 | ADS0 | Analog input channel | Input source |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | ANI0 | P33/$AV_{REFP}$/ANI0 |
| 0 | 0 | 0 | 0 | 0 | 1 | ANI1 | P34/$AV_{REFM}$/ANI1 |
| 0 | 0 | 0 | 0 | 1 | 0 | ANI2 | P80/ANI2/ANO0 |
| 0 | 0 | 0 | 0 | 1 | 1 | ANI3 | P81/ANI3/IVCMP00 |
| 0 | 0 | 0 | 1 | 0 | 0 | ANI4 | P82/ANI4/IVCMP01 |
| 0 | 0 | 0 | 1 | 0 | 1 | ANI5 | P83/ANI5/IVCMP02 |
| 0 | 0 | 0 | 1 | 1 | 0 | ANI6 | P84/ANI6/IVCMP03 |
| 0 | 0 | 0 | 1 | 1 | 1 | ANI7 | P85/ANI7/IVREF0 |
| 0 | 0 | 1 | 0 | 0 | 0 | ANI8 | P86/ANI8 |
| 0 | 0 | 1 | 0 | 0 | 1 | ANI9 | P87/ANI9 |
| 0 | 0 | 1 | 0 | 1 | 0 | ANI10 | P90/ANI10 |
| 0 | 0 | 1 | 0 | 1 | 1 | ANI11 | P91/ANI11 |
| 0 | 0 | 1 | 1 | 0 | 0 | ANI12 | P92/ANI12 |
| 0 | 0 | 1 | 1 | 0 | 1 | ANI13 | P93/ANI13 |
| 0 | 0 | 1 | 1 | 1 | 0 | ANI14 | P94/ANI14 |
| 0 | 0 | 1 | 1 | 1 | 1 | ANI15 | P95/ANI15 |
| 0 | 1 | 0 | 0 | 0 | 0 | ANI16 | P96/ANI16 Note 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | ANI17 | P97/ANI17 Note 2 |
| 0 | 1 | 0 | 0 | 1 | 0 | ANI18 | P100/ANI18 |
| 0 | 1 | 0 | 0 | 1 | 1 | ANI19 | P101/ANI19 |
| 0 | 1 | 0 | 1 | 0 | 0 | ANI20 | P102/ANI20 |
| 0 | 1 | 0 | 1 | 0 | 1 | ANI21 | P103/ANI21 |
| 0 | 1 | 0 | 1 | 1 | 0 | ANI22 | P104/ANI22 |
| 0 | 1 | 0 | 1 | 1 | 1 | ANI23 | P105/ANI23 |
| 0 | 1 | 1 | 0 | 0 | 0 | ANI24 | P125/ANI24 |
| 0 | 1 | 1 | 0 | 0 | 1 | ANI25 | P120/ANI25 |
| 0 | 1 | 1 | 0 | 1 | 0 | ANI26 | P96/ANI26 Note 3, P70/ANI26 Note 4 |
| 0 | 1 | 1 | 0 | 1 | 1 | ANI27 | P97/ANI27 Note 5, P71/ANI27 Note 6 |
| 0 | 1 | 1 | 1 | 0 | 0 | ANI28 | P72/ANI28 |
| 0 | 1 | 1 | 1 | 0 | 1 | ANI29 | P73/ANI29 |
| 0 | 1 | 1 | 1 | 1 | 0 | ANI30 | P74/ANI30 |
| 1 | 0 | 0 | 0 | 0 | 0 | — | Temperature sensor output |
| 1 | 0 | 0 | 0 | 0 | 1 | — | Internal reference voltage output (1.45 V) |
| Other than above | | | | | | Setting prohibited | |

(**Notes** are listed on the next page.)

**Notes 1.** RL78/F14 products with 64 or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory, or RL78/F14 products with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory.

**2.** RL78/F14 products with 80 pins and 128 Kbytes to 256 Kbytes of code flash memory, or RL78/F14 products with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory.

**3.** RL78/F14 products with 64 or 80 pins and 64 Kbytes to 96 Kbytes of code flash memory, RL78/F13 (CAN and LIN incorporated) products with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory,
or RL78/F13 (CAN and LIN incorporated) products with 64 pins and 32 Kbytes to 128 Kbytes of code flash memory,
or RL78/F13 (LIN incorporated) products with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory,
or RL78/F13 (LIN incorporated) products with 64 pins and 96 Kbytes to 128 Kbytes of code flash memory.

**4.** RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory, or RL78/F14 products with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory.

**5.** RL78/F14 products with 80 pins and 64 Kbytes to 96 Kbytes of code flash memory, RL78/F13 (CAN and LIN incorporated) or RL78/F13 (LIN incorporated) products with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory.

**6.** RL78/F14 products with 48 or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory, or RL78/F14 products with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

**Figure 12-11. Format of Analog Input Channel Specification Register (ADS) (2/2)**

Address: FFF31H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADS | ADISS | 0 | 0 | ADS4 | ADS3 | ADS2 | ADS1 | ADS0 |

○ Scan mode (ADMD = 1)

| ADISS | ADS4 | ADS3 | ADS2 | ADS1 | ADS0 | Analog input channel | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Scan 0 | Scan 1 | Scan 2 | Scan 3 |
| 0 | 0 | 0 | 0 | 0 | 0 | ANI0 | ANI1 | ANI2 | ANI3 |
| 0 | 0 | 0 | 0 | 0 | 1 | ANI1 | ANI2 | ANI3 | ANI4 |
| 0 | 0 | 0 | 0 | 1 | 0 | ANI2 | ANI3 | ANI4 | ANI5 |
| 0 | 0 | 0 | 0 | 1 | 1 | ANI3 | ANI4 | ANI5 | ANI6 |
| 0 | 0 | 0 | 1 | 0 | 0 | ANI4 | ANI5 | ANI6 | ANI7 |
| 0 | 0 | 0 | 1 | 0 | 1 | ANI5 | ANI6 | ANI7 | ANI8 |
| 0 | 0 | 0 | 1 | 1 | 0 | ANI6 | ANI7 | ANI8 | ANI9 |
| 0 | 0 | 0 | 1 | 1 | 1 | ANI7 | ANI8 | ANI9 | ANI10 |
| 0 | 0 | 1 | 0 | 0 | 0 | ANI8 | ANI9 | ANI10 | ANI11 |
| 0 | 0 | 1 | 0 | 0 | 1 | ANI9 | ANI10 | ANI11 | ANI12 |
| 0 | 0 | 1 | 0 | 1 | 0 | ANI10 | ANI11 | ANI12 | ANI13 |
| 0 | 0 | 1 | 0 | 1 | 1 | ANI11 | ANI12 | ANI13 | ANI14 |
| 0 | 0 | 1 | 1 | 0 | 0 | ANI12 | ANI13 | ANI14 | ANI15 |
| 0 | 1 | 0 | 0 | 0 | 0 | ANI16 | ANI17 | ANI18 | ANI19 |
| 0 | 1 | 0 | 0 | 0 | 1 | ANI17 | ANI18 | ANI19 | ANI20 |
| 0 | 1 | 0 | 0 | 1 | 0 | ANI18 | ANI19 | ANI20 | ANI21 |
| 0 | 1 | 0 | 0 | 1 | 1 | ANI19 | ANI20 | ANI21 | ANI22 |
| 0 | 1 | 0 | 1 | 0 | 0 | ANI20 | ANI21 | ANI22 | ANI23 |
| Other than the above | | | | | | Setting prohibited | | | |

**Cautions 1. Be sure to clear bits 5 and 6 to 0.**

**2. Set the port that is set to analog input by the ADPC and PMCxx registers to the input mode by using port mode registers 3, 7 to 10, or 12 (PM3, PM7 to PM10, PM12).**

**3. Do not set the pin that is set by the A/D port configuration register (ADPC) as digital I/O by the ADS register.**

**4. Do not set the pin that is set by port mode control registers 7, 9, or 12 (PMC7, PMC9, PMC12) as digital I/O by the ADS register.**

**5. Only rewrite the value of the ADISS bit while A/D voltage comparator operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).**

**6. If using $AV_{REFP}$ as the + side reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.**

**7. If using $AV_{REFM}$ as the – side reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.**

**8. If ADISS is set to 1, the internal reference voltage (1.45 V) cannot be used for the + side reference voltage source. In addition, the result of the first conversion after ADISS has been set to 1 is not usable.**

**9. When entering STOP mode or HALT mode while the CPU is operating on the subsystem/low-speed on-chip oscillator select clock, do not set ADISS to 1. When ADISS is set to 1, the current value defined in the supply current characteristics in CHAPTER 34 to CHAPTER 36 ELECTRICAL SPECIFICATIONS must be added.**

**10. Ignore the conversion result if the corresponding ANI pin does not exist in the product used.**

### 12.3.8 Conversion result comparison upper limit setting register (ADUL)

This register is used to specify the setting for checking the upper limit of the A/D conversion results.

The A/D conversion results and ADUL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 12-8**).

The ADUL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

**Figure 12-12. Format of Conversion Result Comparison Upper Limit Setting Register (ADUL)**

Address: F0011H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADUL | ADUL7 | ADUL6 | ADUL5 | ADUL4 | ADUL3 | ADUL2 | ADUL1 | ADUL0 |

**Cautions 1. When A/D conversion with 10-bit resolution is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the value in the ADUL register.**

**2. Writing new values to the ADUL and ADLL registers is prohibited while conversion is enabled. Write new values to these registers while conversion is stopped (ADCE = 0).**

**3. The setting of the ADUL registers must be greater than that of the ADLL register.**

### 12.3.9 Conversion result comparison lower limit setting register (ADLL)

This register is used to specify the setting for checking the lower limit of the A/D conversion results.

The A/D conversion results and ADLL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 12-8**).

The ADLL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 12-13. Format of Conversion Result Comparison Lower Limit Setting Register (ADLL)**

Address: F0012H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADLL | ADLL7 | ADLL6 | ADLL5 | ADLL4 | ADLL3 | ADLL2 | ADLL1 | ADLL0 |

**Cautions 1. When A/D conversion with 10-bit resolution is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the value in the ADLL register.**

**2. Writing new values to the ADUL and ADLL registers is prohibited while conversion is enabled. Write new values to these registers while conversion is stopped (ADCE = 0).**

**3. The setting of the ADUL registers must be greater than that of the ADLL register.**

### 12.3.10 A/D test register (ADTES)

This register is used to select the + side reference voltage ($AV_{REFP}$) or - side reference voltage ($AV_{REFM}$) of the A/D converter, or the analog input channel (ANIxx) as the A/D conversion target for the A/D test function.

The ADTES register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 12-14. Format of A/D Test Register (ADTES)**

Address: F0013H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADTES | 0 | 0 | 0 | 0 | 0 | 0 | ADTES1 | ADTES0 |

| ADTES1 | ADTES0 | A/D conversion target |
|---|---|---|
| 0 | 0 | ANIxx (This is specified using the analog input channel specification register (ADS).) |
| 1 | 0 | $AV_{REFM}$ |
| 1 | 1 | $AV_{REFP}$ |
| Other than the above | | Setting prohibited |

**Caution For details of the A/D test function, see CHAPTER 27 SAFETY FUNCTIONS.**

### 12.3.11 A/D port configuration register (ADPC)

This register switches the ANI0/P33 to ANI23/P105 pins to analog input of A/D converter or digital I/O of port.

The ADPC register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 12-15. Format of A/D Port Configuration Register (ADPC)**

Address: F0076H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADPC | 0 | 0 | 0 | ADPC4 Note | ADPC3 | ADPC2 | ADPC1 | ADPC0 |

| | | | | | Analog input (A)/digital I/O (D) switching | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADPC4 | ADPC3 | ADPC2 | ADPC1 | ADPC0 | ANI23/P105 | ANI22/P104 | ANI21/P103 | ANI20/P102 | ANI19/P101 | ANI18/P100 | ANI17/P97 | ANI16/P96 | ANI15/P95 | ANI14/P94 | ANI13/P93 | ANI12/P92 | ANI11/P91 | ANI10/P90 | ANI9/P87 | ANI8/P86 | ANI7/IVREF0/P85 | ANI6/IVCMP03/P84 | ANI5/IVCMP02/P83 | ANI4/IVCMP01/P82 | ANI3/IVCMP00/P81 | ANI2/ANO0/P80 | ANI1/$AV_{REFM}$/P34 | ANI0/$AV_{REFP}$/P33 |
| 0 | 0 | 0 | 0 | 0 | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 0 | 0 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D |
| 0 | 0 | 0 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A |
| 0 | 0 | 0 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A |
| 0 | 0 | 1 | 0 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A |
| 0 | 0 | 1 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A |
| 0 | 0 | 1 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A |
| 0 | 0 | 1 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A |
| 0 | 1 | 0 | 0 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A |
| 0 | 1 | 0 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 0 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 0 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 0 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 0 | 0 | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 0 | 1 | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 1 | 0 | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 1 | 1 | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 0 | 0 | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 0 | 1 | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 1 | 0 | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 1 | 1 | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 1 | 0 | 0 | 0 | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| Other than above | | | | | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | | |

(**Note** and **Cautions** are listed on the next page.)

**Note** Not provided in the RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory.

**Cautions 1. Set the port that is set to analog input by the ADPC register to the input mode by using port mode registers 3, 8 to 10, or 12 (PM3, PM8 to PM10, PM12).**

**2. Do not set the pin set by the ADPC register as digital I/O by the analog input channel specification register (ADS).**

**3. When using $AV_{REFP}$ and $AV_{REFM}$, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.**

### 12.3.12 A/D converter trigger select register 0 (ADTRGS0) (RL78/F13 only)

This register is used to enable or disable operation trigger generation of the A/D converter when the timer RD0 input capture B/compare match B interrupt request is generated.

Set this register while the timer RD0 input capture B/compare match B interrupt request is not generated.

Reset signal generation clears this register to 00H.

**Figure 12-16. Format of A/D Converter Trigger Select Register 0 (ADTRGS0)**

Address: F0789H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADTRGS0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ADTRGS00 |

| ADTRGS00 | Selection of the operation trigger of the A/D converter when the timer RD0 input capture B/compare match B interrupt request is generated |
|---|---|
| 0 | The operation trigger of the A/D converter is not generated when the interrupt request is generated. |
| 1 | The operation trigger of the A/D converter is generated when the interrupt request is generated (A/D conversion is started). |

### 12.3.13 A/D converter trigger select register 1 (ADTRGS1) (RL78/F13 only)

This register is used to enable or disable operation trigger generation of the A/D converter when the timer RJ0 interrupt request is generated.

Set this register while the timer RJ0 interrupt request is not generated.

Reset signal generation clears this register to 00H.

**Figure 12-17. Format of A/D Converter Trigger Select Register 1 (ADTRGS1)**

Address: F078DH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADTRGS1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ADTRGS10 |

| ADTRGS10 | Selection of the operation trigger of the A/D converter when the timer RJ0 interrupt request is generated |
|---|---|
| 0 | The operation trigger of the A/D converter is not generated when the interrupt request is generated. |
| 1 | The operation trigger of the A/D converter is generated when the interrupt request is generated (A/D conversion is started). |

### 12.3.14 Port mode control registers 7, 9, and 12 (PMC7, PMC9, PMC12)

These registers are used to switch the ANI24 to ANI30 pins between the analog input of the A/D converter and the digital I/O of the port.

The PMC7, PMC9, and PMC12 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

**Figure 12-18. Formats of Port Mode Control Registers 7, 9, and 12 (PMC7, PMC9, PMC12)**

Address: F0067H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PMC7 | 1 | 1 | 1 | PMC74 Note 1 | PMC73 Note 1 | PMC72 Note 1 | PMC71 Note 1 | PMC70 |

Address: F0069H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PMC9 | PMC97 Note 2 | PMC96 Note 2 | 1 | 1 | 1 | 1 | 1 | 1 |

Address: F006CH After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PMC12 | 1 | 1 | PMC125 | 1 | 1 | 1 | 1 | PMC120 |

| PMC | Digital I/O/analog input selection |
|---|---|
| 0 | Digital I/O (dual-use function other than analog input) |
| 1 | Analog input |

<R> **Notes 1.** Be sure to clear the following bits to 0.
PMC71 to PMC74 bits in the RL78/F14 products with 64 pins and 128 Kbytes to 256 Kbytes of code flash memory.
PMC73 bit in the RL78/F14 products with 48 pins and 128 Kbytes to 256 Kbytes of code flash memory.

**2.** The ADPC and PMC9 registers are used to select the digital I/O or analog input functions for the P96/ANI16 and P97/ANI17 pins and for the P96/ANI26 and P97/ANI27 pins, respectively. For details on pin functions allocated to each product, see **1.5 Pin Configurations**.

**Cautions 1. Set port pins specified as analog input pins to input mode by using port mode register x (PMx).**

<R> **2. Be sure to set bits for pins that are not present to their initial values , and see Note 1 for PMC71 to PMC74.**

### 12.3.15 Port mode registers 3, 7 to 10, and 12 (PM3, PM7 to PM10, PM12)

When using the ANI0/P33 to ANI23/P105 and ANI24/P125 to ANI30/P74 pins for an analog input port, set the PMmn bit to 1. The output latches of PMmn at this time may be 0 or 1.

If the PMmn bits are set to 0, they cannot be used as analog input port pins.

The PMm registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

**Cautions 1. Available pins differ depending on the products. For details, see CHAPTER 2 PIN FUNCTIONS.**

**2. If a pin is set as an analog input port, not the pin level but 0 is always read.**

**Remark** m = 3, 7 to 10, 12; n = 0 to 7

**Figure 12-19. Formats of Port Mode Registers 3, 7 to 10, and 12 (PM3, PM7 to PM10, PM12) (100-pin Products in the RL78/F14)**

Address: FFF23H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM3 | 1 | 1 | 1 | PM34 | PM33 | 1 | 1 | 1 |

Address: FFF27H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM7 | 1 | 1 | 1 | PM74 | PM73 | PM72 | PM71 | PM70 |

Address: FFF28H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM8 | PM87 | PM86 | PM85 | PM84 | PM83 | PM82 | PM81 | PM80 |

Address: FFF29H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM9 | PM97 | PM96 | PM95 | PM94 | PM93 | PM92 | PM91 | PM90 |

Address: FFF2AH After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM10 | 1 | 1 | PM105 | PM104 | PM103 | PM102 | PM101 | PM100 |

Address: FFF2CH After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM12 | 1 | 1 | PM125 | 1 | 1 | 1 | 1 | PM120 |

| PM bit | I/O mode selection |
|---|---|
| 0 | Output mode (output buffer on) |
| 1 | Input mode (output buffer off) |

**Caution When using $AV_{REFP}$ and $AV_{REFM}$, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.**

The ANI0/P33 to ANI23/P105 pins are as shown below depending on the settings of the A/D port configuration register (ADPC), analog input channel specification register (ADS), PM3, PM8, PM9, and PM10 registers.

**Table 12-4. Setting Functions of ANI0/P33 to ANI23/P105 Pins**

| ADPC | PM3, PM8, PM9, PM10 | ADS | ANI0/P33 to ANI23/P105 Pins |
|---|---|---|---|
| Digital I/O selection | Input mode | – | Digital input |
| | Output mode | – | Digital output |
| Analog input selection | Input mode | Selects ANI. | Analog input (to be converted) |
| | | Does not select ANI. | Analog input (not to be converted) |
| | Output mode | Selects ANI. | Setting prohibited |
| | | Does not select ANI. | |

The ANI24 to ANI30 pins are as shown below depending on the settings of port mode control registers 7, 9, and 12 (PMC7, PMC9, PMC12), analog input channel specification register (ADS), PM7, PM9, and PM12 registers.

**Table 12-5. Setting Functions of ANI24 to ANI30 Pins**

| PMC7, PMC9, and PMC12 | PM7, PM9, and PM12 | ADS | ANI24 to ANI30 Pins |
|---|---|---|---|
| Digital I/O selection | Input mode | – | Digital input |
| | Output mode | – | Digital output |
| Analog input selection | Input mode | Selects ANI. | Analog input (to be converted) |
| | | Does not select ANI. | Analog input (not to be converted) |
| | Output mode | Selects ANI. | Setting prohibited |
| | | Does not select ANI. | |

## 12.4 A/D Converter Conversion Operations

The A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<2> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) $AV_{REF}$ by the tap selector.

<4> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than (1/2) $AV_{REF}$, the MSB bit of the SAR register remains set to 1. If the analog input is smaller than (1/2) $AV_{REF}$, the MSB bit is reset to 0.

<5> Next, bit 8 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) $AV_{REF}$
- Bit 9 = 0: (1/4) $AV_{REF}$

The voltage tap and sampled voltage are compared and bit 8 of the SAR register is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison of 10 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched **Note 1**.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated **Note 1**.

<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0 **Note 2**.
To stop the A/D converter, clear the ADCS bit to 0.

**Notes 1.** If the A/D conversion result is outside the range specified by using the A/D conversion result comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 12-8**), the A/D conversion end interrupt request signal (INTAD) is not generated and no A/D conversion results are stored in the ADCR and ADCRH registers.

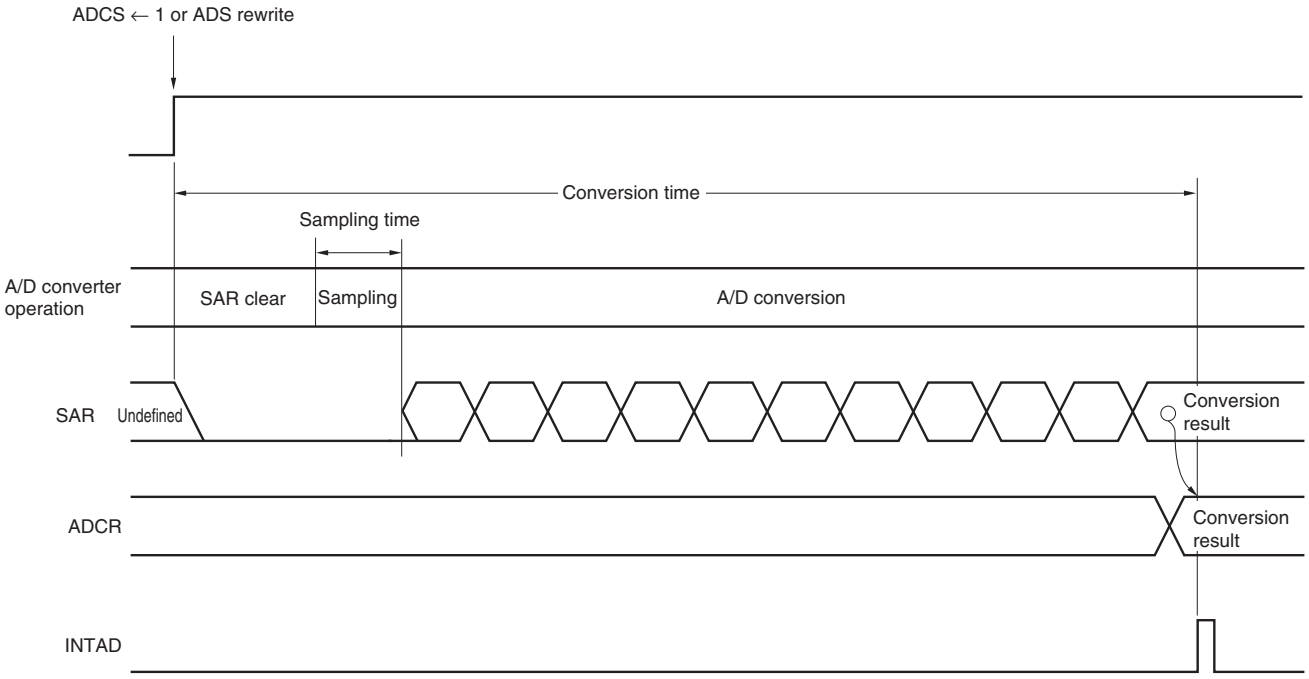
**2.** While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

**Remarks 1.** Two types of the A/D conversion result registers are available.

- ADCR register (16 bits): Store 10-bit A/D conversion value
- ADCRH register (8 bits): Store 8-bit A/D conversion value

**2.** $AV_{REF}$: The + side reference voltage of the A/D converter. This can be selected from $AV_{REFP}$, the internal reference voltage (1.45 V), and $V_{DD}$.

**Figure 12-20. Conversion Operation of A/D Converter (Software Trigger Mode)**




A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register (ADM) is reset (0) by software.

If a write operation is performed to the analog input channel specification register (ADS) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (1), conversion starts again from the beginning.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

## 12.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI23, ANI24 to ANI30) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

$V_{AIN}$: Analog input voltage

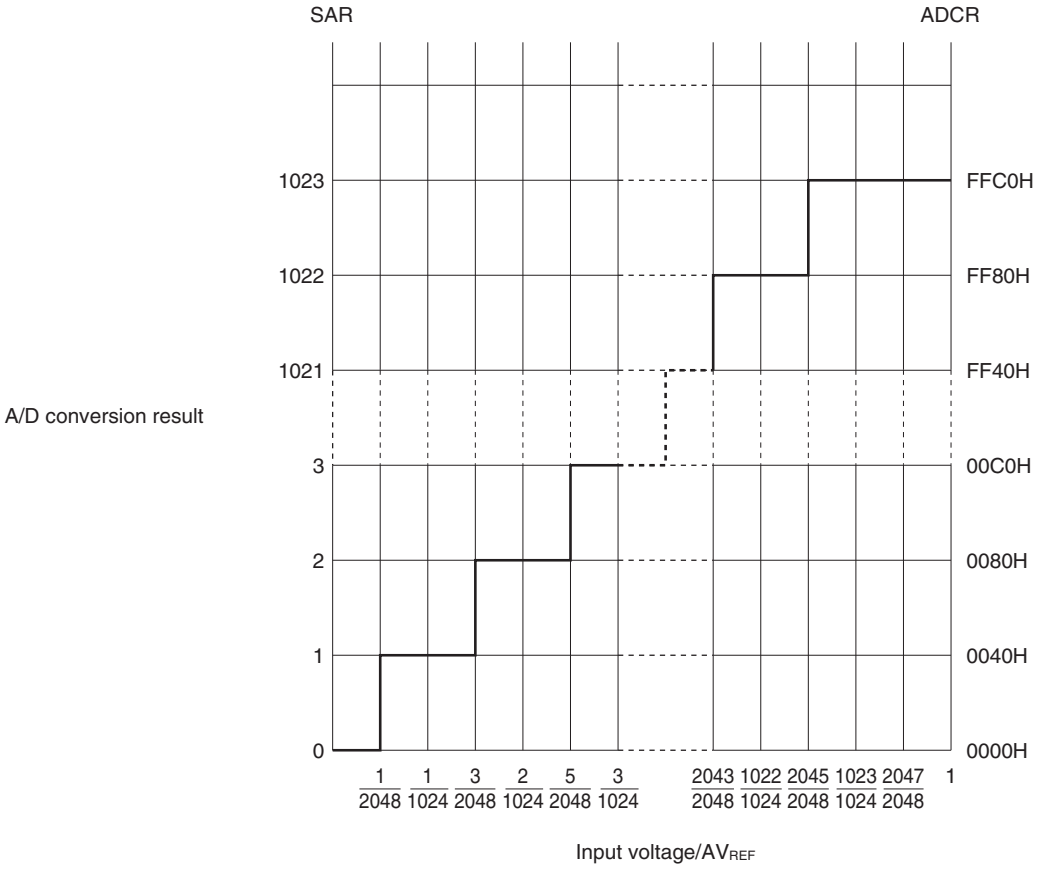
$AV_{REF}$: $AV_{REF}$ pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 12-21 shows the relationship between the analog input voltage and the A/D conversion result.

**Figure 12-21 Relationship Between Analog Input Voltage and A/D Conversion Result**




**Remark** $AV_{REF}$: The + side reference voltage of the A/D converter. This can be selected from $AV_{REFP}$, the internal reference voltage (1.45 V), and $V_{DD}$.

## 12.6 A/D Converter Operation Modes

The operation of each A/D converter mode is described below. In addition, the procedure for specifying each mode is described in **12.7 A/D Converter Setup Flowchart**.

### 12.6.1 Software trigger mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

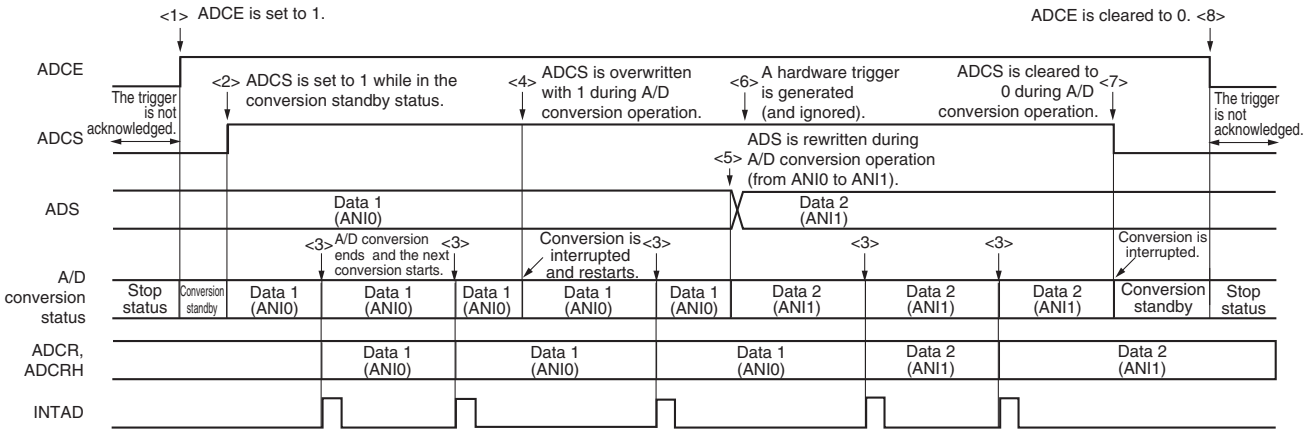
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

**Figure 12-22. Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing**



### 12.6.2 Software trigger mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

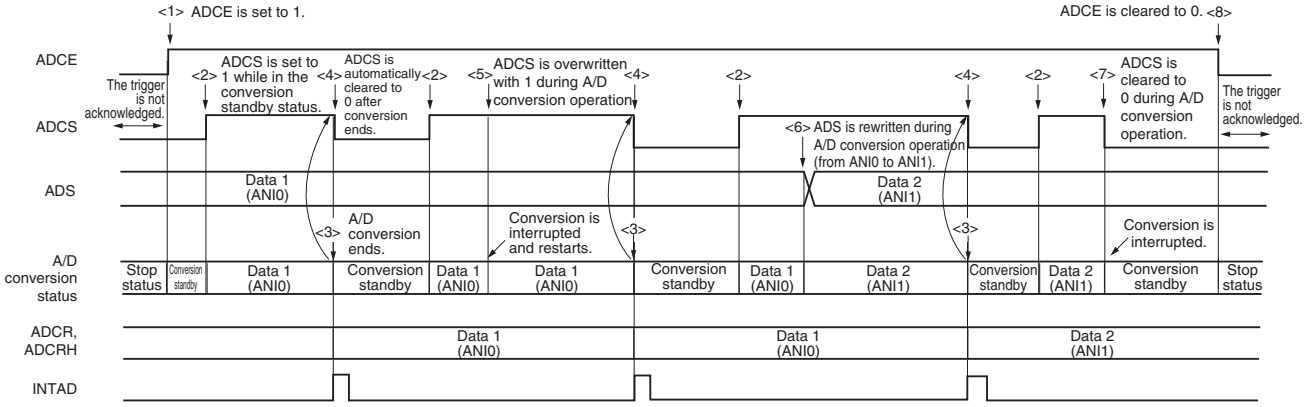
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

**Figure 12-23. Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing**



### 12.6.3 Software trigger mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts (until all four channels are finished).

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

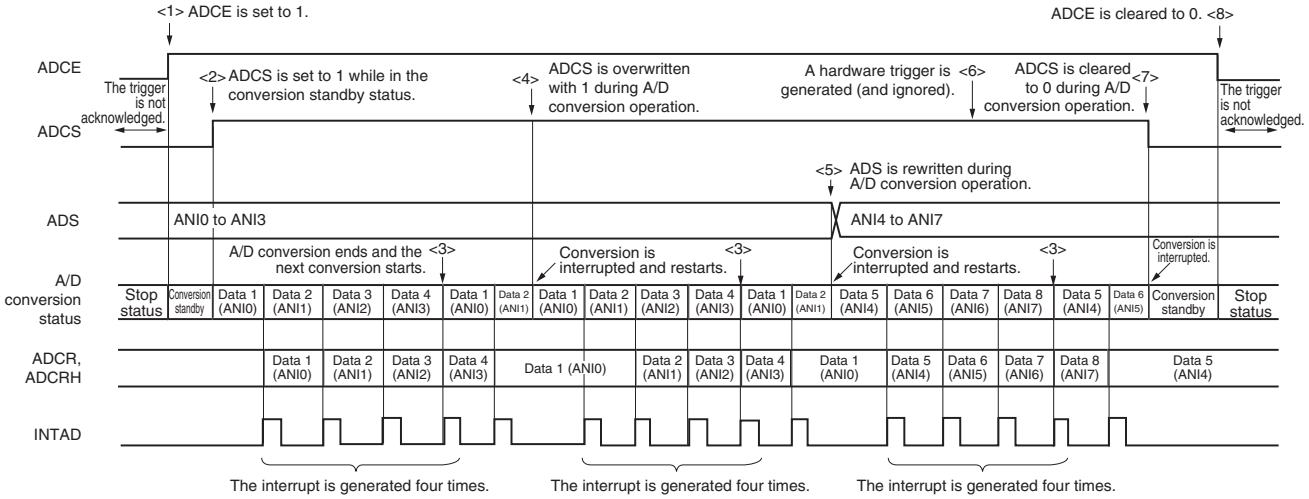
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

**Figure 12-24. Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing**



### 12.6.4 Software trigger mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

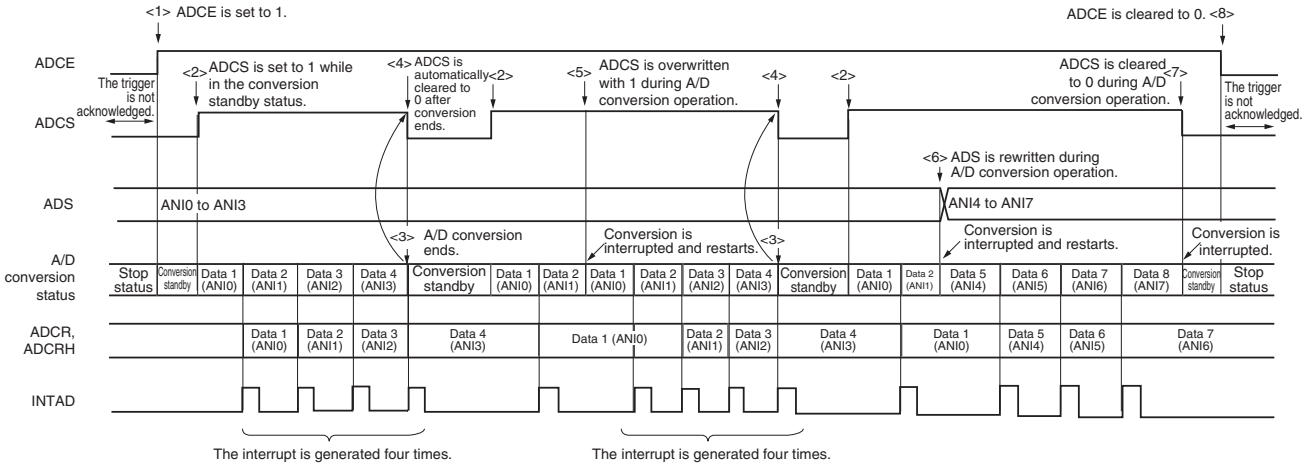
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

**Figure 12-25. Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing**



### 12.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

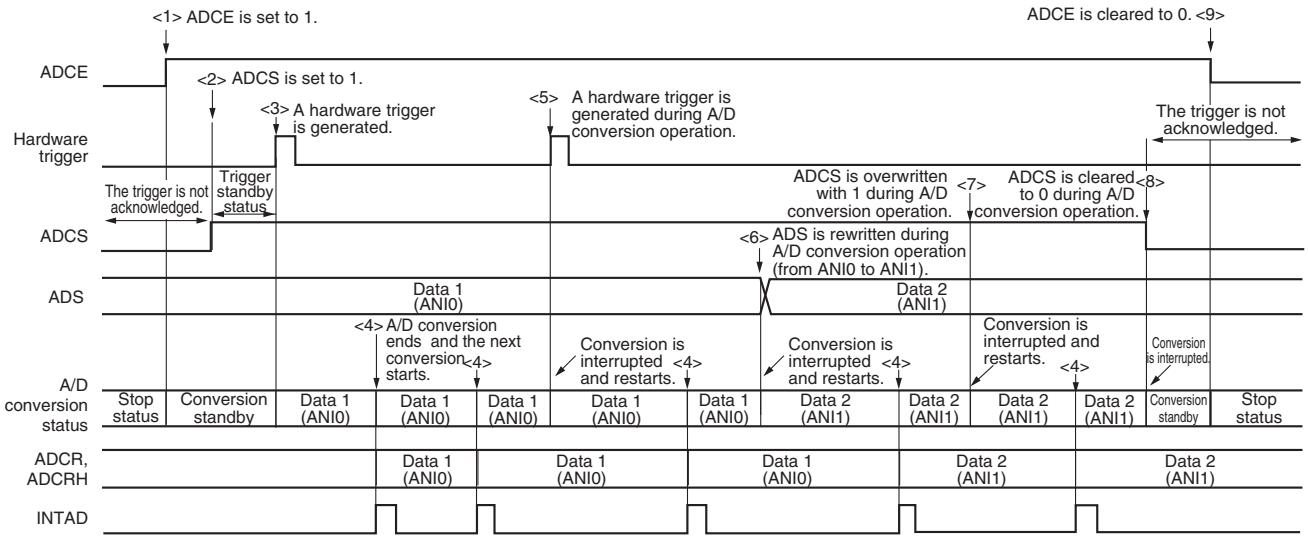
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

**Figure 12-26. Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing**



### 12.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

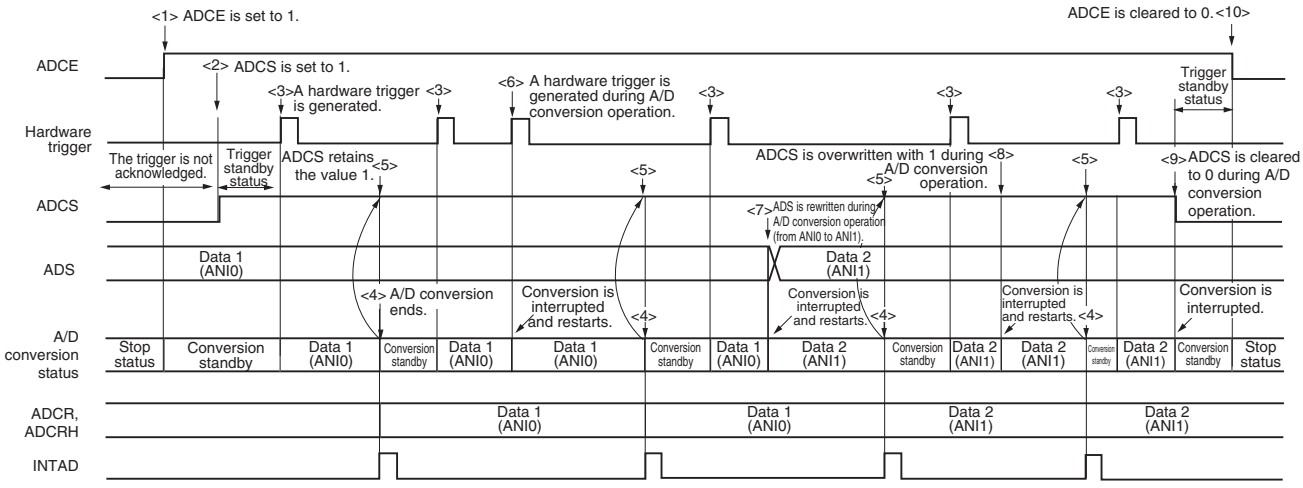
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

**Figure 12-27. Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing**



### 12.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

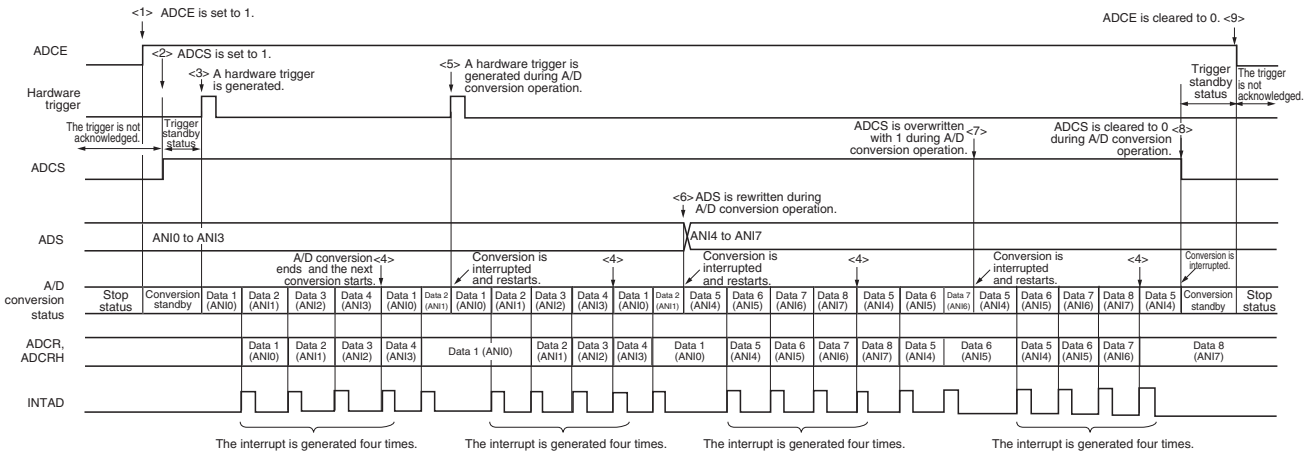
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

**Figure 12-28. Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing**



### 12.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

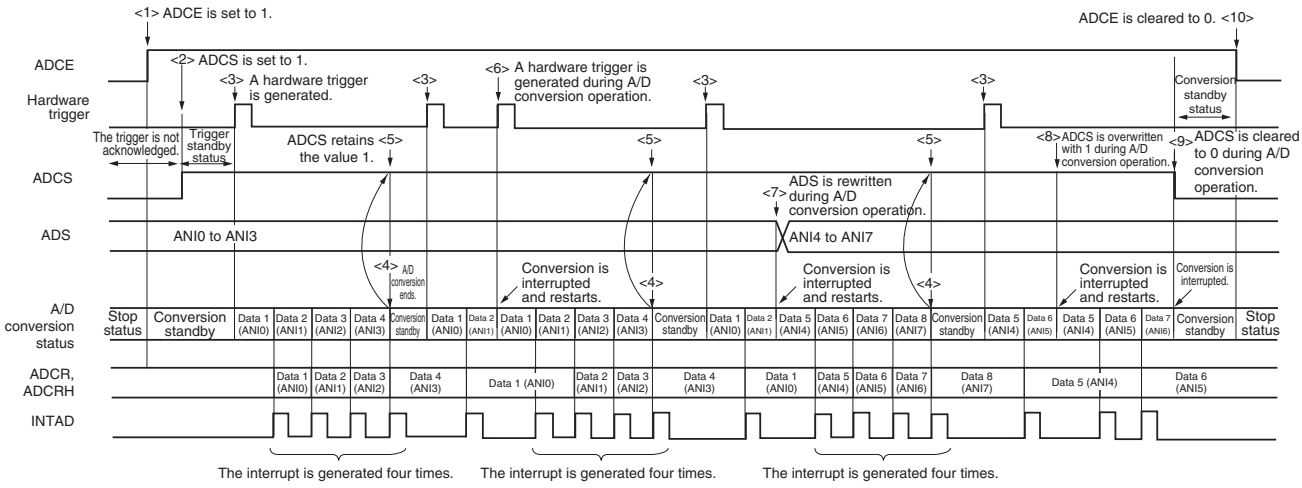
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

**Figure 12-29. Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing**



### 12.6.9 Hardware trigger wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)

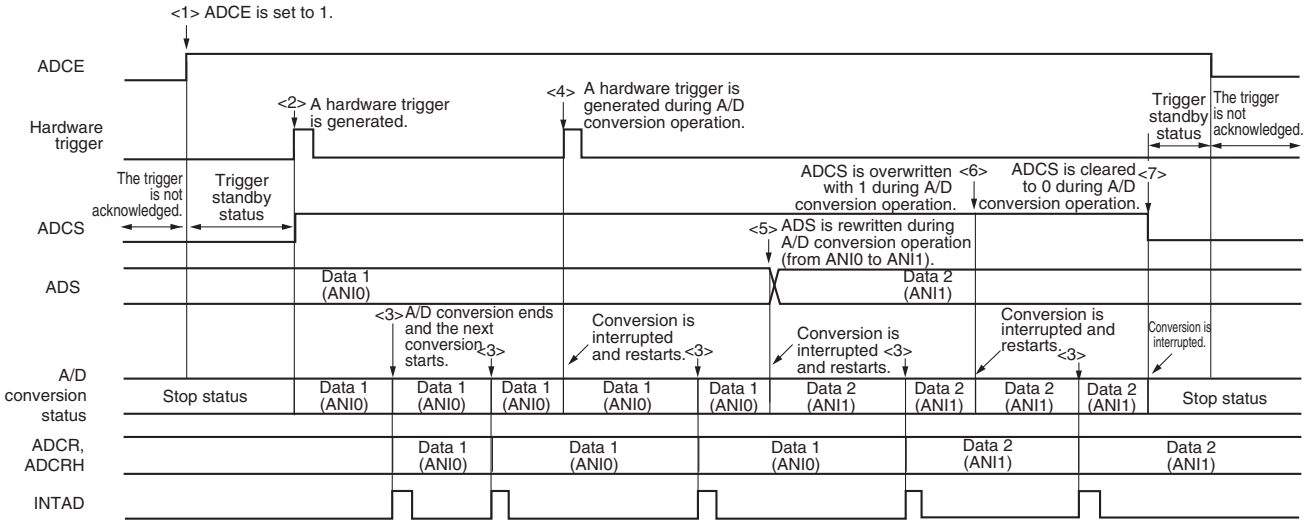
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

**Figure 12-30. Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing**



### 12.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

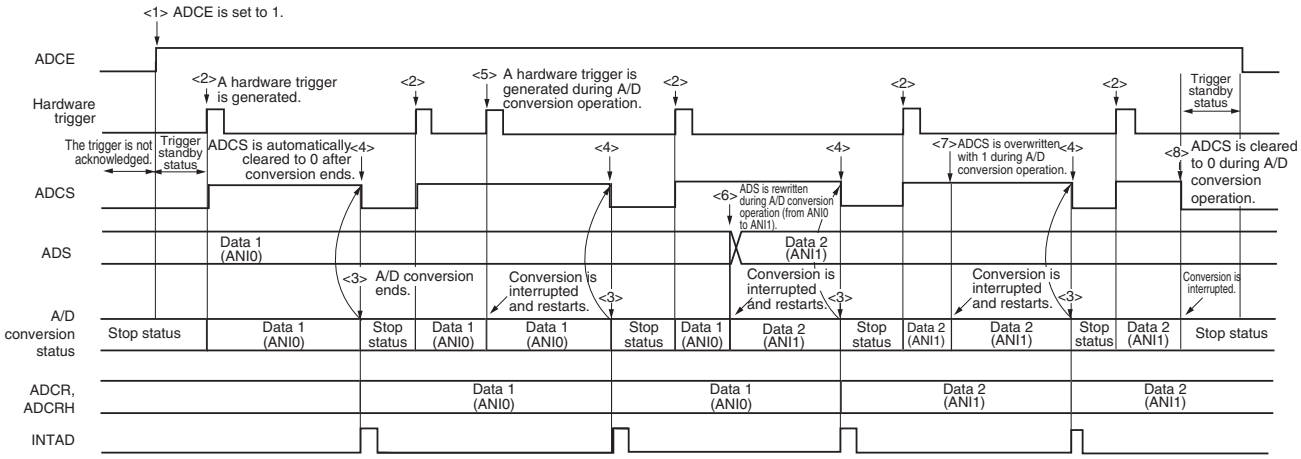
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is initialized.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

**Figure 12-31. Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing**



### 12.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

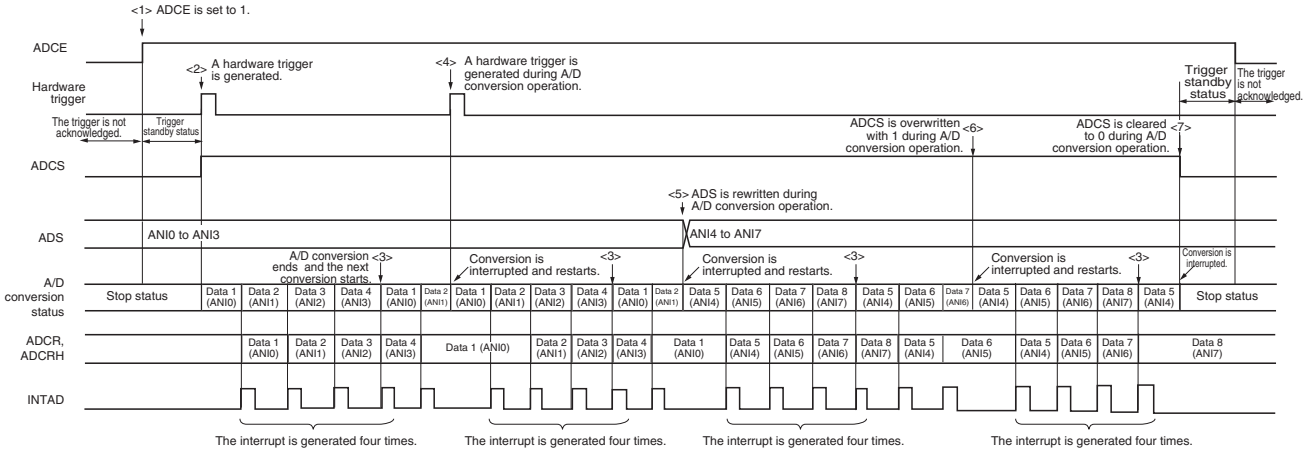
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

**Figure 12-32. Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing**



### 12.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

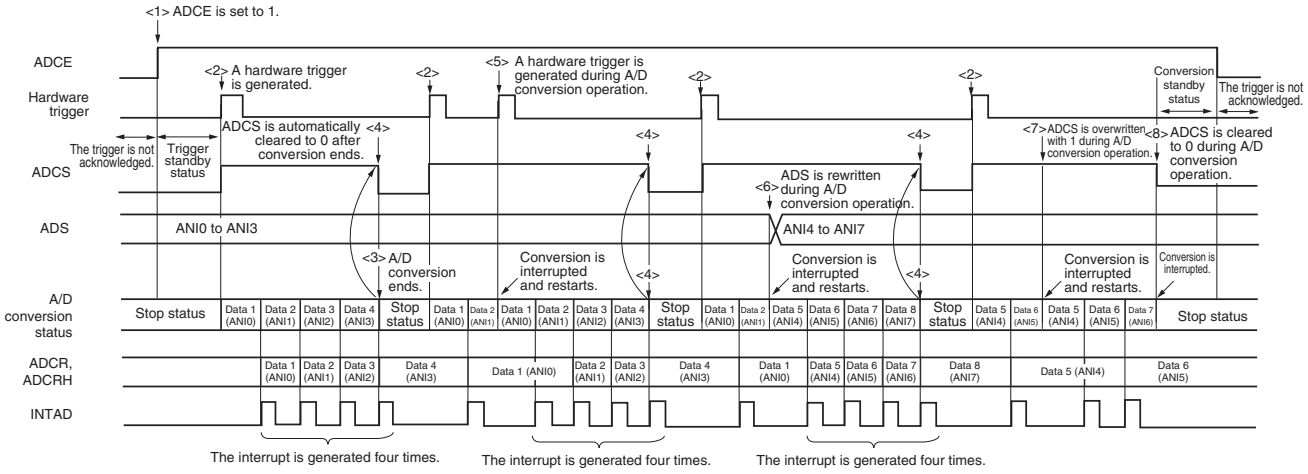
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

**Figure 12-33. Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing**

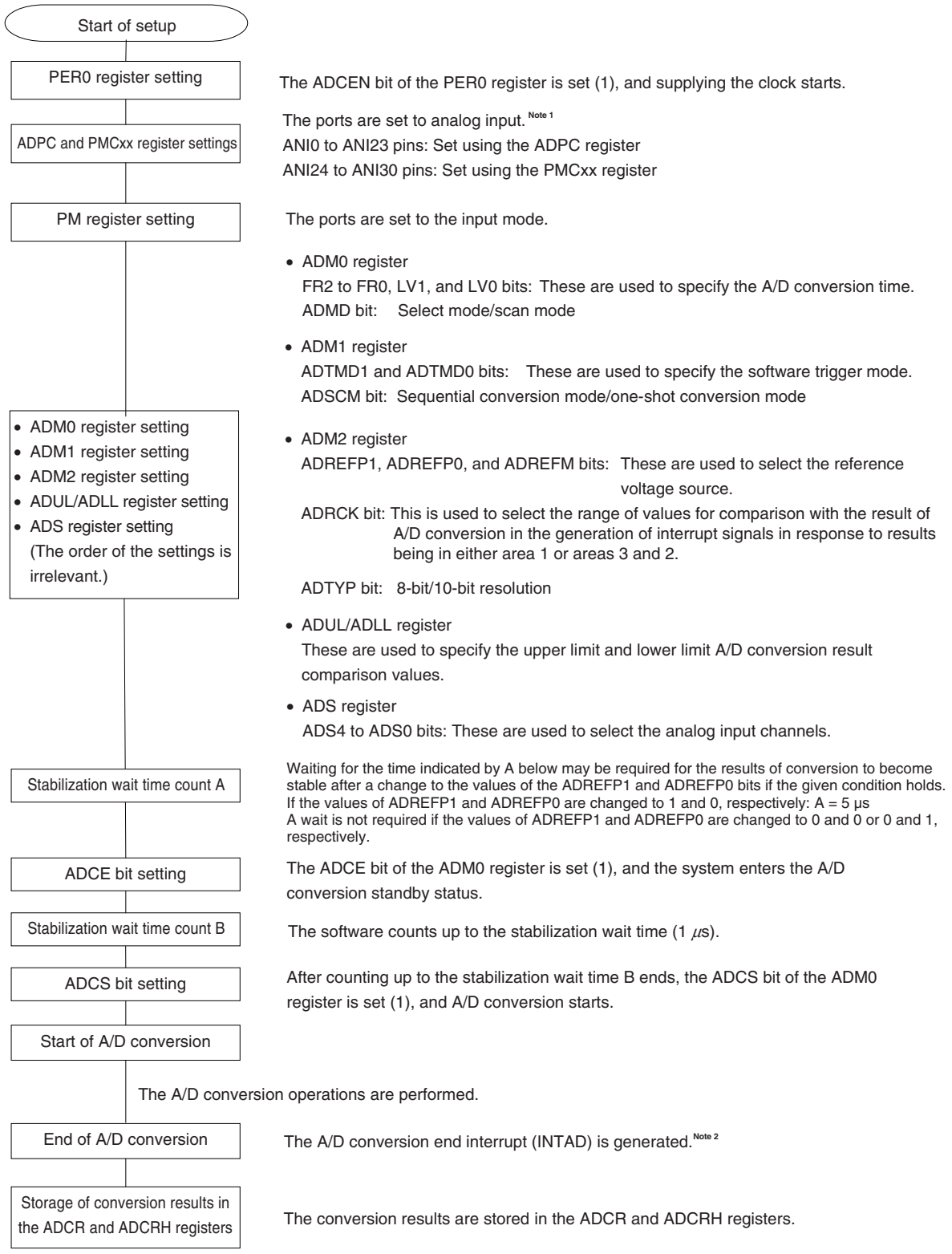


## 12.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

### 12.7.1 Setting up software trigger mode

**Figure 12-34. Setting up Software Trigger Mode**

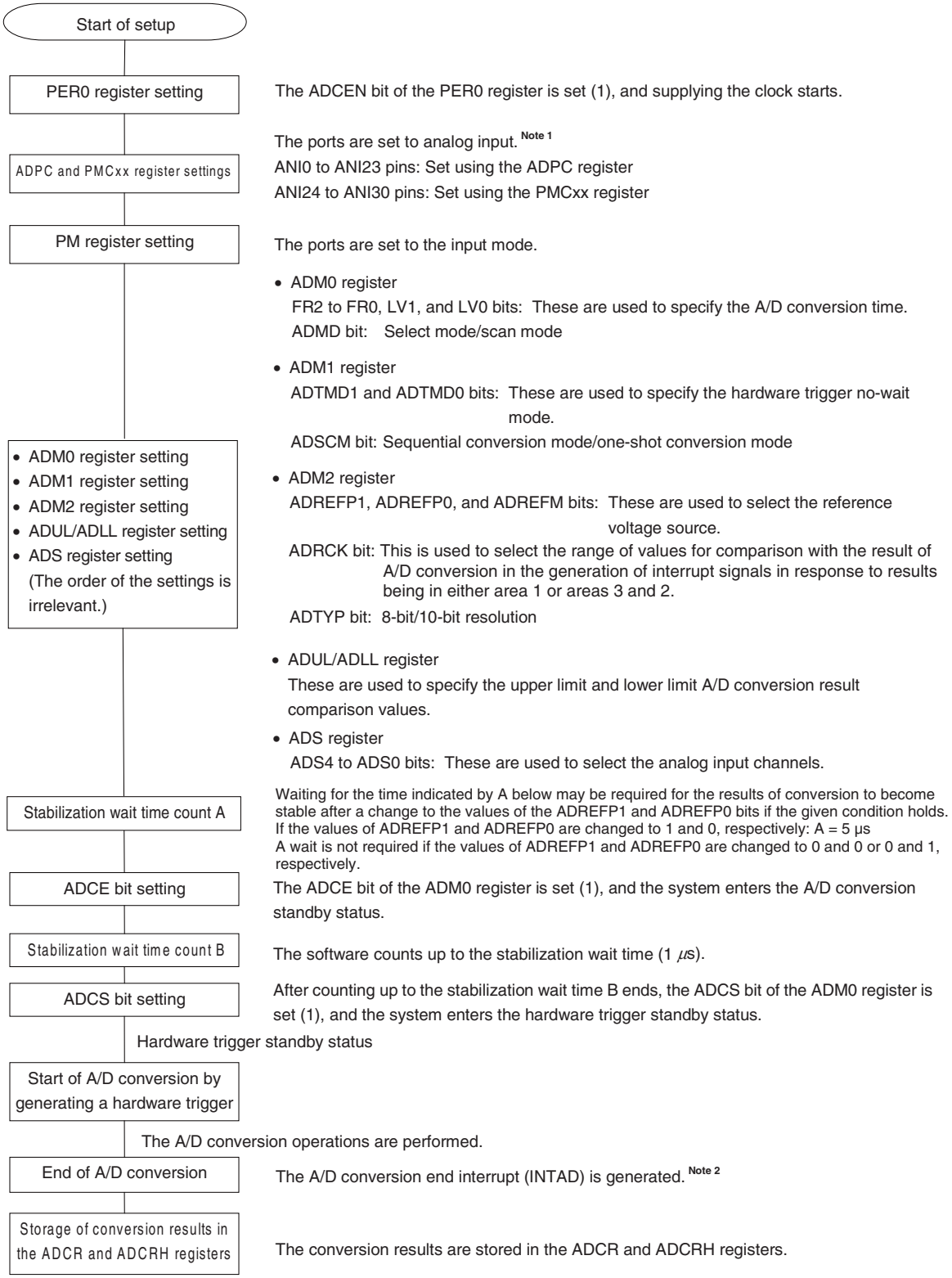



**Notes 1.** Depends on the products.

**2.** Depending on the settings of the ADRCK bit and ADUL/ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

### 12.7.2 Setting up hardware trigger no-wait mode

**Figure 12-35. Setting up Hardware Trigger No-Wait Mode**

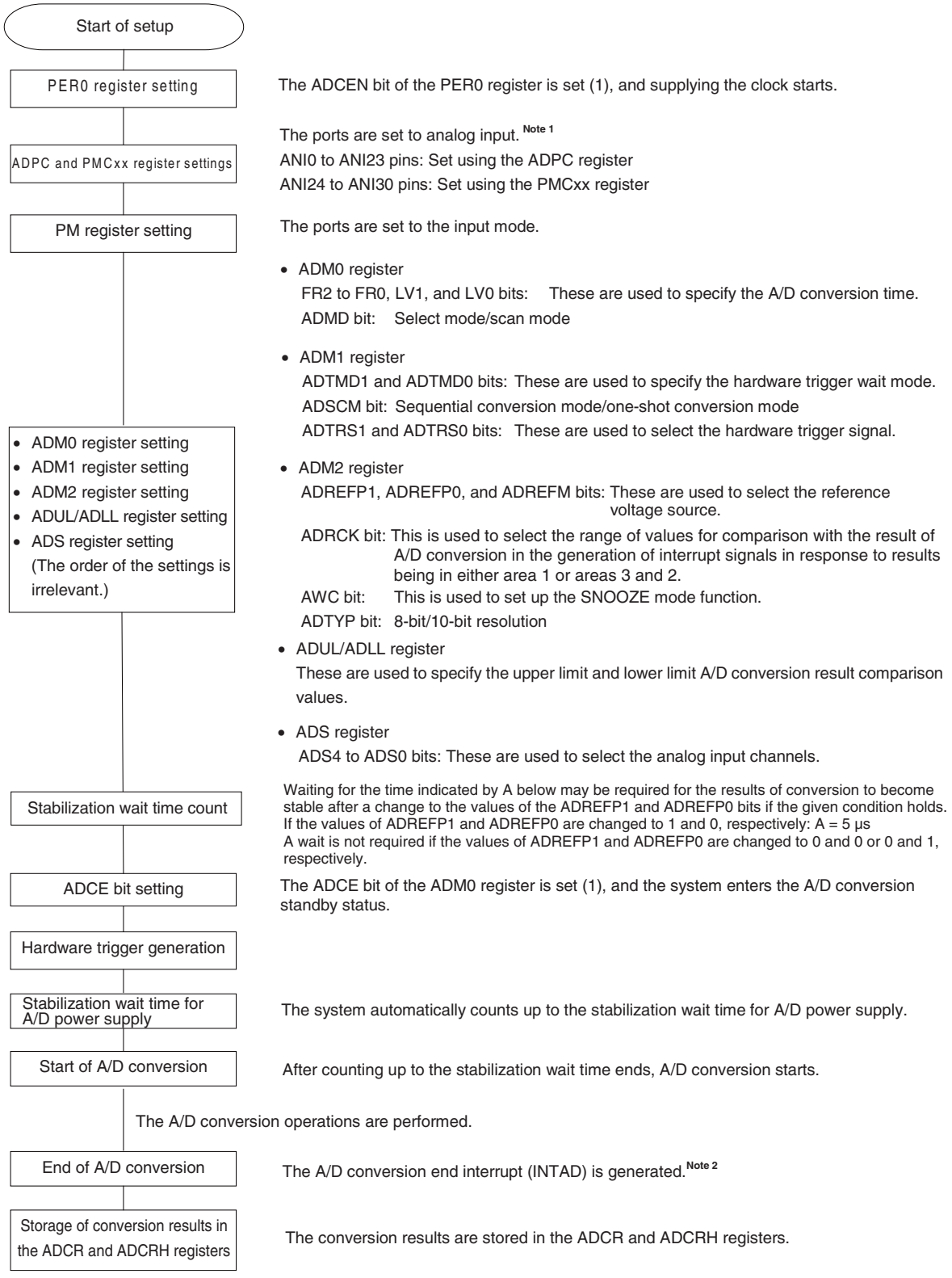



**Notes 1.** Depends on the products.

**2.** Depending on the settings of the ADRCK bit and ADUL/ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

### 12.7.3 Setting up hardware trigger wait mode

**Figure 12-36. Setting up Hardware Trigger Wait Mode**

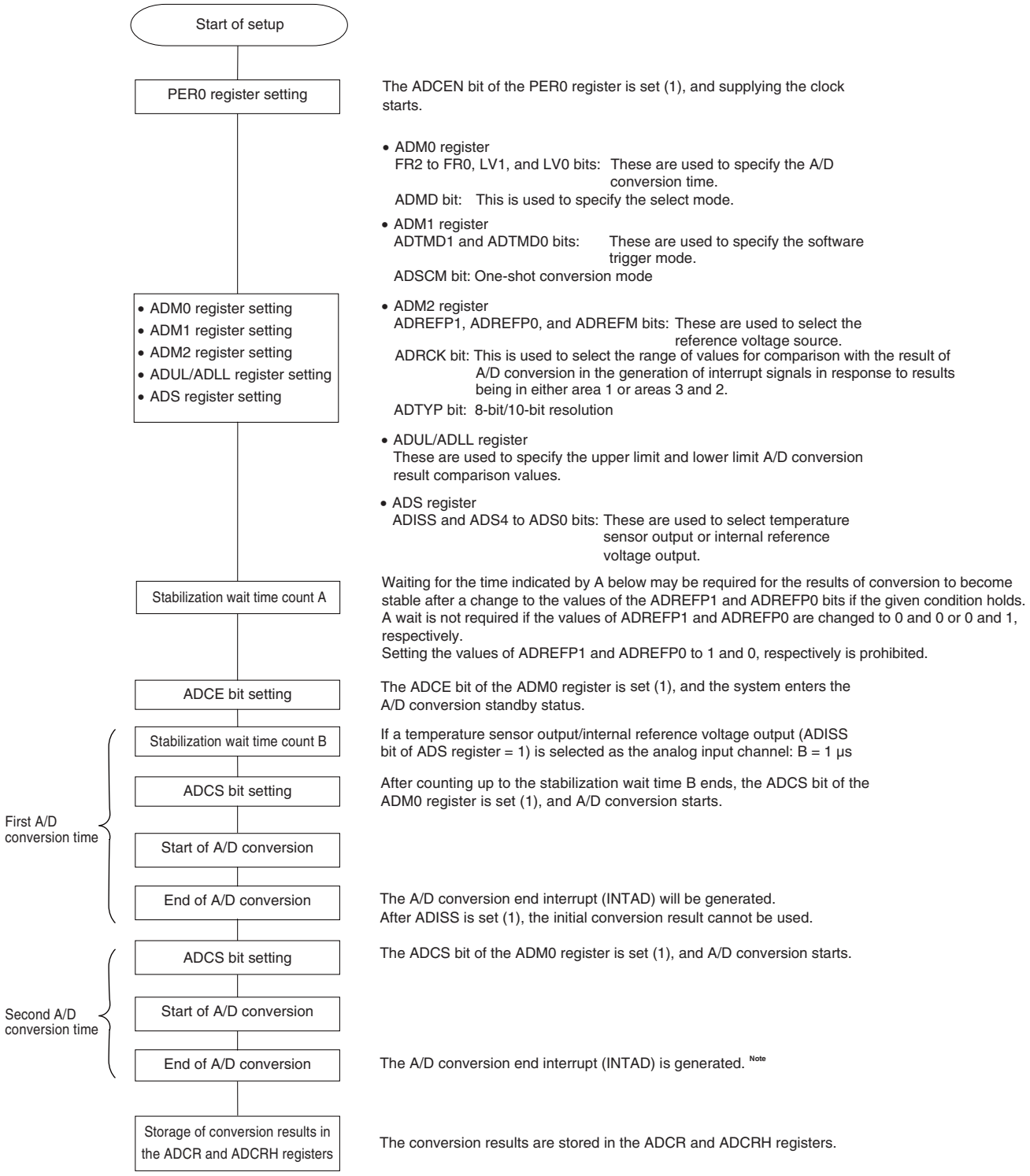



**Notes 1.** Depends on the products.

**2.** Depending on the settings of the ADRCK bit and ADUL/ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

### 12.7.4 Setup when using temperature sensor (example for software trigger mode and one-shot conversion mode)

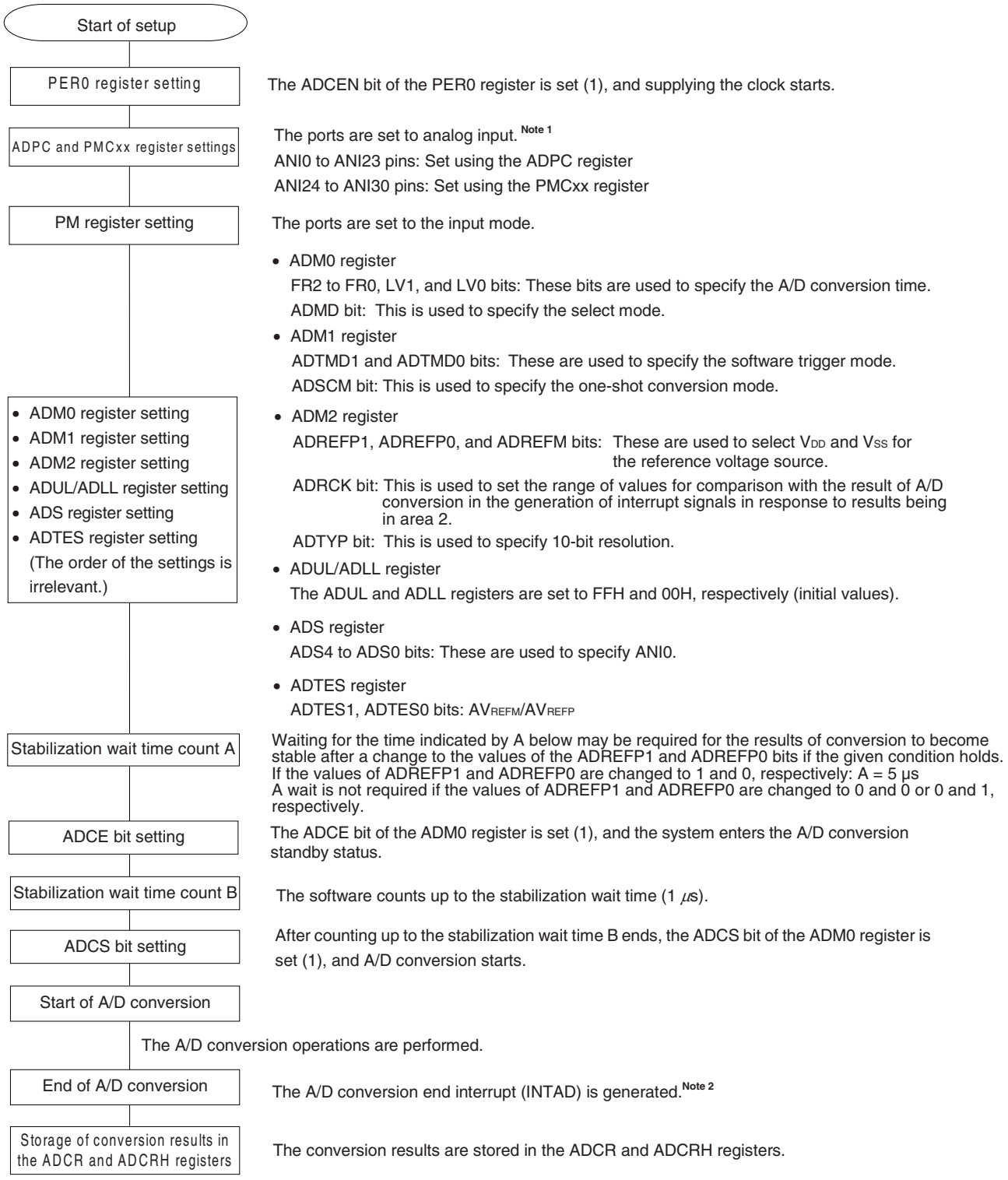
**Figure 12-37. Setup When Using Temperature Sensor**




**Note** Depending on the settings of the ADRCK bit and ADUL/ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

### 12.7.5 Setting up test mode

**Figure 12-38. Setting up Test Trigger Mode**




**Notes 1.** Depends on the products.

**2.** Depending on the settings of the ADRCK bit and ADUL/ADLL registers, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

## 12.8 SNOOZE Mode Function

In the SNOOZE mode, A/D conversion is triggered by inputting a hardware trigger in the STOP mode. Normally, A/D conversion is stopped while in the STOP mode, but, by using the SNOOZE mode, A/D conversion can be performed without operating the CPU by inputting a hardware trigger. This is effective for reducing the operation current.

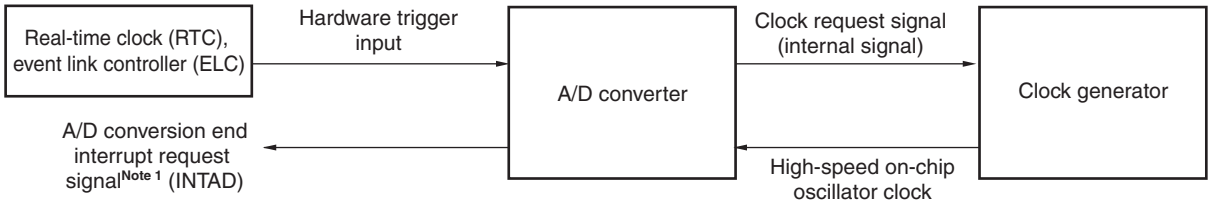
If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time in SNOOZE mode. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

In the SNOOZE mode, only the following two conversion modes can be used:

- Hardware trigger wait mode (select mode, one-shot conversion mode)
- Hardware trigger wait mode (scan mode, one-shot conversion mode)

**Caution The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for $f_{CLK}$.**

**Figure 12-39. Block Diagram When Using SNOOZE Mode Function**




When using the SNOOZE mode function, the initial setting of each register is specified before switching to the STOP mode. (For details about these settings, see **12.7.3 Setting up hardware trigger wait mode Note 2**.) At this time, bit 2 (AWC) of A/D converter mode register 2 (ADM2) is set to 1. After the initial settings are specified, bit 0 (ADCE) of A/D converter mode register 0 (ADM0) is set to 1.

If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the system automatically counts up to the stabilization wait time, and then A/D conversion starts.

The SNOOZE mode operation after A/D conversion ends differs depending on whether an interrupt signal is generated Note 1.

**Notes 1.** Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL/ADLL register), there is a possibility of no interrupt signal being generated.

**2.** Be sure to set the ADM1 register to E1H, E2H, or E3H.

**Remark** The hardware trigger is event selected by ELC, INTRTC, or INTTM01.
Specify the hardware trigger by using the A/D converter mode register 1 (ADM1).

### 12.8.1 If an interrupt is generated after A/D conversion ends

If the A/D conversion result value is inside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is generated.

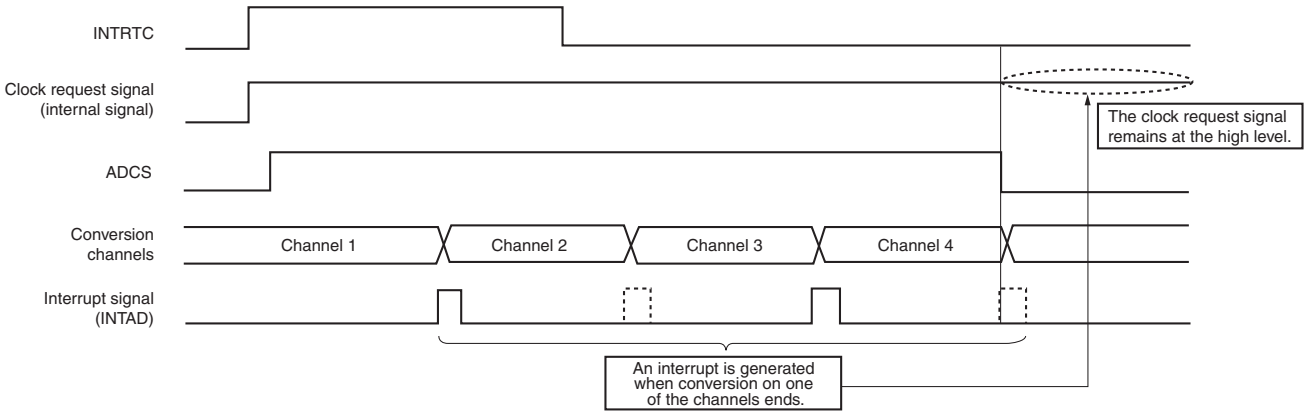
- While in the select mode

  When A/D conversion ends and an A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter returns to normal operation mode from SNOOZE mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of the A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

- While in the scan mode

  If even one A/D conversion end interrupt request signal (INTAD) is generated during A/D conversion of the four channels, the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of the A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

**Figure 12-40. Operation Example When Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)**



### 12.8.2 If no interrupt is generated after A/D conversion ends

If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is not generated.

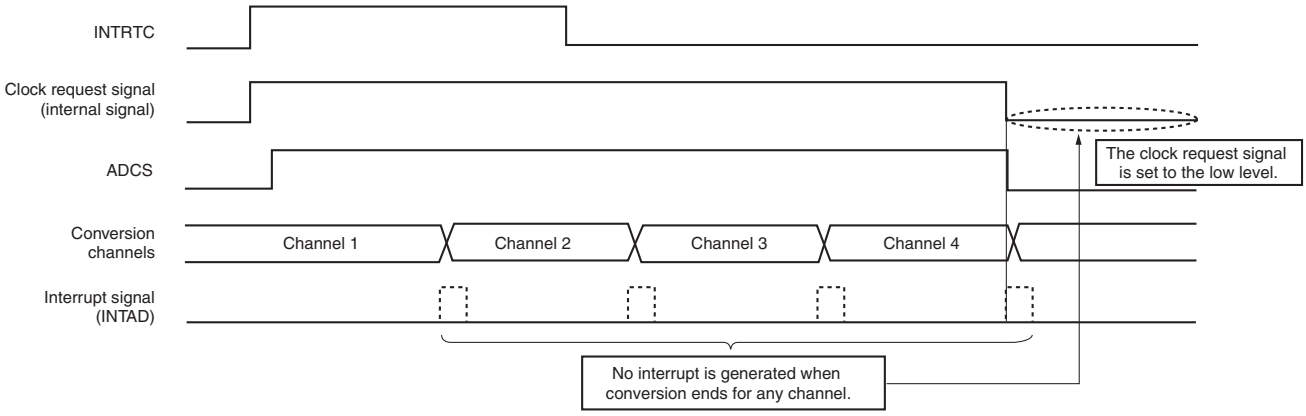
- While in the select mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

- While in the scan mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated even once during A/D conversion of the four channels, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels ends, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

**Figure 12-41. Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)**



## 12.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

**(1) Resolution**

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

**(2) Overall error**

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

**(3) Quantization error**

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.
Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

**Figure 12-42. Overall Error**

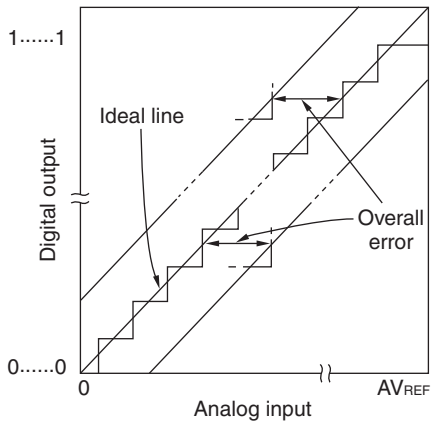
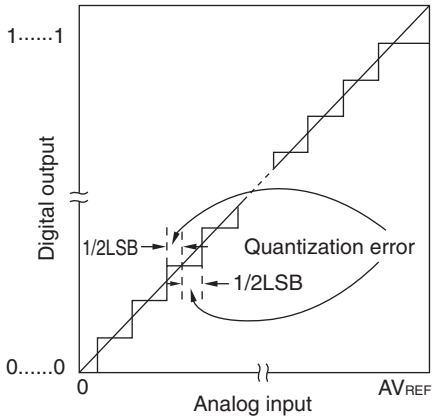



**Figure 12-43. Quantization Error**




**(4) Zero-scale error**

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.
If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

**(5) Full-scale error**

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

**(6) Integral linearity error**

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

**(7) Differential linearity error**

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

**Figure 12-44. Zero-Scale Error**

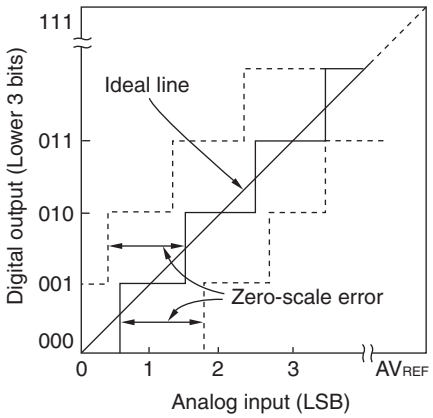



**Figure 12-45. Full-Scale Error**

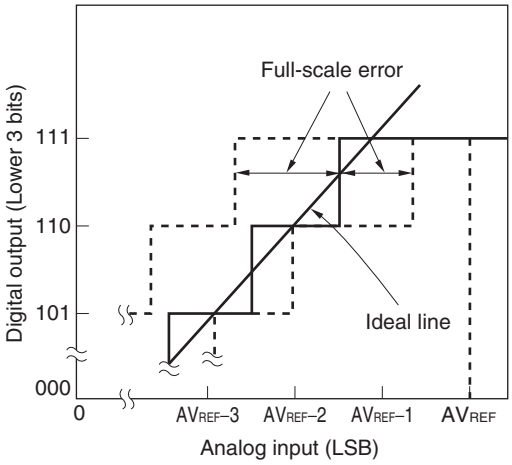



**Figure 12-46. Integral Linearity Error**

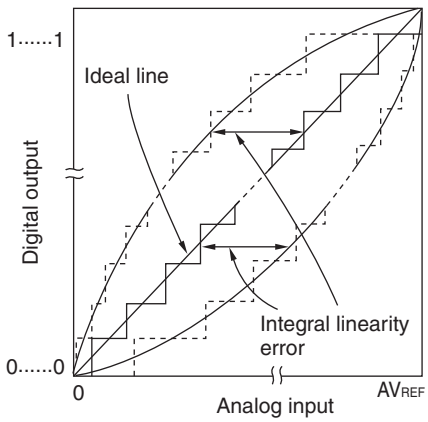
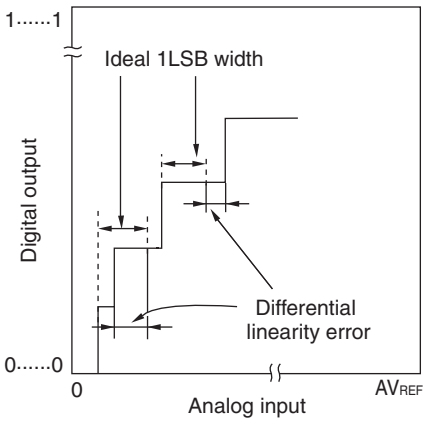



**Figure 12-47. Differential Linearity Error**



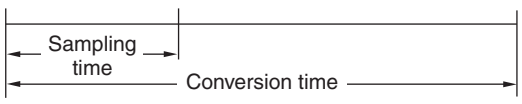

**(8) Conversion time**

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

**(9) Sampling time**

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



## 12.10 Cautions for A/D Converter

**(1) Operating current in STOP mode**

Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the ADM0 register to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1H (IF1H) to 0 and start operation.

**(2) Input range of ANI0 to ANI23 and ANI24 to ANI30 pins**

Observe the rated range of the ANI0 to ANI23 and ANI24 to ANI30 pins input voltage. If a voltage higher than $V_{DD}$ and $AV_{REFP}$ and less than $V_{SS}$ and $AV_{REFM}$ (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

When internal reference voltage (1.45 V) is selected reference voltage source for the + side of the A/D converter, do not input a voltage higher than the internal reference voltage to a pin selected by the ADS register. However, it is no problem that a voltage higher than the internal reference voltage is input to a pin not selected by the ADS register.

**(3) Conflicting operations**

<1> Conflict between the A/D conversion result register (ADCR, ADCRH) write and the ADCR or ADCRH register read by instruction upon the end of conversion

The ADCR or ADCRH register read has priority. After the read operation, the new conversion result is written to the ADCR or ADCRH registers.

<2> Conflict between the ADCR or ADCRH register write and the A/D converter mode register 0 (ADM0) write, the analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion

The ADM0, ADS, or ADPC registers write has priority. The ADCR or ADCRH register write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

**(4) Noise countermeasures**

To maintain the 10-bit resolution, attention must be paid to noise input to the $AV_{REFP}$, $V_{DD}$, ANI0 to ANI23, and ANI24 to ANI30 pins.

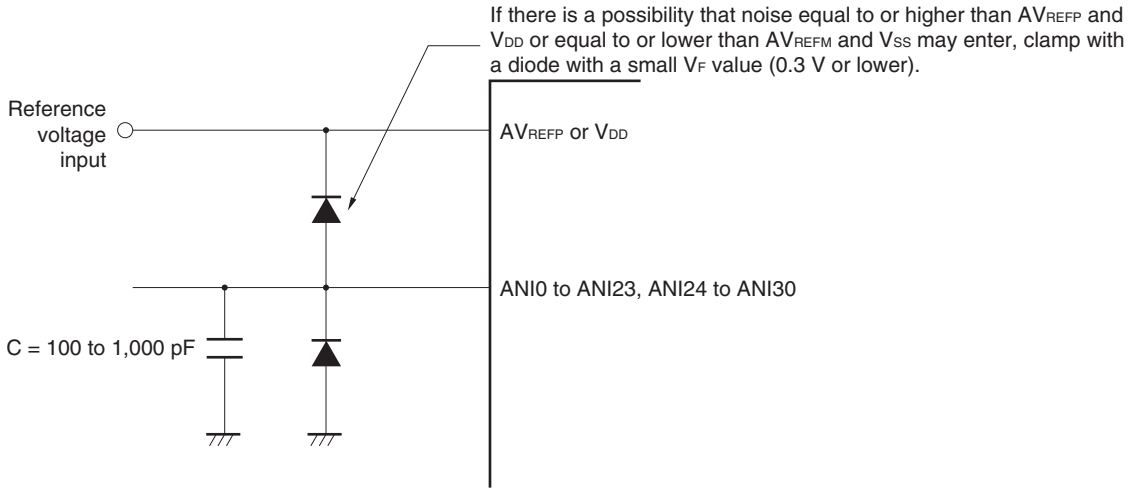
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external C as shown in Figure 12-48 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

**Figure 12-48. Analog Input Pin Connection**




**(5) Analog input (ANIn) pins**

<1> The analog input pins (ANI0 to ANI23, ANI24 to ANI30) are also used as input port pins (P33, P34, P70 to P74, P80 to P87, P90 to P97, P100 to P105, P120, P125).
Do not change the output values for the port-pin functions P33, P34, P70 to P74, P80 to P87, P90 to P97, P100 to P105, P120, and P125 while A/D conversion of the signals on the ANI0 to ANI23 or ANI24 to ANI30 pins is selected and conversion is in progress, since doing so may lower the precision of the results of conversion.

<2> When a pin adjacent to one on which A/D conversion is in progress is used as a digital I/O port pin, coupling may lead to noise that causes the results of A/D conversion to differ from the expected values. Be sure to prevent the input or output of pulses on such pins while conversion is in progress.

**(6) Input impedance of analog input (ANIn) pins**

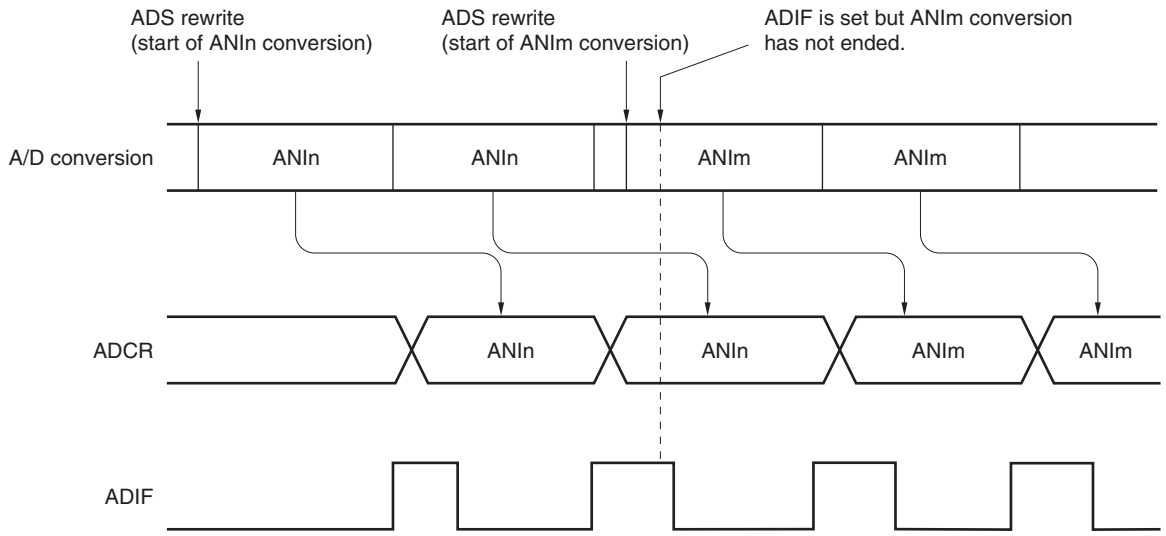
This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 1 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI23 and ANI24 to ANI30 pins (see **Figure 12-48**).

**(7) Interrupt request flag (ADIF)**

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed. Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF flag for the pre-change analog input may be set just before the ADS register rewrite. Caution is therefore required since, at this time, when ADIF flag is read immediately after the ADS register rewrite, ADIF flag is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

**Figure 12-49. Timing of A/D Conversion End Interrupt Request Generation**




**(8) Conversion results just after A/D conversion start**

While in the software trigger mode or hardware trigger no-wait mode, the first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

**(9) A/D conversion result register (ADCR, ADCRH) read operation**

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), A/D port configuration register (ADPC), and port mode control register xx (PMCxx), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, ADPC, or PMC register. Using a timing other than the above may cause an incorrect conversion result to be read.

**(10) Internal equivalent circuit**

The equivalent circuit of the analog input block is shown below.

**Figure 12-50. Internal Equivalent Circuit of ANIn Pin**

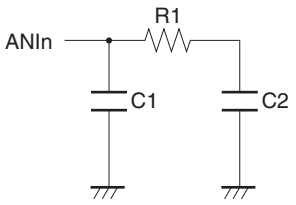



**Table 12-6. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)**

| $AV_{REFP}$, $V_{DD}$ | ANIn Pins | R1 [kΩ] | C1 [pF] | C2 [pF] |
|---|---|---|---|---|
| 3.6 V ≤ $V_{DD}$ ≤ 5.5 V | ANI0 to ANI23 | 14 | 8 | 2.5 |
| | ANI24 to ANI30 | 18 | 8 | 7.0 |
| 2.7 V ≤ $V_{DD}$ < 3.6 V | ANI0 to ANI23 | 39 | 8 | 2.5 |
| | ANI24 to ANI30 | 53 | 8 | 7.0 |

**Remark** The resistance and capacitance values shown in Table 12-6 are not guaranteed values.

**(11) Starting the A/D converter**

Start the A/D converter after the $AV_{REFP}$ and $V_{DD}$ voltages stabilize.

**(12) Temperature sensor output**

If the internal reference voltage (1.45 V) is selected as the reference voltage of comparator 0 or comparator 1 in RL78/F14, the temperature sensor output cannot be selected.

# CHAPTER 13 D/A CONVERTER (RL78/F14 Only)

The D/A converter is an 8-bit resolution R-2R type unit used to control analog outputs.

## 13.1 Function of D/A Converter

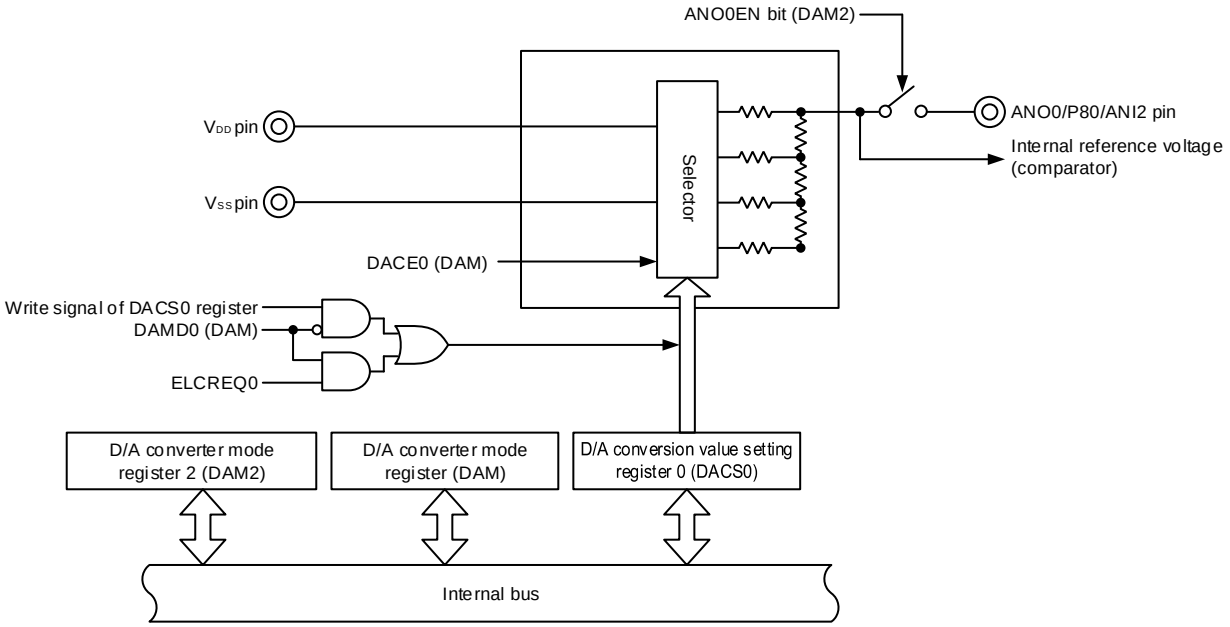
The D/A converter has the following features.

- 8-bit resolution
- R-2R ladder type
- Analog output voltage
  8-bit resolution: $V_{DD}$ × m8/256 (m8: Value set to DACS0 register)
- Operation mode
  Normal mode
  Real-time output mode

## 13.2 Configuration of D/A Converter

Figure 13-1 shows the block diagram of the D/A converter.

**Figure 13-1. Block Diagram of D/A Converter**




**Remarks 1.** ELCREQ0 is a trigger signal (request signal from the ELC) that is used in the real-time output mode.

**2.** The internal reference voltage (comparator) is used to select the reference voltage of the comparator. When setting bits 5 and 4 (CVRS1 and CVRS0) in the comparator I/O select register (CMPSEL) to 10B (internal reference voltage (DAC output) is selected), set the ANO0EN bit in this register to 0 (analog output is disabled).

## 13.3 Registers of D/A Converter

The D/A converter uses the following six registers.

- A/D port configuration register (ADPC)
- Peripheral enable register 1 (PER1)
- D/A converter mode register (DAM)
- D/A converter mode register 2 (DAM2)
- D/A conversion value setting register 0 (DACS0)
- Port mode register 8 (PM8)

### 13.3.1 A/D Port Configuration Register (ADPC)

This register switches the ANI0/P33 to ANI23/P105 pins to either analog input or port digital I/O.

When the D/A converter is used, this register should be used to set the ANI2/ANO0/P80 pin to analog input.

Set the ADPC register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 13-2. Format of A/D Port Configuration Register (ADPC)**

Address: F0076H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADPC | 0 | 0 | 0 | ADPC4 | ADPC3 | ADPC2 | ADPC1 | ADPC0 |

| ADPC4 | ADPC3 | ADPC2 | ADPC1 | ADPC0 | Switching between analog input (A) and digital input/output (D) | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | ANI23/P105 | ANI22/P104 | ANI21/P103 | ANI20/P102 | ANI19/P101 | ANI18/P100 | ANI17/P97 | ANI16/P96 | ANI15/P95 | ANI14/P94 | ANI13/P93 | ANI12/P92 | ANI11/P91 | ANI10/P90 | ANI9/P87 | ANI8/P86 | ANI7/IVREF0/P85 | ANI6/IVCMP03/P84 | ANI5/IVCMP02/P83 | ANI4/IVCMP01/P82 | ANI3/IVCMP00/P81 | ANI2/ANO0/P80 | ANI1/P34 | ANI0/P33 |
| 0 | 0 | 0 | 0 | 0 | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 0 | 0 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D |
| 0 | 0 | 0 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A |
| 0 | 0 | 0 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A |
| 0 | 0 | 1 | 0 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A |
| 0 | 0 | 1 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A |
| 0 | 0 | 1 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A |
| 0 | 0 | 1 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A |
| 0 | 1 | 0 | 0 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A |
| 0 | 1 | 0 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A |
| 0 | 1 | 0 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A |
| 0 | 1 | 0 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 0 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 0 | 0 | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 0 | 1 | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 1 | 0 | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 1 | 1 | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 0 | 0 | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 0 | 1 | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 1 | 0 | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 1 | 1 | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 1 | 0 | 0 | 0 | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| Other than the above | | | | | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | |

**Cautions 1. Set a channel to be used for D/A conversion to the input mode by using port mode register 8 (PM8).**

**2. Do not set the pin that is set by the ADPC register as digital I/O to D/A conversion operation enable by using the D/A converter mode register (DAM).**

### 13.3.2 Peripheral Enable Register 1 (PER1)

The PER1 register enables or disables clock supply to each peripheral hardware unit. Clock supply to a hardware unit that is not used is stopped in order to reduce the power consumption and noise.

When the D/A converter is used, be sure to set bit 7 (DACEN) of this register to 1.

Set the PER1 register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 13-3. Format of Peripheral Enable Register 1 (PER1)**

Address: F007AH After reset: 00H

| Symbol | <7> | <6> | <5> | <4> | <3> | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PER1 | DACEN Note | TRGEN | CMPEN Note | TRD0EN | DTCEN | 0 | 0 | TRJ0EN |

| DACEN Note | Control of D/A converter input clock | R/W |
|---|---|---|
| 0 | Stops input clock supply.<br>• SFR used by the D/A converter cannot be written to.<br>• The D/A converter is in the reset state. | R/W |
| 1 | Supplies input clock.<br>• SFR used by the D/A converter can be read/written to. | |

**Note** Only for RL78/F14.

**Cautions 1. When setting the D/A converter, be sure to set the DACEN bit to 1 first. If DACEN = 0, writing to a control register of the D/A converter is ignored, and all read values are default values (except for port mode register 8 (PM8), port register 8 (P8), A/D port configuration register (ADPC), and D/A converter mode register 2 (DAM2)).**

**2. Be sure to clear the following bits to 0.**

**RL78/F13: bits 1, 2, 5, 6, and 7**

**RL78/F14: bits 1, 2, and 6**

### 13.3.3 D/A Converter Mode Register (DAM)

This register controls the operation of the D/A converter.

Set the DAM register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 13-4. Format of D/A Converter Mode Register (DAM)**

Address: FFF36H After reset: 00H

| Symbol | 7 | 6 | <5> | <4> | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DAM | – | – | – | DACE0 | – | – | – | DAMD0 |

| DACE0 | D/A conversion operation control | R/W |
|---|---|---|
| 0 | Stops D/A conversion operation. | R/W |
| 1 | Enables D/A conversion operation. | |

| DAMD0 | D/A converter operation mode selection | R/W |
|---|---|---|
| 0 | Normal mode | R/W |
| 1 | Real-time output mode | |

When the D/A converter is not used, set the DACE0 bit to 0 (output disable) and set the DACS0 register to 00H to prevent current from flowing into the R-2R resistor ladder to reduce unnecessary current consumption.

### 13.3.4 D/A Converter Mode Register 2 (DAM2)

When the P80/ANO0 pin is in use to output analog signal from the D/A converter, this register is used to control the output from the ANO0 pin. When setting bits 5 and 4 (CVRS1 and CVRS0) in the comparator I/O select register (CMPSEL) to 10B (internal reference voltage (DAC output) is selected), set the ANO0EN bit in this register to 0 (analog output is disabled).

Set the DAM2 register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 13-5. Format of D/A Converter Mode Register 2 (DAM2)**

Address: F0227H After reset: 00H R/W

| Symbol | 7 | 6 | <5> | <4> | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DAM2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ANO0EN |

| ANO0EN | Analog output (ANO0) control | R/W |
|---|---|---|
| 0 | Disables analog output (ANO0). | R/W |
| 1 | Enables analog output (ANO0). | |

### 13.3.5 D/A Conversion Value Setting Register 0 (DACS0)

This register is used to set the analog voltage value to be output to the ANO0 pin when the D/A converter is used.

Set the DACS0 register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 13-6. Format of D/A Conversion Value Setting Register 0 (DACS0)**

Address: FFF34H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DACS0 | DACS07 | DACS06 | DACS05 | DACS04 | DACS03 | DACS02 | DACS01 | DACS00 |

**Remark** The analog output voltage (VANO0) of the D/A converter is defined as follows.

VANO0 = Reference voltage for D/A converter × (DACS0)/256

### 13.3.6 Port Mode Register 8 (PM8)

When using the ANO0/ANI2/P80 pin as an analog input port, set bit PM80 to 1.

If bit PM80 is set to 0, this pin cannot be used as an analog input port.

Set the PM8 register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

**Caution If a pin is set as an analog input port, not the pin level but 0 is always read.**

**Figure 13-7. Format of Port Mode Register 8 (PM8)**

Address: FFF22H After reset: FFH

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM8 | PM87 | PM86 | PM85 | PM84 | PM83 | PM82 | PM81 | PM80 |

| PM8n | P8n pin I/O mode selection (n = 0 to 7) | R/W |
|---|---|---|
| 0 | Output mode (output buffer on) | R/W |
| 1 | Input mode (output buffer off) | |

The function of the ANO0/ANI2/P80 pin can be selected by using the A/D port configuration register (ADPC), the D/A converter mode register (DAM), D/A converter mode register 2 (DAM2), the analog input channel specification register (ADS), and the PM8 register.

<R>

**Table 13-1. Setting Functions of ANO0/ANI2/P80 Pin**

| ADPC Register | PM8 Register | DAM Register | DAM2 Register | ADS Register | Functions of ANO0/ANI2/P80 Pin |
|---|---|---|---|---|---|
| Digital I/O | Input mode | — | Enables analog output | — | Setting prohibited |
| | | | Disables analog output | | Digital input |
| | Output mode | — | Enables analog output | — | Setting prohibited |
| | | | Disables analog output | | Digital input |
| Analog I/O | Input mode | Enables D/A conversion operation | Enables analog output | Selects ANI | Setting prohibited |
| | | | | Does not selects ANI | Analog output (D/A conversion output) |
| | | | Disables analog output | Selects ANI | Analog input (to be converted) |
| | | | | Does not selects ANI | Analog input (not to be converted) Note |
| | | Stops D/A conversion operation | Enables analog output | Selects ANI | Setting prohibited |
| | | | | Does not selects ANI | Setting prohibited |
| | | | Disables analog output | Selects ANI | Analog input (to be converted) |
| | | | | Does not selects ANI | Analog input (not to be converted) |
| | Output mode | — | — | — | Setting prohibited |

**Note** This is a setting that the D/A converter is used for internal reference voltage of comparator. In this case, set CVRS1, CVRS0 bits of CMPSEL register to 10b (internal reference voltage (DAC output) is selected).

## 13.4 Operations of D/A Converter

### 13.4.1 Operation in Normal Mode

D/A conversion is performed using write operation to the DACS0 register as the trigger.

The setting method is described below.

<1> Set the DACEN bit of the peripheral enable register 1 (PER1) to 1 to start the supply of the input clock to the D/A converter.

<2> Use the A/D port configuration register (ADPC) to set the ports to analog pins.

<3> Set the ANO0EN bit of the D/A converter mode register 2 (DAM2) to 1 (analog output enable). When setting bits 5 and 4 (CVRS1 and CVRS0) in the comparator I/O select register (CMPSEL) to 10B (internal reference voltage (DAC output) is selected), set the ANO0EN bit in this register to 0 (analog output is disabled).

<4> Set the DAMD0 bit of the D/A converter mode register (DAM) to 0 (normal mode).

<5> Set the analog voltage value to be output to the ANO0 pin to the D/A conversion value setting register 0 (DACS0).

Steps <1> to <5> above constitute the initial settings.

<6> Set the DACE0 bit of the DAM register to 1 (D/A conversion enable).
D/A conversion starts, and then, after the settling time elapses, the analog voltage set in step <5> is output to the ANO0 pin.

<7> To perform subsequent D/A conversions, write to the DACS0 register.

The previous D/A conversion result is held until the next D/A conversion is performed.

When the DACE0 bit of the DAM register is set to 0 (D/A conversion operation stop), D/A conversion stops.

If the ports are set to digital pins using the ADPC register, the ANO0 pin goes into a high-impedance state when the PM80 bit of the PM8 register for the port = 1 (input mode), and the ANO0 pin outputs the set value of the P8 register when the PM80 bit = 0 (output mode).

**Cautions 1. Even if 1, 0, and then 1 is set to the DACE0 bit, there is a wait after 1 is set for the last time.**

**2. If the DACS0 register is rewritten during the settling time, D/A conversion is aborted and reconversion by using the newly written values starts.**

### 13.4.2 Operation in Real-Time Output Mode

D/A conversion is performed on each channel using the individual interrupt request signals from the ELC as triggers.

The setting method is described below.

<1> Set the DACEN bit of the peripheral enable register 1 (PER1) to 1 to start the supply of the input clock to the D/A converter.

<2> Use the port configuration register (ADPC) to set the ports to analog pins.

<3> Set the ANO0EN bit of the D/A converter mode register 2 (DAM2) to 1 (analog output enable). When setting bits 5 and 4 (CVRS1 and CVRS0) in the comparator I/O select register (CMPSEL) to 10B (internal reference voltage (DAC output) is selected), set the ANO0EN bit in this register to 0 (analog output is disabled).

<4> Set the DAMD0 bit of the D/A converter mode register (DAM) to 0 (normal mode).

<5> Set the analog voltage value to be output to the ANO0 pin to the D/A conversion value setting register 0 (DACS0).

<6> Set the DACE0 bit of the DAM register to 1 (D/A conversion enable).

D/A conversion starts, and then, after the settling time elapses, the analog voltage set in step <5> is output to the ANO0 pin.

<7> Use the event output destination select register (ELSELRn) to set the real-time trigger signal.

<8> Set the DAMD0 bit of the DAM register to 1 (real-time output mode).

<9> Start the operation of the ECL request source.

Steps <1> to <9> above constitute the initial settings.

<10> Generation of the real-time output triggers starts D/A conversion and the analog voltage set in step <5> will be output to the ANO0 pin after a settling time has elapsed.

Set the analog voltage value to be output to the ANO0 pin, to the DACS0 register before performing the next D/A conversion (real-time output trigger is generated).

When the DACE0 bit of the DAM register is set to 0 (D/A conversion operation stop), D/A conversion stops.

If the ports are set to digital pins by using the ADPC register, the ANO0 pin goes into a high-impedance state when the PM80 bit of the PM8 register for the port = 1 (input mode), and the ANO0 pin outputs the set value of the P8 register when the PM80 bit = 0 (output mode).

**Cautions 1. Even if 1, 0, and then 1 is set to the DACE0 bit, there is a wait after 1 is set for the last time.**

**2. Make the interval between each generation of the ELC event request trigger signal longer than the settling time. If an ELC event request trigger signal is generated during the settling time, D/A conversion is aborted and reconversion starts.**

**3. Even if the generation of the ELC event request trigger signal and rewriting of the DACS0 register conflict, the correct D/A conversion result is output.**

## 13.5 Cautions for D/A Converter

Observe the following cautions when using the D/A converter.

(1) The digital port I/O function, which is the alternate function of the ANO0 pin, does not operate if the ports are set to analog pins by using the A/D port configuration register (ADPC). When the P8 register is read while the ports are set to analog pins by using the ADPC register, 0 is read in the input mode and the set value of the P8 register is read in the output mode. If the digital output mode is set, no data is output to the pins.

(2) The operation of the D/A converter continues in the HALT and STOP modes. To lower the power consumption, therefore, clear the DACE0 bit to 0, and execute the HALT or STOP instruction after stopping the operation of the D/A converter.

(3) To stop the real-time output mode (including when changing to normal mode), one of the following procedures must be used:

- Wait for at least three clocks after stopping the trigger output source and then set bits DACE0 and DAMD0 to 0.
- After setting bits DACE0 and DAMD0, set the DACEN bit of the PER1 register to 0 (DAC stop).
- When the DACEN bit is set to 0, all the registers in the DAC are cleared, so the settings of the SFRs are required to start the operation again.

(4) When D/A conversion operation is enabled, do not perform A/D conversions from the analog input pin multiplexed with the ANO0 pin.

(5) In the real-time output mode, set the value of the DACS0 register before a timer trigger is generated. Do not change the set value of the DACS0 register while the trigger signal is output.

(6) Since the output impedance of the D/A converter is high, no current can be taken out from the ANO0 pin. If the input impedance of the load is low, insert a follower amplifier between the load and the ANO0 pin before use. In addition, the wiring length between the follower amplifier and the load must be as short as possible due to the high output impedance. If the wiring length is long, take measures such as placing a ground pattern around the wiring area.

(7) When entering the STOP state while the real-time output mode for D/A conversion is enabled, disable linking of ELC events before entering STOP.

# CHAPTER 14 COMPARATOR (RL78/F14 Only)

## 14.1 Overview

The comparator compares a reference voltage to an analog input voltage. The results of a comparison of reference voltage and analog input voltage can be read by software. The comparison result is output externally and an interrupt or ELC event is requested upon detection of a change between the two voltages.

The reference input voltage can be either the input from the IVREF0 pin or the output from the on-chip D/A converter. There are four analog input pins, one of which is to be selected.

Table 14-1 lists the comparator specifications and figure 14-1 shows the Comparator Block Diagram.

**Table 14-1. Comparator Specifications**

| Item | Specification |
|---|---|
| Number of channels | One (comparator 0) |
| Analog input voltage | Input voltage from the IVCMP00 to IVCMP03 pins (one of them to be selected) |
| Reference voltage | • Internal reference voltage (output from on-chip D/A converter)<br>• Input voltage from the external reference voltage input pin (IVREF0) |
| Comparator output | • Comparison result<br>• Generation of ELC event output<br>• Monitor output from register |
| Interrupt request signal | • An interrupt request is generated on detecting a valid edge of comparison result.<br>• Rising edge, falling edge, or both edges can be selected. |
| Digital filter function | • One of three sampling frequencies can be selected.<br>• Not using the filter function can be selected. |

**Figure 14-1. Comparator Block Diagram**

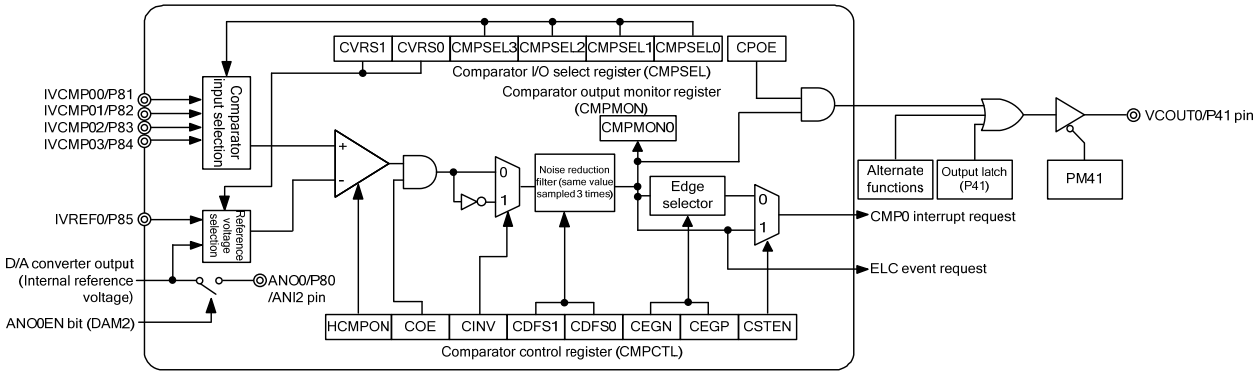



**Table 14-2. Comparator Pin Configuration**

| Pin Name | I/O | Function |
|---|---|---|
| IVCMP00 to IVCMP03 | Input | Analog voltage input pins |
| IVREF0 | Input | External reference voltage input pin |
| VCOUT0 | Output | Comparator output pin |

## 14.2 Registers to Control the Comparator

The comparator is controlled by using the following registers.

**Table 14-3. Registers to Control the Comparator**

| Register Name | Symbol | After Reset | Address | Access Size |
|---|---|---|---|---|
| Peripheral Enable Register 1 | PER1 | 00H | F02C0H | 8 |
| Comparator Control Register | CMPCTL | 00H | F02A0H | 1, 8 |
| Comparator I/O Select Register | CMPSEL | 00H | F02A1H | 1, 8 |
| Comparator Output Monitor Register | CMPMON | 00H | F02A3H | 1, 8 |
| A/D port configuration register | ADPC | 00H | F0076H | 8 |
| D/A converter mode register 2 | DAM2 | 00H | F0227H | 1, 8 |
| Port mode register 4 | PM4 | FFH | FFF24H | 1, 8 |
| Port mode register | PM8 | FFH | FFF28H | 1, 8 |

### 14.2.1 Peripheral Enable Register 1 (PER1)

The PER1 register enables or disables clock supply to each peripheral hardware unit. Clock supply to a hardware unit that is not used is stopped in order to reduce the power consumption and noise.

When the comparator is used, be sure to set bit 5 (CMPEN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 14-2. Format of Peripheral Enable Register 1 (PER1)**

Address: F007AH After reset: 00H

| Symbol | <7> | <6> | <5> | <4> | <3> | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PER1 | DACEN Note | TRGEN | CMPEN Note | TRD0EN | DTCEN | 0 | 0 | TRJ0EN |

| CMPEN Note | Control of comparator input clock | R/W |
|---|---|---|
| 0 | Stops input clock supply.<br>• SFR used by the comparator cannot be written to.<br>• The comparator is in the reset state. | R/W |
| 1 | Supplies input clock.<br>• SFR used by the comparator can be read/written to. | |

**Note** Only for the RL78/F14.

**Cautions 1. When setting the comparator, be sure to set the CMPEN bit to 1 first. If CMPEN = 0, writing to a control register of the comparator is ignored, and all read values are default values (except for port mode registers 4 and 8 (PM4 and PM8), and port registers 4 and 8 (P4 and P8)).**

**2. Be sure to clear the following bits to 0.**

**RL78/F13: bits 1, 2, 5, 6, and 7**

**RL78/F14: bits 1, 2, and 6**

### 14.2.2 Comparator Control Register (CMPCTL)

This register is used to control the comparator operation, enable or disable the comparator output, select the noise filter, select the valid edge of the interrupt signal, and enable/disable release from the STOP mode.

Set the CMPCTL register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

**Figure 14-3. Format of Comparator Control Register (CMPCTL)**

Address: F0341H After reset: 00H

| Symbol | <7> | 6 | 5 | 4 | 3 | 2 | <1> | 0 |
|---|---|---|---|---|---|---|---|---|
| CMPCTL | HCMPON | CDFS1 | CDFS0 | CEGN | CEGP | CSTEN | COE | CINV |

| HCMPON | Comparator operation control Note 1 | R/W |
|---|---|---|
| 0 | Operation stopped (the comparator outputs a low-level signal) | R/W |
| 1 | Operation enabled (input to the comparator pins is enabled) | |

| CDFS1 | CDFS0 | Noise filter selection Notes 2, 3, 4 | R/W |
|---|---|---|---|
| 0 | 0 | Noise filter not used | R/W |
| 0 | 1 | Noise filter sampling frequency is $2^3/f_{CLK}$. | |
| 1 | 0 | Noise filter sampling frequency is $2^4f_{CLK}$. | |
| 1 | 1 | Noise filter sampling frequency is $2^5f_{CLK}$. | |

| CEGN | CEGP | Selection of valid edge of INTCMP interrupt signal | R/W |
|---|---|---|---|
| 0 | 0 | No edge selection | R/W |
| 0 | 1 | Falling edge selection | |
| 1 | 0 | Rising edge selection | |
| 1 | 1 | Both-edge selection | |
| The valid edge is set for the signal after the comparator polarity is selected by using the CINV bit and the filter is selected by using CDFS1 and CDFS0 bits. | | | |

| CSTEN | STOP mode release enable Notes 5, 6 | R/W |
|---|---|---|
| 0 | Releasing STOP mode by comparator interrupt disabled | R/W |
| 1 | Releasing STOP mode by comparator interrupt enabled | |

| COE | Comparator output enable | R/W |
|---|---|---|
| 0 | Comparator output disabled (the output signal is low level) | R/W |
| 1 | Comparator output enabled | |

| CINV | Comparator output polarity selection Notes 2, 3, 6 | R/W |
|---|---|---|
| 0 | Comparator output not inverted | R/W |
| 1 | Comparator output inverted | |

**Notes 1.** Do not modify bits HCMPON and COE simultaneously. The operation stabilization wait time (1 μs when 3.3 V ≤ $V_{DD}$ ≤ 5.5 V or 3 μs when 2.7 V ≤ $V_{DD}$ < 3.3 V) is required after enabling comparator operation (HCMPON = 1).

**2.** Change bits CDFS1, CDFS0, CEGN, CEGP, CSTEN and CINV only after disabling the comparator output (COE = 0).

**3.** Changes to the values of the CDFS1, CDFS0, CEGN, CEGP, CSTEN, and CINV bits may lead to a comparator interrupt request, ELC event request, DTC transfer request, or setting of the INTFLG06 bit in the interrupt source determination flag register 0. Change these bits only after

setting the ELSELR19 register to 00H (no linking of the comparator output 0) and the DTCEN44 bit in the DTCEN4 register to 0 (disabling DTC activation by the comparator detection 0 signal). Also, after changing these bits, initialize the CMPIF0 bit in the interrupt request flag register and the INTFLG06 bit in the interrupt source determination flag register 0 (INTFLG0) to 0 (clearing interrupt request flags).

4. If bits CDFS1 and CDFS0 are changed from 00B (noise filter not used) to a value other than 00B (noise filter used), perform sampling four times and update the filter output, and then use the comparator interrupt request or the ELC event.
5. To enable releasing STOP mode by the comparator interrupt, set this bit to 0 and also set bits CDFS1, CDFS0, and CINV to 00B (noise filter not used).
6. To enable releasing STOP mode by the comparator interrupt and to release from STOP mode by the falling edge of the comparator output, set the CSTEN bit to 1 and CINV bit to 1 (comparator output inverted).

### 14.2.3 Comparator I/O Select Register (CMPSEL)

This register is used to select the comparator input, reference voltage, and to enable or disable the VCOUT0 output.

The CMPSEL register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

**Figure 14-4. Format of Comparator I/O Select Register (CMPSEL)**

Address: F0340H After reset: 00H

| Symbol | 7 | <6> | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CMPSEL | 0 Note 4 | CPOE | CVRS1 | CVRS0 | CMPSEL3 | CMPSEL2 | CMPSEL1 | CMPSEL0 |

| CPOE | VCOUT0 pin output enable | R/W |
|---|---|---|
| 0 | VCOUT0 pin output of the comparator is disabled (the output signal is low level). | R/W |
| 1 | VCOUT0 pin output of the comparator is enabled. | |

| CVRS1 | CVRS0 | Reference voltage selection | R/W |
|---|---|---|---|
| 0 | 0 | No reference voltage | R/W |
| 0 | 1 | External reference voltage (IVREF0) selected | |
| 1 | 0 | Internal reference voltage (D/A converter output) selected Note 1 | |
| 1 | 1 | Setting prohibited Note 2 | |

| CMPSEL3 | CMPSEL2 | CMPSEL1 | CMPSEL0 | Comparator input selection | R/W |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | No input | R/W |
| 0 | 0 | 0 | 1 | IVCMP00 selected | |
| 0 | 0 | 1 | 0 | IVCMP01 selected | |
| 0 | 1 | 0 | 0 | IVCMP02 selected | |
| 1 | 0 | 0 | 0 | IVCMP03 selected | |
| Setting the other values is prohibited. For details, see note 3. | | | | | |

**Notes 1.** When the internal reference voltage is used, set the D/A converter to be used for generating the internal reference voltage before enabling comparator operation (HCMPON = 1). For details on setting the internal reference voltage, see **CHAPTER 13 D/A CONVERTER (RL78/F14 Only)**.

**2.** Modify bits CVRS1 and CVRS0 in the following procedure. Particularly, be sure to set bits CVRS1 and CVRS0 to 00B before changing the set value. Writing a value other than 00B while the value of these bits is not 00B is invalid and the previous value is retained.

1. Set bit COE in CMPCTL register to 0.
2. Set bits CVRS1 and CVRS0 to 00B.
3. Set a new value to bits CVRS1 and CVRS0 (with 1 set in only one of the bits).
4. Wait for the input switching stabilization wait time (300 ns)
5. Set bit COE in CMPCTL register to 1.
6. Clear flag bit CMPIF0 in the control register.

**3.** Modify bits CMPSEL3 to CMPSEL0 in the following procedure. Writing a value other than 0000B while the value of these bits is not 0000B is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.

1. Set bit COE in CMPCTL register to 0.
2. Set bits CMPSEL3 to CMPSEL0 to 0000B.
3. Set a new value to bits CMPSEL3 to CMPSEL0 (with 1 set in only one of the bits).

4. Wait for the input switching stabilization wait time (300 ns)
5. Set bit COE in CMPCTL register to 1.
6. Clear flag bit CMPIF0 in the control register.

**4.** Be sure to set bit 7 to 0.

### 14.2.4 Comparator Output Monitor Register (CMPMON)

This register is used to monitor the comparator output.

The CMPMON register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

**Figure 14-5. Format of Comparator Output Monitor Register (CMPMON)**

Address: F02A2H After reset: 00H

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CMPMON | 0 | 0 | 0 | 0 | 0 | 0 | 0 | CMPMON0 |

| CMPMON0 | Comparator output monitor value | R/W |
|---|---|---|
| 0 | When CINV = 0 (comparator output is not inverted)<br>• Comparator input voltage (IVCMP0n) < reference voltage<br>• Comparator operation disabled (HCMPON = 0)<br>• Comparator output is disabled (COE = 0)<br>When CINV = 1 (converter output is inverted)<br>• Comparator input voltage (IVCMP0n) > reference voltage | R/W |
| 1 | When CINV = 0 (comparator output is not inverted)<br>• Comparator input voltage (IVCMP0n) > reference voltage<br>When CINV = 1 (comparator output is inverted)<br>• Comparator input voltage (IVCMP0n) < reference voltage<br>• Comparator operation disabled (HCMPON = 0)<br>• Comparator output is disabled (COE = 0) | |

**Cautions 1. When comparator operation is enabled (HCMPON = COE = 1) but the noise filter is not in use (CDFS1 and CDFS0 = 00B), write the software so that the CMPMON0 bit is read twice and the values are only used after the two consecutive values match.**

**2. Be sure to set bits 7 to 1 to 0.**

### 14.2.5 A/D port configuration register (ADPC)

This register switches the ANI0/P33 to ANI23/P105 pins to digital I/O of port or analog input.

When the comparator is in use, set the pins selected from among P81/ANI3/IVCMP00, P82/ANI4/IVCMP01, P83/ANI5/IVCMP02, P84/ANI6/IVCMP03, and P85/ANI7/IVREF0 to analog input by using the ADPC register.

The ADPC register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

**Figure 14-6. Format of A/D Port Configuration Register (ADPC)**

Address: F0076H After reset: 00H

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADPC | 0 | 0 | 0 | ADPC4 | ADPC3 | ADPC2 | ADPC1 | ADPC0 |

| ADPC4 | ADPC3 | ADPC2 | ADPC1 | ADPC0 | Analog input (A)/digital I/O (D) switching | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | ANI23/P105 | ANI22/P104 | ANI21/P103 | ANI20/P102 | ANI19/P101 | ANI18/P100 | ANI17/P97 | ANI16/P96 | ANI15/P95 | ANI14/P94 | ANI13/P93 | ANI12/P92 | ANI11/P91 | ANI10/P90 | ANI9/P87 | ANI8/P86 | ANI7/IVREF0/P85 | ANI6/IVCMP03/P84 | ANI5/IVCMP02/P83 | ANI4/IVCMP01/P82 | ANI3/IVCMP00/P81 | ANI2/ANO0/P80 | ANI1/P34 | ANI0/P33 |
| 0 | 0 | 0 | 0 | 0 | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 0 | 0 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D |
| 0 | 0 | 0 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A |
| 0 | 0 | 0 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A |
| 0 | 0 | 1 | 0 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A |
| 0 | 0 | 1 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A |
| 0 | 0 | 1 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A |
| 0 | 0 | 1 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A |
| 0 | 1 | 0 | 0 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A |
| 0 | 1 | 0 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A |
| 0 | 1 | 0 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A |
| 0 | 1 | 0 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 0 | 0 | D | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 0 | 1 | D | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 1 | 0 | D | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A |
| 0 | 1 | 1 | 1 | 1 | D | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 0 | 0 | D | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 0 | 1 | D | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 1 | 0 | D | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 0 | 1 | 1 | D | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 0 | 0 | D | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 0 | 1 | D | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 1 | 0 | D | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 0 | 1 | 1 | 1 | D | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| 1 | 1 | 0 | 0 | 0 | D | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A | A |
| Other than above | | | | | Setting prohibited | | | | | | | | | | | | | | | | | | | | | | |

**Caution Set the pins to be used for the comparator (P81/ANI3/IVCMP00, P82/ANI4/IVCMP01, P83/ANI5/IVCMP02, P84/ANI6/IVCMP03, and P85/ANI7/IVREF0) to the input mode by using port mode registers 8 (PM8).**

### 14.2.6 D/A converter mode register 2 (DAM2)

When the P80/ANO0 pin is in use to output analog signal from the D/A converter, this register is used to control the output from the ANO0 pin. When setting bits 5 and 4 (CVRS1 and CVRS0) in the comparator I/O select register (CMPSEL) to 10B (internal reference voltage (DAC output) is selected), set the ANO0EN bit in this register to 0 (analog output is disabled).

The DAM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

**Figure 14-7. Format of D/A Converter Mode Register (DAM2)**

Address: F0227H After reset: 00H

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| DAM2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ANO0EN |

| ANO0EN | Analog output (ANO0) control | R/W |
|---|---|---|
| 0 | Analog output (ANO0) is disabled. | R/W |
| 1 | Analog output (ANO0) is enabled. | |

### 14.2.7 Port mode register 4 (PM4)

This register is used to set input/output of port 4 in 1-bit units.

When using the port (P41/VCOUT0) to be shared with the comparator output pin, set the corresponding bit in the port mode register 4 (PM4) and port mode register 4 (P4) to 0.

Example)
When P41/VCOUT0 is used for comparator output pin
Set the PM41 bit in the port mode register 4 to 0.
Set the P41 bit in the port register 4 to 0.

The PM4 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

**Figure 14-8. Format of Port Mode Register 4 (PM4) (100-Pin Products)**

Address: FFF24H After reset: FFH

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM4 | PM47 | PM46 | PM45 | PM44 | PM43 | PM42 | PM41 | PM40 |

| PM4n | P4n pin input/output mode selection (n = 0 to 7) | R/W |
|---|---|---|
| 0 | 0: Output mode (output buffer on) | R/W |
| 1 | 1: Input mode (output buffer off) | |

### 14.2.8 Port mode register (PM8)

When using the P81/ANI3/IVCMP00, P82/ANI4/IVCMP01, P83/ANI5/IVCMP02, P84/ANI6/IVCMP03, or P85/ANI7/IVREF0 pin for an analog input port of the comparator, set the PM81, PM82, PM84, or PM85 bit to 1 corresponding to the port to be used.

If the PM81 to PM85 bits are set to 0, they cannot be used as analog input port pins.

The PM8 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

**Caution If a pin is set as an analog input port, not the pin level but "0" is always read.**

**Figure 14-9. Format of Port Mode Register 8 (PM8)**

Address: FFF28H After reset: FFH

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM8 | PM87 | PM86 | PM85 | PM84 | PM83 | PM82 | PM81 | PM80 |

| PM8n | P8n pin input/output mode selection (n = 0 to 7) | R/W |
|---|---|---|
| 0 | 0: Output mode (output buffer on) | R/W |
| 1 | 1: Input mode (output buffer off) | |

The P81/ANI3/IVCMP00, P82/ANI4/IVCMP01, P83/ANI5/IVCMP02, P84/ANI6/IVCMP03, and P85/ANI7/IVREF0 pins are as shown below depending on the settings of the A/D port configuration register (ADPC), analog input channel specification register (ADS), and PM8 registers.

**Table 14-4. Setting Functions of P81/ANI3/IVCMP00 to P85/ANI7/IVREF0 Pins**

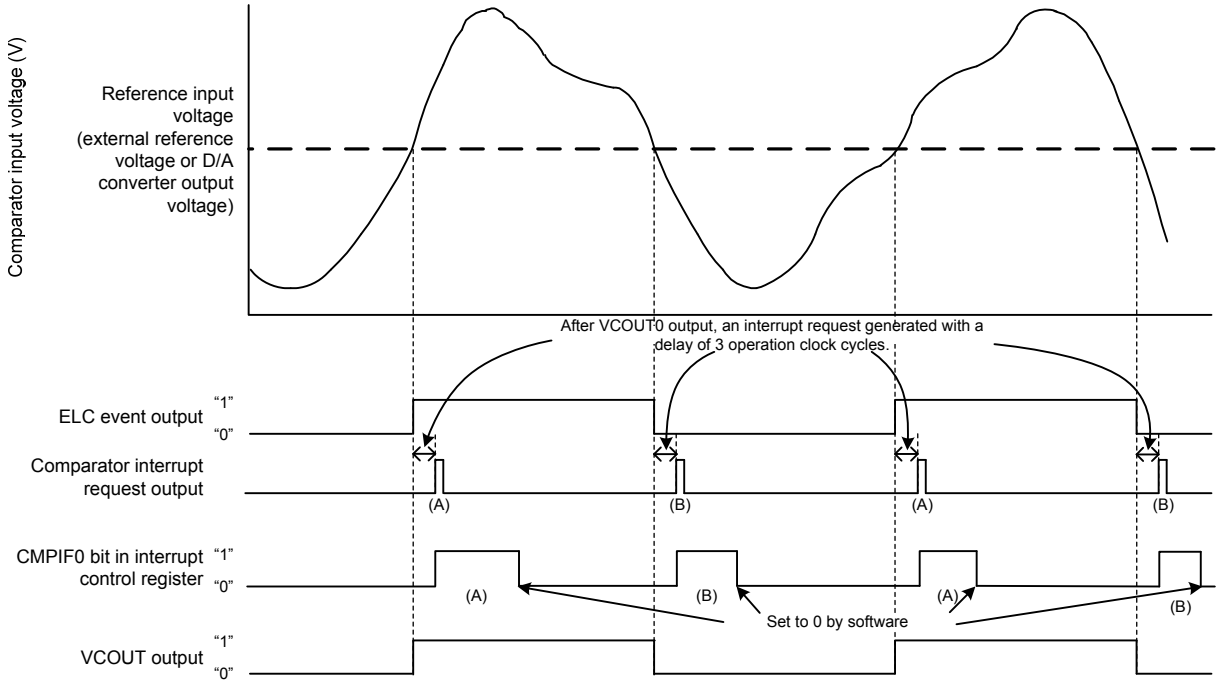
| ADPC Register | PM8 Register | ADS Register | P81/ANI3/IVCMP00 to P85/ANI7/IVREF0 |
|---|---|---|---|
| Digital I/O selection | Input mode | – | Digital input |
| | Output mode | – | Digital output |
| Analog I/O selection | Input mode | Selects ANI. | Analog input (to be converted) |
| | | Does not select ANI. | Analog input (not to be converted) |
| | Output mode | Selects ANI. | Setting prohibited |
| | | Does not select ANI. | |

Reset signal generation sets all the P81/ANI3/IVCMP00 to P85/ANI7/IVREF0 pins to analog input.

## 14.3 Operation

Figure 14-10 shows a comparator operation example. The VCOUT0 output becomes 1 when the analog input voltage is higher than the comparator input voltage voltage, and the VCOUT0 output becomes 0 when the analog input voltage is lower than the reference voltage. When the comparator output changes, an interrupt request and an ELC event are output.

**Figure 14-10. Comparator Operation Example**




**Caution The above diagram applies when CPOE = 1 (pin output enabled), CDFS1 and CDFS0 = 00B (filter not used), and CEGP = CEGN = 1 (both-edge selection). When CINV = 0, CEGP = 1, and CEGN = 0 (rising-edge selection for non-inversion output signal from the comparator), CMPIF0 changes as shown by (A) only. When CINV = 0, CEGP = 0, and CEGN = 1 (falling-edge selection for non-inversion output signal from the comparator), CMPIF0 changes as shown by (B) only. When CPOE = 1, VCOUT0 directly outputs the ELC event output.**

### 14.3.1 Noise Filter

The comparator contains a noise filter. The sampling clock can be selected by bits CDFS1 and CDFS0 in the CMPCTL register.

The comparator signal is sampled every sampling clock and if the same value is sampled three times, the noise filter output at the next sampling clock cycle is used as the comparator output.

Figure 14-11 shows the configuration of the noise filter and edge detector and figure 14-12 shows an example of noise filter and interrupt operation.

**Figure 14-11. Noise Filter and Edge Detection Configuration**

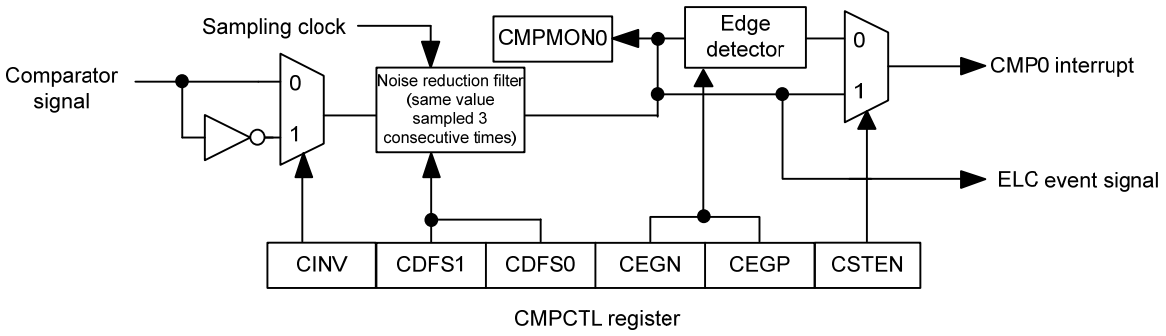
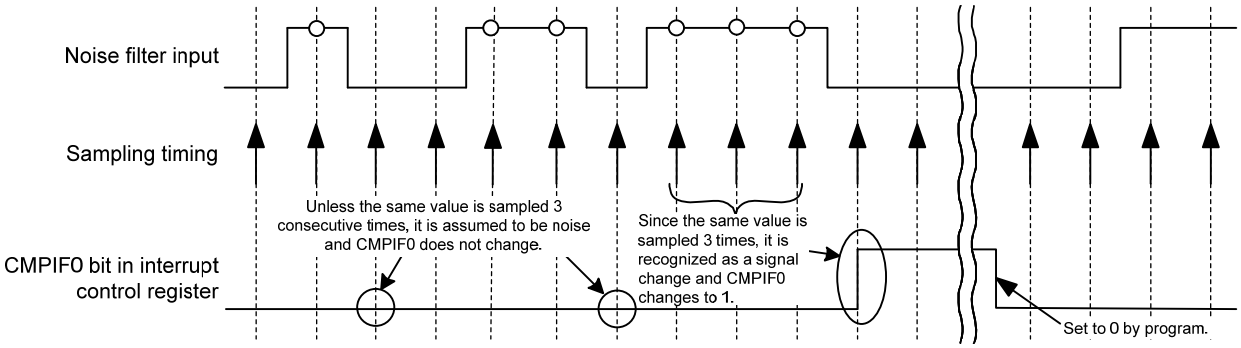



**Figure 14-12. Noise Filter and Interrupt Operation Example**




**Caution The above operation example applies when bits CDFS1 and CDFS0 in the CMPCTL register is 01B, 10B, or 11B (noise filter used).**

### 14.3.2 Comparator Interrupts

The comparator generates an interrupt request. The comparator interrupt functions provide priority specification flag, interrupt mask flag, interrupt request flag, and interrupt vector.

When using the comparator interrupt, set at least one of bits CEGP and CEGN in the CMPCTL register to 1 (to a value other than 00B (no edge selection)).

For details on the register setting related to comparator interrupt request, refer to **14.2.2 Comparator Control Register (CMPCTL)**.

To use the comparator interrupt in STOP mode, set the CSTEN bit in CMPCTL register to 1 (releasing STOP mode by comparator interrupt enabled) and set the CDFS1 and CDFS0 bits to 00 (digital noise filter not used).

### 14.3.3 Comparator ELC Event Output

An ELT event is generated in accord to settings of the comparator output inversion control (CINV bit) and noise filter output (CDFS1 and CDFS0 bits) in the CMPCTL register. Use the ELSELR19 register of the ELC for selection of event output destination and disabling the event link operation.

### 14.3.4 Comparator Pin Output

The comparison result from the comparator can be output to external pins. Bits CINV and CPOE in the CMPSEL register can be used to set the output polarity (output is inverted or not) and to enable or disable the output. For the correspondence between the register setting and the comparator pin output, refer to **14.2.2 Comparator Control Register (CMPCTL)**.

### 14.3.5 Stopping or Supplying Comparator Clock

To stop the comparator by setting peripheral enable register 1 (PER1), use the following procedure:

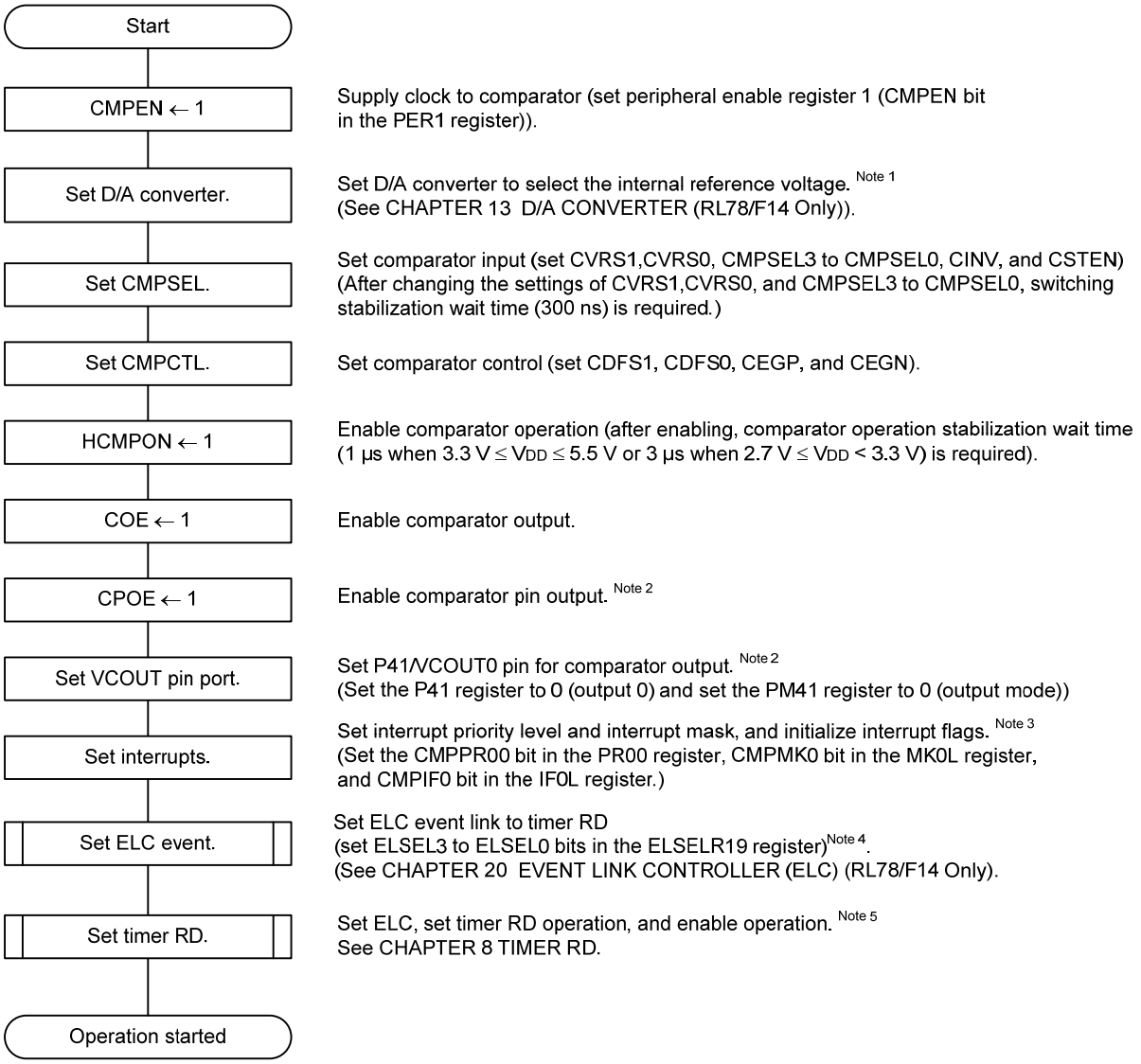
<1> Set the HCMPON bit in the CMPCTL register to 0 (stop the comparator input).
<2> Set the CMPEN bit in the PER1 register to 0.
<3> Set the interrupt flag (CMPIF0 bit in the IF0L register) to 0 (clear any unnecessary interrupt before stopping the comparator).

When the clock is stopped by setting PER1, all the internal registers in the comparator are initialized. To use the comparator again, follow the procedure in table 14-13 to set the registers.

### 14.3.6 Comparator Setting Flowchart

Figure 14-13 shows the flowchart for setting the comparator-related registers.

**Figure 14-13. Comparator Operation Setting Flowchart (when Using the timer RD Operation Triggered by Internal Reference Voltage (D/A Converter Output), INTCMP0 Interrupt, or ELC Event)**




**Notes 1.** This is not required when the external reference voltage is used.

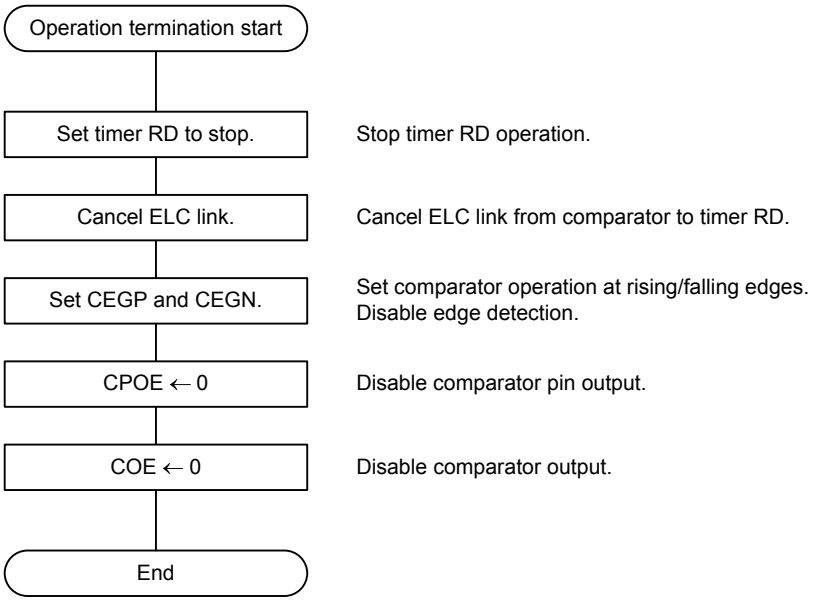
**2.** This is not required when the comparator output is not output to the external pin.

**3.** Set the registers assigned to interrupt control.

**4.** This is not required when the ELC event is not used.

**5.** This is not required when the timer RD by the ELC event is not used.

**Figure 14-14. Comparator Operation Termination Flowchart (when Using the Timer RD Operation by the ELC Event)**



# CHAPTER 15 SERIAL ARRAY UNIT

Serial array unit has two serial channels. Each channel can achieve 3-wire serial (CSI), UART, and simplified $I^2C$ communication.

Function assignment of each channel supported by the RL78/F13 and RL78/F14 is as shown below.

- Group A products

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note 2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note 2 | | IIC01 |

- Products of Groups C-1 and D-1

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note 2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01(supporting SPI function) Note 2 | | IIC01 |
| 1 | 0 | CSI10 (supporting SPI function)Note 1, 2 | UART1 | IIC10 |
| | 1 | - | | - |

- Products of Groups B, C-2, D-2, and E

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note 2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01(supporting SPI function) Note 2 | | IIC01 |
| 1 | 0 | CSI10(supporting SPI function)Note 1, 2 | UART1 | IIC10 |
| | 1 | CSI11 (supporting SPI function) Note 2 | | IIC11 |

**Notes 1.** 48-pin, 32-pin and 30-pin products do not have $\overline{SSI10}$ pin.

**2.** Set CKPmn bit of SCRmn register to 1, when SSEmn = 1 (Enables $\overline{SSImn}$ pin input).
(m = 0, 1, n = 0, 1)

When “UART0” is used for channels 0 and 1 of the unit 0, CSI00 and CSI01 cannot be used.

**Caution Most of the following descriptions in this chapter use the units and channels of the 80-pin products of RL78/F13 (CAN and LIN incorporated) as an example.**

**Remark** Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory
Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory
Group C-1: RL78/F13 (CAN and LIN incorporated) products with 30 or 32 pins
Group C-2: RL78/F13 (CAN and LIN incorporated) products with 48, 64, or 80 pins
Group D-1: RL78/F14 products with 30 or 32 pins
Group D-2: RL78/F14 products with 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory
Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

## 15.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/F13 and RL78/F14 has the following features.

### 15.1.1 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11)

Data is transmitted or received in synchronization with the serial clock ($\overline{SCK}$) output from the master channel.

3-wire serial communication is clocked communication performed by using three communication lines: one for the serial clock ($\overline{SCK}$), one for transmitting serial data (SO), one for receiving serial data (SI).

For details about the settings, see **15.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) Communication.**

[Data transmission/reception]

- Data length of 7 to 16 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- Level setting of transmit/receive data

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate

  During master communication: Max. $f_{MCK}/4$ **Note**

  During slave communication: Max. $f_{MCK}/6$ **Note**

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

CSI00, CSI01, CSI10, and CSI11 support the SPI function.

[Extended function]

- Slave select function of the SPI function

**Note** Use the clocks within a range satisfying the $\overline{SCK}$ cycle time ($t_{KCY}$) characteristics (see **CHAPTER 34** to **CHAPTER 36 ELECTRICAL SPECIFICATIONS**).

### 15.1.2 UART (UART0, UART1)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can be implemented by using timer array unit with an external interrupt (INTP0).

For details about the settings, see **15.7 Operation of UART (UART0, UART1) Communication.**

[Data transmission/reception]

- Data length of 7, 8, 9, 16 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Framing error, parity error, or overrun error

The LIN-bus is accepted in UART0 (0 and 1 channels of unit 0).

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit

### 15.1.3 Simplified I²C (IIC00, IIC01, IIC10, IIC11)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

For details about the settings, see **15.9 Operation of Simplified I²C (IIC00, IIC01, IIC10, IIC11) Communication.**

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function**Note** and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- ACK error, or overrun error

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

**Note** When receiving the last data, ACK will not be output if 0 is written to the SOEmn bit (serial output enable register m (SOEm)) and serial communication data output is stopped. See **15.9.3 (2) Processing flow** for details.

**Remarks 1.** To use an I²C bus of full function, see **CHAPTER 16 SERIAL INTERFACE IICA**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

## 15.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

**Table 15-1. Configuration of Serial Array Unit**

| Item | Configuration |
|---|---|
| Shift register | 16 bits |
| Buffer register | Serial data register mn (SDRmn)[Note] |
| Serial clock I/O | $\overline{SCK00}$, $\overline{SCK01}$, $\overline{SCK10}$, $\overline{SCK11}$ pins (for 3-wire serial I/O), SCL00, SCL01, SCL10, SCL11 pins (for simplified $I^2C$) |
| Serial data input | SI00, SI01, SI10, SI11 pins (for 3-wire serial I/O), RxD0 pin (for UART supporting LIN-bus), RxD1 pin (for UART) |
| Serial data output | SO00, SO01, SO10, SO11 pins (for 3-wire serial I/O), TxD0 pin (for UART supporting LIN-bus), TxD1 pin (for UART), output controller |
| Serial data I/O | SDA00, SDA01, SDA10, SDA11 pins (for simplified $I^2C$) |
| Slave select input | $\overline{SSI00}$, $\overline{SSI01}$, $\overline{SSI10}$, $\overline{SSI11}$ pin (for 3-wire serial I/O) |
| Control registers | <Registers of unit setting block><br>• Peripheral enable register 0 (PER0)<br>• Serial clock select register m (SPSm)<br>• Serial channel enable status register m (SEm)<br>• Serial channel start register m (SSm)<br>• Serial channel stop register m (STm)<br>• Serial output enable register m (SOEm)<br>• Serial output register m (SOm)<br>• Serial output level register m (SOLm)<br>• Serial slave select enable register m (SSEm)<br>• Input switch control register (ISC)<br>• Noise filter enable register 0 (NFEN0) |
| | <Registers of each channel><br>• Serial data register mn (SDRmn)<br>• Serial mode register mn (SMRmn)<br>• Serial communication operation setting register mn (SCRmn)<br>• Serial status register mn (SSRmn)<br>• Serial flag clear trigger register mn (SIRmn) |
| | • Port input mode registers 1, 3, 5 to 7, 12 (PIM1, PIM3, PIM5 to PIM7, PIM12)<br>• Port output mode registers 1, 6, 7, 12 (POM1, POM6, POM7, POM12)<br>• Port mode registers 1, 3 to 7, 12 (PM1, PM3 to PM7, PM12)<br>• Port registers 1, 3 to 7, 12 (P1, P3 to P7, P12) |

(**Notes** and **Remark** are listed on the next page.)

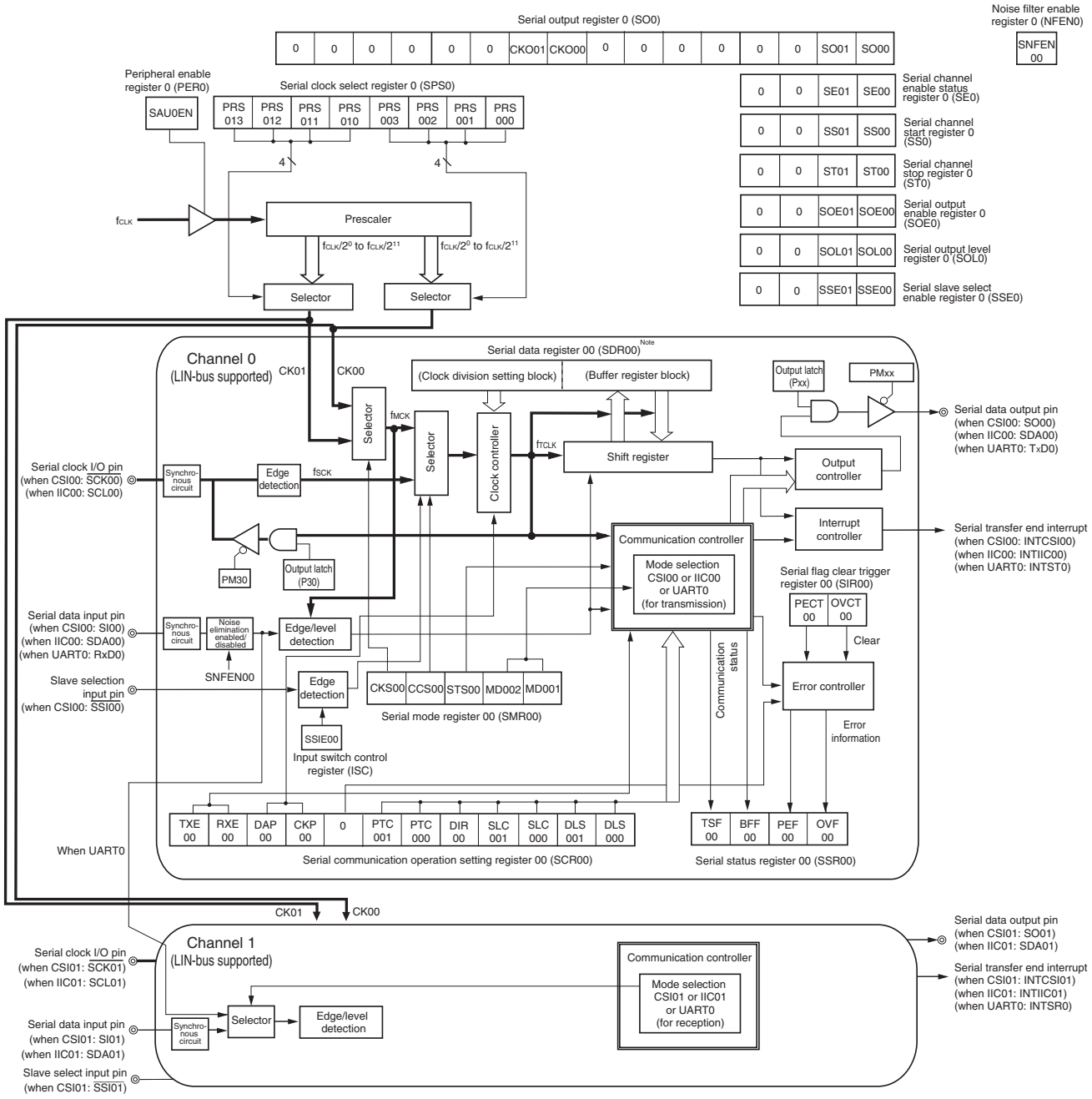
**Note** When SEmn is 1, the lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

- CSIp communication … $SDR_pL$ (CSIp data register)
- UARTq reception … $SDR_{mm}L$ (UARTq receive data register)
- UARTq transmission … SDRmnL (UARTq transmit data register)
- IICr communication … SDRrL (IICr data register)

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), q: UART number (q = 0, 1), r: IIC number (r = 00, 01, 10, 11)

Figure 15-1 shows the block diagram of the serial array unit 0.

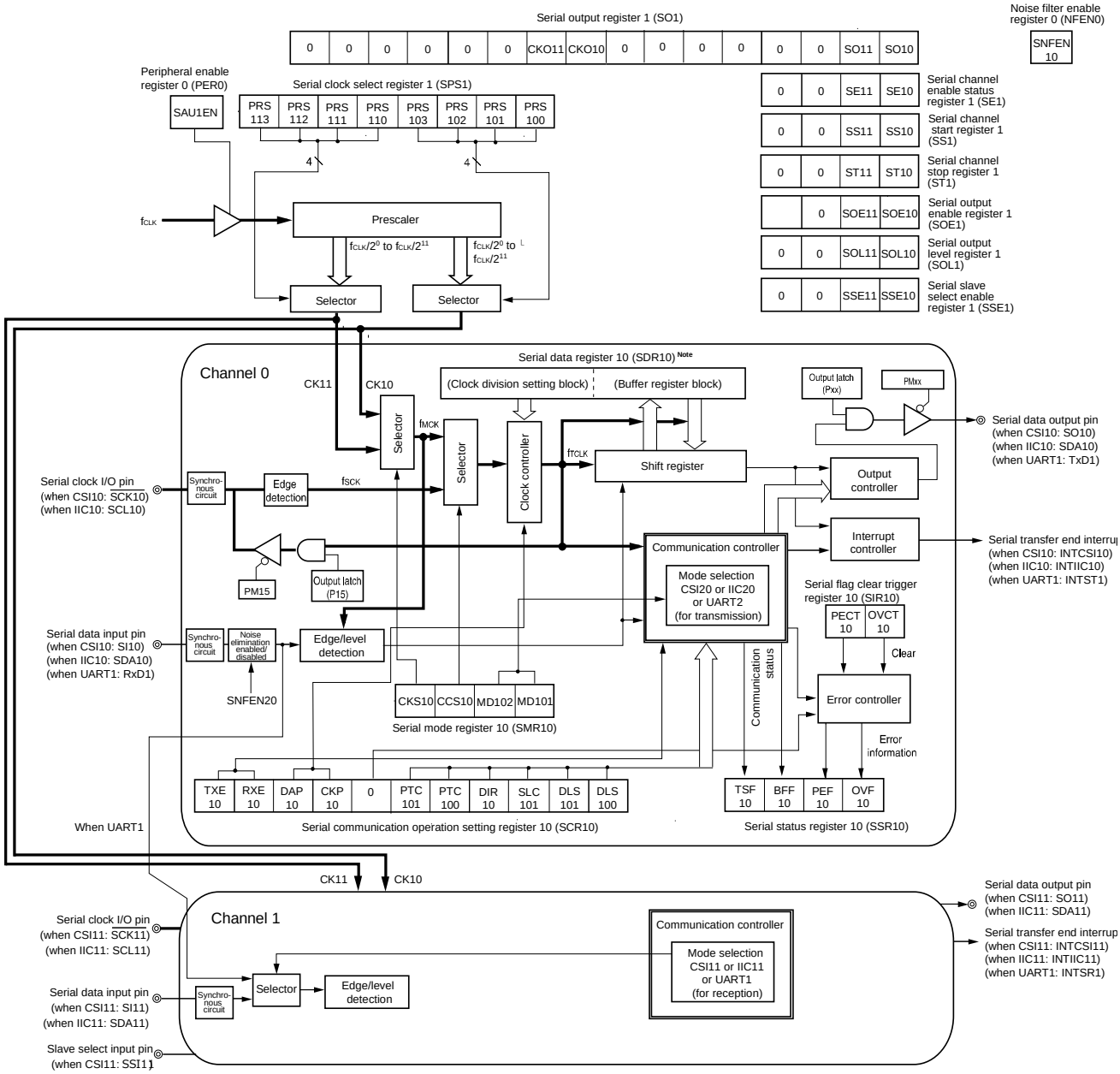
**Figure 15-1. Block Diagram of Serial Array Unit 0**




**Caution: If operation is stopped (SEmn = 0), the upper 7 bits set the clock division, and the lower bits have no meaning.**
**If operation is in progress (SEmn = 1), the serial data register 10 functions as the buffer register.**

Figure 15-2 shows the block diagram of the serial array unit 1.

**Figure 15-2. Block Diagram of Serial Array Unit 1**




**Caution: If operation is stopped (SEmn = 0), the upper 7 bits set the clock division, and the lower bits have no meaning.**

**If operation is in progress (SEmn = 1), the serial data register 10 functions as the buffer register.**

**(1) Shift register**

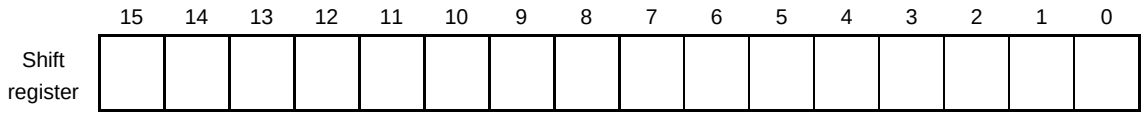
This is a 16-bit register that converts parallel data into serial data or vice versa.

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the serial data register mn (SDRmn) when operation is in progress (SEmn = 1).




**(2) Serial data register mn (SDRmn)**

The SDRmn register is the transmit/receive data register (16 bits) of channel n. If operation is stopped (SEmn = 0), bits 15 to 9 are used as a register that sets the division ratio of the operation clock ($f_{MCK}$). If operation is in progress (SEmn = 1), the SDRmn register functions as a transmit/receive buffer register.

When data is received, parallel data converted by the shift register is stored. When data is to be transmitted, set transmit data to be transferred to the shift register .

The data to be stored is as follows, depending on the setting of bits 4 to 0 (DLSmn4 to DLSmn0) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of SDRmn register)
- 9-bit data length (stored in bits 0 to 8 of SDRmn register)

:

- 16-bit data length (stored in bits 0 to 15 of SDRmn register)

The SDRmn register can be read or written in 16-bit units.

The lower 8 bits of the SDRmn register can be read or written as the following SFR when operation is in progress (SEmn = 1). The following SDRmnL registers are available, depending on the communication mode.

- CSIp communication … SDRpL
- UARTq reception … SDRmnL
- UARTq transmission … SDRmnL
- IICr communication … SDRrL

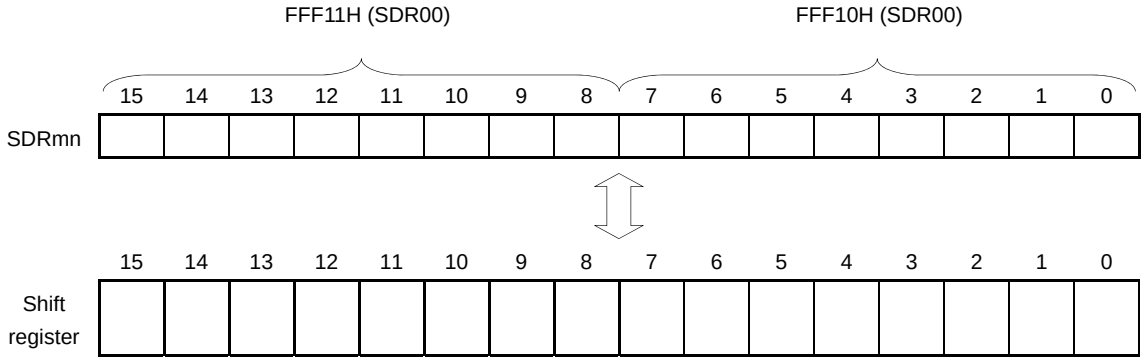
Reset signal generation clears the SDRmn register to 0000H.

**Note** Writing in 8-bit units is prohibited when the operation is stopped (SEmn = 0).

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), q: UART number (q = 0, 1), r: IIC number (r = 00, 01, 10, 11)

**Figure 15-3. Format of Serial Data Register mn (SDRmn) (mn = 00, 01, 10, 11)**

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)




**Remark** For the function of the higher 7 bits of the SDRmn register, see **15.3 Registers Controlling Serial Array Unit**.

## 15.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm)
- Serial slave select enable register m (SSEm)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 1, 3, 5 to 7, 12 (PIM1, PIM3, PIM5 to PIM7, PIM12)
- Port output mode registers 1, 6, 7, 12 (POM1, POM6, POM7, POM12)
- Port mode registers 1, 3, 5 to 7, 12 (PM1, PM3, PM5 to PM7, PM12)
- Port registers 1, 3, 5 to 7, 12 (P1, P3, P5 to P7, P12)
- Port mode control register 12 (PMC12)
- Port input threshold control registers 1, 3, 5 to 7, and 12 (PITHL1, PITHL3, PITHL5 to PITHL7, PITHL12)
- Port output slew rate select register (PSRSEL)

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

### 15.3.1 Peripheral enable register 0 (PER0)

PER0 is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.

When serial array unit 1 is used, be sure to set bit 3 (SAU1EN) of this register to 1.

Set the PER0 register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the PER0 register to 00H.

**Figure 15-4. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

| Symbol | <7> | 6 | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PER0 | RTCEN | 0 | ADCEN | IICA0EN | SAU1EN | SAU0EN | TAU1EN | TAU0EN |

| SAUmEN | Control of serial array unit m input clock supply |
|---|---|
| 0 | Stops supply of input clock.<br>• SFR used by serial array unit m cannot be written.<br>• Serial array unit m is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by serial array unit m can be read/written. |

**Cautions 1. When setting serial array unit m, be sure to set the SAUmEN bit to 1 first. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read (except for the input switch control register (ISC), noise filter enable register 0 (NFEN0), port input mode registers 1, 3, 5 to 7, 12 (PIM1, PIM3, PIM5 to PIM7, PIM12), port output mode registers 1, 6, 7, 12 (POM1, POM6, POM7, POM12), port mode registers 1, 3, 5 to 7, 12 (PM1, PM3, PM5 to PM7, PM12), port registers 1, 3, 5 to 7, 12 (P0, P1, P3, P5 to P7, P12), port mode control register 12 (PMC12), port input threshold control registers 1, 3, 5 to 7, and 12 (PITHL1, PITHL3, PITHL5 to PITHL7, and PITHL12), and port output slew rate select register (PSRSEL)).**

**2. Be sure to clear the following bits to 0.**
**Bits 1, 3, 4, and 6 in the RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory**
**Bits 4 and 6 in 30-pin products of the RL78/F13 (CAN and LIN incorporated) and in 30-pin products of the RL78/F14**
**Bit 6 in the products other than above**

### 15.3.2 Serial clock select register m (SPSm)

The SPSm register is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of the SPSm register, and CKm0 is selected by bits 3 to 0.

Rewriting the SPSm register is prohibited when the register is in operation (when SEmn = 1).

The SPSm register can be set by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the SPSm register with an 8-bit memory manipulation instruction with SPSmL.

Reset signal generation clears the SPSm register to 0000H.

**Figure 15-5. Format of Serial Clock Select Register m (SPSm)**

Address: F0116H, F0117H (SPS0), F0156H, F0157H (SPS1) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PRS m13 | PRS m12 | PRS m11 | PRS m10 | PRS m03 | PRS m02 | PRS m01 | PRS m00 |

| PRS mk3 | PRS mk2 | PRS mk1 | PRS mk0 | Section of operation clock (CKmk) Note | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | $f_{CLK}$ = 2 MHz | $f_{CLK}$ = 5 MHz | $f_{CLK}$ = 10 MHz | $f_{CLK}$ = 20 MHz | $f_{CLK}$ = 32 MHz |
| 0 | 0 | 0 | 0 | $f_{CLK}$ | 2 MHz | 5 MHz | 10 MHz | 20 MHz | 32 MHz |
| 0 | 0 | 0 | 1 | $f_{CLK}/2$ | 1 MHz | 2.5 MHz | 5 MHz | 10 MHz | 16 MHz |
| 0 | 0 | 1 | 0 | $f_{CLK}/2^2$ | 500 kHz | 1.25 MHz | 2.5 MHz | 5 MHz | 8 MHz |
| 0 | 0 | 1 | 1 | $f_{CLK}/2^3$ | 250 kHz | 625 kHz | 1.25 MHz | 2.5 MHz | 4 MHz |
| 0 | 1 | 0 | 0 | $f_{CLK}/2^4$ | 125 kHz | 313 kHz | 625 kHz | 1.25 MHz | 2 MHz |
| 0 | 1 | 0 | 1 | $f_{CLK}/2^5$ | 62.5 kHz | 156 kHz | 313 kHz | 625 kHz | 1 MHz |
| 0 | 1 | 1 | 0 | $f_{CLK}/2^6$ | 31.3 kHz | 78.1 kHz | 156 kHz | 313 kHz | 500 kHz |
| 0 | 1 | 1 | 1 | $f_{CLK}/2^7$ | 15.6 kHz | 39.1 kHz | 78.1 kHz | 156 kHz | 250 kHz |
| 1 | 0 | 0 | 0 | $f_{CLK}/2^8$ | 7.81 kHz | 19.5 kHz | 39.1 kHz | 78.1 kHz | 125 kHz |
| 1 | 0 | 0 | 1 | $f_{CLK}/2^9$ | 3.91 kHz | 9.77 kHz | 19.5 kHz | 39.1 kHz | 62.5 kHz |
| 1 | 0 | 1 | 0 | $f_{CLK}/2^{10}$ | 1.95 kHz | 4.88 kHz | 9.77 kHz | 19.5 kHz | 31.3 kHz |
| 1 | 0 | 1 | 1 | $f_{CLK}/2^{11}$ | 977 Hz | 2.44 kHz | 4.88 kHz | 9.77 kHz | 15.6 kHz |
| Other than above | | | | Setting prohibited | | | | | |

**Note** When changing the clock selected for $f_{CLK}$ (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 0003H) the operation of the serial array units (SAUs).

**Caution Be sure to clear bits 15 to 8 to 0.**

**Remarks 1.** $f_{CLK}$: CPU/peripheral hardware clock frequency
$f_{SUB}$: Subsystem clock frequency

**2.** m: Unit number (m = 0, 1), k = 0, 1

### 15.3.3 Serial mode register mn (SMRmn)

The SMRmn register is a register that sets an operation mode of channel n. It is also used to select an operation clock ($f_{MCK}$), specify whether the serial clock ($f_{SCK}$) may be input or not, set a start trigger, an operation mode (CSI, UART, or simplified $I^2C$), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMRmn register is prohibited when the register is in operation (when SEmn = 1). However, the MDmn0 bit can be rewritten during operation.

Set the SMRmn register by a 16-bit memory manipulation instruction.

Reset signal generation sets the SMRmn register to 0020H.

**Figure 15-6. Format of Serial Mode Register mn (SMRmn) (1/2)**

Address: F0108H, F0109H (SMR00), F010AH, F010BH (SMR01), F0148H, F0149H (SMR10), F014AH, F014BH (SMR11) After reset: 0020H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SMRmn | CKS mn | CCS mn | 0 | 0 | 0 | 0 | 0 | STS mn | 0 | SIS mn0 | 1 | 0 | 0 | MD mn2 | MD mn1 | MD mn0 |

| CKS mn | Selection of operation clock ($f_{MCK}$) of channel n |
|---|---|
| 0 | Operation clock CKm0 set by the SPSm register |
| 1 | Operation clock CKm1 set by the SPSm register |
| Operation clock ($f_{MCK}$) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock ($f_{TCLK}$) is generated. | |

| CCS mn | Selection of transfer clock ($f_{TCLK}$) of channel n |
|---|---|
| 0 | Divided operation clock $f_{MCK}$ specified by the CKSmn bit |
| 1 | Clock input $f_{SCK}$ from the $\overline{SCKp}$ pin (slave transfer in CSI mode) |
| Transfer clock $f_{TCLK}$ is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operation clock ($f_{MCK}$) is set by the higher 7 bits of the SDRmn register. | |

| STS mn | Selection of start trigger source |
|---|---|
| 0 | Only software trigger is valid (selected for CSI, UART transmission, and simplified $I^2C$). |
| 1 | Valid edge of the RxDq pin (selected for UART reception) |
| Transfer is started when the above source is satisfied after 1 is set to the SSm register. | |

**Caution Be sure to clear bits 13 to 9, 7, 4, and 3 to 0. Be sure to set bit 5 to 1.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), q: UART number (q = 0, 1), r: IIC number (r = 00, 01, 10, 11)

**Figure 15-6. Format of Serial Mode Register mn (SMRmn) (2/2)**

Address: F0108H, F0109H (SMR00), F010AH, F010BH (SMR01), F0148H, F0149H (SMR10), F014AH, F014BH (SMR11) After reset: 0020H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SMRmn | CKS mn | CCS mn | 0 | 0 | 0 | 0 | 0 | STS mn | 0 | SIS mn0 | 1 | 0 | 0 | MD mn2 | MD mn1 | MD mn0 |

| SIS mn0 | Controls inversion of level of receive data of channel n in UART mode |
|---|---|
| 0 | Falling edge is detected as the start bit.<br>The input communication data is captured as is. |
| 1 | Rising edge is detected as the start bit.<br>The input communication data is inverted and captured. |

| MD mn2 | MD mn1 | Setting of operation mode of channel n |
|---|---|---|
| 0 | 0 | CSI mode |
| 0 | 1 | UART mode |
| 1 | 0 | Simplified $I^2C$ mode |
| 1 | 1 | Setting prohibited |

| MD mn0 | Selection of interrupt source of channel n |
|---|---|
| 0 | Transfer end interrupt |
| 1 | Buffer empty interrupt<br>(Occurs when data is transferred from the SDRmn register to the shift register.) |
| For successive transmission, the next transmit data is written by setting the MDmn0 bit to 1 when SDRmn data has run out. | |

**Caution Be sure to clear bits 13 to 9, 7, 4, and 3 to 0. Be sure to set bit 5 to 1.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), q: UART number (q = 0, 1), r: IIC number (r = 00, 01, 10, 11)

### 15.3.4 Serial communication operation setting register mn (SCRmn)

The SCRmn register is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCRmn register is prohibited when the register is in operation (when SEmn = 1).

Set the SCRmn register by a 16-bit memory manipulation instruction.

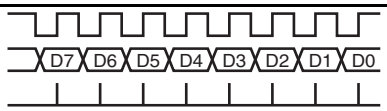
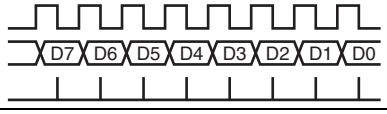
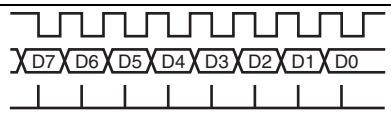
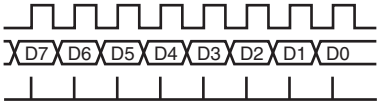
Reset signal generation sets the SCRmn register to 0087H.

**Figure 15-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/3)**

Address: F010CH, F010DH (SCR00), F010EH, F010FH (SCR01), F014CH, F014DH (SCR10), F014EH, F014FH (SCR11) After reset: 0087H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SCRmn | TXE mn | RXE mn | DAP mn | CKP mn | 0 | 0 | PTC mn1 | PTC mn0 | DIR mn | 0 | SLC mn1 | SLC mn0 | DLS mn3 | DLS mn2 | DLS mn1 | DLS mn0 |

| TXE mn | RXE mn | Setting of operation mode of channel n |
|---|---|---|
| 0 | 0 | Disable communication. |
| 0 | 1 | Reception only |
| 1 | 0 | Transmission only |
| 1 | 1 | Transmission/reception |

| DAP mn | CKP mn | Selection of data and clock phase in CSI mode | Type |
|---|---|---|---|
| 0 | 0 |  | 1 |
| 0 | 1 |  | 2 |
| 1 | 0 |  | 3 |
| 1 | 1 |  | 4 |
| Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified $I^2C$ mode.<br>Set CKPmn to 1, when SSEmn = 1 (Enables $\overline{SSImn}$ pin input). | | | |

**Caution Be sure to clear bits 6, 10, and 11 to 0.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

**Figure 15-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/3)**

Address: F010CH, F0110DH (SCR00), F010EH, F010FH (SCR01), F014CH, F014DH (SCR10), F014EH, F014FH (SCR11) After reset: 0087H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SCRmn | TXE mn | RXE mn | DAP mn | CKP mn | 0 | 0 | PTC mn1 | PTC mn0 | DIR mn | 0 | SLC mn1 | SLC mn0 | DLS mn3 | DLS mn2 | DLS mn1 | DLS mn0 |

| PTC mn1 | PTC mn0 | Setting of parity bit in UART mode | |
|---|---|---|---|
| | | Transmission | Reception |
| 0 | 0 | Does not output the parity bit. | Receives without parity |
| 0 | 1 | Outputs 0 parity Note. | No parity judgment |
| 1 | 0 | Outputs even parity. | Judged as even parity. |
| 1 | 1 | Outputs odd parity. | Judges as odd parity. |
| Be sure to set PTCmn1, PTCmn0 = 0, 0 in the CSI mode and simplified $I^2C$ mode. | | | |

| DIR mn | Selection of data transfer sequence in CSI and UART modes |
|---|---|
| 0 | Inputs/outputs data with MSB first. |
| 1 | Inputs/outputs data with LSB first. |
| Be sure to clear DIRmn = 0 in the simplified $I^2C$ mode. | |

| SLCmn1 | SLC mn0 | Setting of stop bit in UART mode |
|---|---|---|
| 0 | 0 | No stop bit |
| 0 | 1 | Stop bit length = 1 bit |
| 1 | 0 | Stop bit length = 2 bits (mn = 00, 10 only) |
| 1 | 1 | Setting prohibited |
| When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred.<br>Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified $I^2C$ mode.<br>Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the CSI mode. | | |

**Note** "0" is always added regardless of the contents of data.

**Caution Be sure to clear bits 6, 10, and 11 to 0.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

**Figure 15-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (3/3)**

Address: F010CH, F0110DH (SCR00), F010EH, F010FH (SCR01), After reset: 0087H R/W
F014CH, F014DH (SCR10), F014EH, F014FH (SCR11)

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SCRmn | TXE mn | RXE mn | DAP mn | CKP mn | 0 | 0 | PTC mn1 | PTC mn0 | DIR mn | 0 | SLC mn1 | SLC mn0 | DLS mn3 | DLS mn2 | DLS mn1 | DLS mn0 |

| DLS mn3 | DLS mn2 | DLS mn1 | DLS mn0 | Setting of data length in CSI, UART mode | Serial function | | |
|---|---|---|---|---|---|---|---|
| | | | | | CSI | UART | IIC |
| 0 | 1 | 1 | 0 | 7 bits (stored in bits 0 to 6 of SDRmn register) | √ | √ | - |
| 0 | 1 | 1 | 1 | 8 bits (stored in bits 0 to 7 of SDRmn register) | √ | √ | √ |
| 1 | 0 | 0 | 0 | 9 bits (stored in bits 0 to 8 of SDRmn register) | √ | √ | - |
| 1 | 0 | 0 | 1 | 10 bits (stored in bits 0 to 9 of SDRmn register) | √ | - | - |
| 1 | 0 | 1 | 0 | 11 bits (stored in bits 0 to 10 of SDRmn register) | √ | - | - |
| 1 | 0 | 1 | 1 | 12 bits (stored in bits 0 to 11 of SDRmn register) | √ | - | - |
| 1 | 1 | 0 | 0 | 13 bits (stored in bits 0 to 12 of SDRmn register) | √ | - | - |
| 1 | 1 | 0 | 1 | 14 bits (stored in bits 0 to 13 of SDRmn register) | √ | - | - |
| 1 | 1 | 1 | 0 | 15 bits (stored in bits 0 to 14 of SDRmn register) | √ | - | - |
| 1 | 1 | 1 | 1 | 16 bits (stored in bits 0 to 15 of SDRmn register) | √ | √ | - |
| Other than above | | | | Setting prohibited | | | |
| Set DLSmn3 to DLSmn0 to 0111B in the simplified $I^2C$ mode. | | | | | | | |

**Caution Be sure to clear bits 6, 10, and 11 to 0.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

### 15.3.5 Higher 7 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n.

If operation is stopped (SEmn = 0), bits 15 to 9 are used as a register that sets the division ratio of the operation clock ($f_{MCK}$). If operation is in progress (SEmn = 1), the SDRmn register functions as a transmit/receive buffer register.

If the CCSmn bit of the serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by the higher 7 bits of the SDRmn register is used as the transfer clock.

For the function of the SDR register when operation is in progress, see **15.2 Configuration of Serial Array Unit**.

SDRmn can be read or written in 16-bit units.

Reset signal generation clears the SDRmn register to 0000H.

**Figure 15-8. Format of Serial Data Register mn (SDRmn)**

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W
FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11)

| | FFF11H (SDR00) | | | | | | | | FFF10H (SDR00) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SDRmn | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| SDRmn[15:9] | | | | | | | Transfer clock set by dividing the operating clock ($f_{MCK}$) |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | $f_{MCK}/2$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | $f_{MCK}/4$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | $f_{MCK}/6$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | $f_{MCK}/8$ |
| • | • | • | • | • | • | • | • |
| • | • | • | • | • | • | • | • |
| • | • | • | • | • | • | • | • |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | $f_{MCK}/254$ |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | $f_{MCK}/256$ |

(**Cautions** and **Remarks** are listed on the next page.)

**Cautions 1. Be sure to clear bits 8 to 0 to 0 if operation is stopped (SEmn = 0).**

**2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.**

**3. Setting SDRmn[15:9] = 0000000B is prohibited when simplified $I^2C$ is used. Set SDRmn[15:9] to 0000001B or greater.**

**4. Do not write eight bits to the lower eight bits if operation is stopped (SEmn = 0). (If these bits are written to, the higher seven bits are cleared to 0.)**

**Remarks 1.** For the function of the SDRmn register when operation is in progress (SEmn = 1), see **15.2 Configuration of Serial Array Unit**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

### 15.3.6 Serial flag clear trigger register mn (SIRmn)

The SIRmn register is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of the serial status register mn is cleared to 0. Because the SIRmn register is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared.

Set the SIRmn register by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the SIRmn register with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears the SIRmn register to 0000H.

**Figure 15-9. Format of Serial Flag Clear Trigger Register mn (SIRmn)**

Address: F0104H, F0105H (SIR00), F0106H, F0107H (SIR01), F0144H, F0145H (SIR10), F0146H, F0147H (SIR11) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SIRmn | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | FECTmn | PECTmn | OVCTmn |

| FECTmn | Clear trigger of framing error of channel n |
|---|---|
| 0 | Not cleared |
| 1 | Clears the FEFmn bit of the SSRmn register to 0. |

| PECTmn | Clear trigger of parity error flag of channel n |
|---|---|
| 0 | Not cleared |
| 1 | Clears the PEFmn bit of the SSRmn register to 0. |

| OVCTmn | Clear trigger of overrun error flag of channel n |
|---|---|
| 0 | Not cleared |
| 1 | Clears the OVFmn bit of the SSRmn register to 0. |

**Cautions 1. Be sure to clear bits 15 to 3 to 0.**

**2. Use the SIRmn register to clear only the error flag set in the SSRn register. If the error flag not set in this register is cleared, the flag may be erased when an error is detected from reading to clearing this error flag.**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

**2.** When the SIRmn register is read, 0000H is always read.

**3.** If the clear trigger bit is set to 1 and the corresponding error flag is set to 1 at the same time, the error flag setting is prioritized.

### 15.3.7 Serial status register mn (SSRmn)

The SSRmn register is a register that indicates the communication status and error occurrence status of channel n. The errors indicated by this register are a framing error, parity error, and overrun error.

The SSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSRmn register can be read with an 8-bit memory manipulation instruction with SSRmnL.

Reset signal generation clears the SSRmn register to 0000H.

**Figure 15-10. Format of Serial Status Register mn (SSRmn) (1/2)**

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01), F0140H, F0141H (SSR10), F0142H, F0143H (SSR11) After reset: 0000H R

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSRmn | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TSF mn | BFF mn | 0 | 0 | FEF mn | PEF mn | OVF mn |

| TSF mn | Communication status indication flag of channel n |
|---|---|
| 0 | Communication is stopped or suspended. |
| 1 | Communication is in progress. |
| <Clear conditions><br>• The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended).<br>• Communication ends.<br><Set condition><br>• Communication starts. | |

| BFF mn | Buffer register status indication flag of channel n |
|---|---|
| 0 | Valid data is not stored in the SDRmn register. |
| 1 | Valid data is stored in the SDRmn register. |
| <Clear conditions><br>• Transferring transmit data from the SDRmn register to the shift register ends during transmission.<br>• Reading receive data from the SDRmn register ends during reception.<br>• The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled).<br><Set conditions><br>• Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode).<br>• Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode).<br>• A reception error occurs. | |

**Caution If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVEmn = 1) is detected.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

**Figure 15-10. Format of Serial Status Register mn (SSRmn) (2/2)**

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01), F0140H, F0141H (SSR10), F0142H, F0143H (SSR11) After reset: 0000H R

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSRmn | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TSFmn | BFFmn | 0 | 0 | FEFmn | PEFmn | OVFmn |

| FEFmn | Framing error detection flag of channel n |
|---|---|
| 0 | No error occurs. |
| 1 | An error occurs (during UART reception). |
| <Clear condition><br>• 1 is written to the FECTmn bit of the SIRmn register.<br><Set condition><br>• A stop bit is not detected when UART reception ends. | |

| PEFmn | Parity error detection flag of channel n |
|---|---|
| 0 | No error occurs. |
| 1 | An error occurs (during UART reception) or ACK is not detected (during $I^2C$ transmission). |
| <Clear condition><br>• 1 is written to the PECTmn bit of the SIRmn register.<br><Set condition><br>• The parity of the transmit data and the parity bit do not match when UART reception ends (parity error).<br>• No ACK signal is returned from the slave channel at the ACK reception timing during $I^2C$ transmission (ACK is not detected). | |

| OVFmn | Overrun error detection flag of channel n |
|---|---|
| 0 | No error occurs. |
| 1 | An error occurs. |
| <Clear condition><br>• 1 is written to the OVCTmn bit of the SIRmn register.<br><Set condition><br>• Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode).<br>• Transmit data is not ready for slave transmission or transmission and reception in CSI mode. | |

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

### 15.3.8 Serial channel start register m (SSm)

The SSm register is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1 (Operation is enabled). Because the SSmn bit is a trigger bit, it is cleared immediately when SEmn = 1.

Set the SSm register by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the SSm register with an 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears the SSm register to 0000H.

**Figure 15-11. Format of Serial Channel Start Register m (SSm)**

Address: F0112H, F0113H (SS0) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SS0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SS01 | SS00 |

Address: F0152H, F0153H (SS1) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SS1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SS11 | SS10 |

| SSmn | Operation start trigger of channel n |
|---|---|
| 0 | No trigger operation |
| 1 | Sets the SEmn bit to 1 and enters the communication wait status Note. |

**Note** If a communication operation is already under execution, the operation is stopped.

The value of the control register and shift register, and the status of the serial clock I/O pin, serial data output pin, and each error flag (FEFmn: framing error flag, PEFmn: parity error flag, OVFmn: overrun error flag) are held.

**Cautions 1. Be sure to clear bits 15 to 2 of the SS0 register and bits 15 to 2 of the SS1 register to 0.**

**2. For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more $f_{MCK}$ clocks have elapsed.**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

**2.** When the SSm register is read, 0000H is always read.

### 15.3.9 Serial channel stop register m (STm)

The STm register is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (STmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

Set the STm register by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the STm register with a 1-bit or 8-bit memory manipulation instruction with STmL.

Reset signal generation clears the STm register to 0000H.

**Figure 15-12. Format of Serial Channel Stop Register m (STm)**

Address: F0114H, F0115H (ST0) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ST0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ST01 | ST00 |

Address: F0154H, F0155H (ST1) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ST1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ST11 | ST10 |

| STmn | Operation stop trigger of channel n |
|---|---|
| 0 | No trigger operation |
| 1 | Clears the SEmn bit to 0 and stops the communication operation[Note]. |

**Note** Communication stops while holding the value of the control register and shift register, and the status of the serial clock I/O pin, serial data output pin, and each error flag (FEFmn: framing error flag, PEFmn: parity error flag, OVFmn: overrun error flag).

**Caution Be sure to clear bits 15 to 2 of the ST0 register and bits 15 to 2 of the ST1 register to 0.**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

**2.** When the STm register is read, 0000H is always read.

### 15.3.10 Serial channel enable status register m (SEm)

The SEm register indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1. When 1 is written a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of the CKOmn bit (serial clock output of channel n) of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial clock pin.

Channel n that stops operation can set the value of the CKOmn bit of the SOm register by software and output its value from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

Read the SEm register by a 16-bit memory manipulation instruction.

Read the lower 8 bits of the SEm register with a 1-bit or 8-bit memory manipulation instruction with SFmL.

Reset signal generation clears the SEm register to 0000H.

**Figure 15-13. Format of Serial Channel Enable Status Register m (SEm)**

Address: F0110H, F0111H (SE0) After reset: 0000H R

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SE0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SE01 | SE00 |

Address: F0150H, F0151H (SE1) After reset: 0000H R

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SE1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SE11 | SE10 |

| SEmn | Indication of operation enable/stop status of channel n |
|---|---|
| 0 | Operation stops Note |
| 1 | Operation is enabled. |

**Note** The control register, shift register value, serial clock I/O pin, serial data output pin, and error flags (FEFmn: framing error flag, PEFmn: parity error flag, OVFmn: over error flag)) are stopped with the state retained. Bits 6 and 5 (TSFmn and BFFmn) in the SSRmn register are cleared.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

### 15.3.11 Serial output enable register m (SOEm)

The SOEm register is a register that is used to enable or stop output of the serial communication operation of each channel.

Channel n that enables serial output cannot rewrite by software the value of the SOmn bit of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform of the start condition and stop condition can be created by software.

Set the SOEm register by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the SOEm register with a 1-bit or 8-bit memory manipulation instruction with SOEmL.

Reset signal generation clears the SOEm register to 0000H.

**Figure 15-14. Format of Serial Output Enable Register m (SOEm)**

Address: F011AH, F011BH After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOE0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOE01 | SOE00 |

Address: F015AH, F015BH After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOE1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOE11 | SOE10 |

| SOEmn | Serial output enable/stop of channel n |
|---|---|
| 0 | Stops output by serial communication operation. |
| 1 | Enables output by serial communication operation. |

**Caution Be sure to clear bits 15 to 2 of the SOE0 register and bits 15 to 2 of the SOE1 register to 0.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

### 15.3.12 Serial output register m (SOm)

The SOm register is a buffer register for serial output of each channel.

The value of the SOmn bit of this register is output from the serial data output pin of channel n.

The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.

The SOmn bit of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

The CKOmn bit of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of the CKOmn bit can be changed only by a serial communication operation.

To use the pin for serial interface as a port function pin, set the corresponding CKOmn and SOmn bits to 1.

Set the SOm register by a 16-bit memory manipulation instruction.

Reset signal generation clears the SOm register to 0303H.

**Figure 15-15. Format of Serial Output Register m (SOm)**

Address: F0118H, F0119H After reset: 0303H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SO0 | 0 | 0 | 0 | 0 | 0 | 0 | CKO01 | CKO00 | 0 | 0 | 0 | 0 | 0 | 0 | SO01 | SO00 |

Address: F0158H, F0159H After reset: 0303H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SO1 | 0 | 0 | 0 | 0 | 0 | 0 | CKO11 | CKO10 | 0 | 0 | 0 | 0 | 0 | 0 | SO11 | SO10 |

| CKOmn | Serial clock output of channel n |
|---|---|
| 0 | Serial clock output value is 0. |
| 1 | Serial clock output value is 1. |

| SOmn | Serial data output of channel n |
|---|---|
| 0 | Serial data output value is 0. |
| 1 | Serial data output value is 1. |

**Caution Be sure to clear bits 15 to 12 and 7 to 2 of the SOm register to 0.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

### 15.3.13 Serial output level register m (SOLm)

The SOLm register is a register that is used to set inversion of the data output level of each channel.

This register can be set only in the UART mode. Be sure to set 0 for corresponding bit in the CSI mode and simplifies $I^2C$ mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1).

When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting the SOLm register is prohibited when the register is in operation (when SEmn = 1).

Set the SOLm register by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the SOLm register with an 8-bit memory manipulation instruction with SOLmL.

Reset signal generation clears the SOLm register to 0000H.

**Figure 15-16. Format of Serial Output Level Register m (SOLm)**

Address: F0120H, F0121H (SOL0) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOL0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOL01 | SOL00 |

Address: F0160H, F0161H (SOL1) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOL1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOL11 | SOL10 |

| SOLmn | Selects inversion of the level of the transmit data of channel n in UART mode |
|---|---|
| 0 | Communication data is output as is. |
| 1 | Communication data is inverted and output. |

**Caution Be sure to clear bits 15 to 2 of the SOL0 register and bits 15 to 2 of the SOL1 register to 0.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

### 15.3.14 Serial slave select enable register m (SSEm)

The SSEm register controls the $\overline{\text{SSI}}$mn pin input of the channel during CSI communication and in slave mode. While a high-level signal is being input to the $\overline{\text{SSI}}$mn pin, no transmission/reception operation is performed even if a serial clock is input. While a low-level signal is being input to the $\overline{\text{SSI}}$mn pin, a transmission/reception operation is performed according to each mode setting if a serial clock is input.

Reset signal generation clears the SSEm register to 0000H.

**Cautions 1. Writing is prohibited other than during CSI communication and in slave mode.**

**2. Can be set only when the SAU is stopped (SEmn = 0).**

**Figure 15-17. Format of Serial Slave Select Enable Register m (SSEm)**

Address: F0122H, F0123H (SSE0) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSE0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSE01 | SSE00 |

Address: F0162H, F0163H (SSE1) After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSE1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSE11 | SSE10 |

| SSEmn[Note] | Channel n $\overline{\text{SSI}}$mn input setting in CSI communication and slave mode |
|---|---|
| 0 | Disables $\overline{\text{SSI}}$mn pin input. |
| 1 | Enables $\overline{\text{SSI}}$mn pin input. |

**Note** Set CKPmn bit of SCRmn register to 1, when SSEmn = 1.

**Caution** Be sure to clear bits 15 to 2 to 0.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

### 15.3.15 Input switch control register (ISC)

The ISC0 bit of the ISC register is used to realize a LIN-bus communication operation by UART0. Set the ISC0 bit at the same time as setting the TIS17 and TIS16 bits in the TIS1 register (timer input select register 1).

When bit 0 is set to 1, the input signal of the serial data input (RxD0) pin is selected as an external interrupt (INTP0) that can be used to detect a wakeup signal by an INTP0 interrupt.

Set the ISC register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the ISC register to 00H.

**Figure 15-18. Format of Input Switch Control Register (ISC)**

Address: F0073H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ISC | 0 | 0 | 0 | 0 | ISC3 | ISC2 | 0 | ISC0 |

| ISC3 | Switching external input (INTP12) |
|---|---|
| 0 | Uses the input signal of the INTP12 pin as an external interrupt input. |
| 1 | Uses the input signal of the LRxD1 pin as an external interrupt input. |

| ISC2 | Switching external input (INTP11) |
|---|---|
| 0 | Uses the input signal of the INTP11 pin as an external interrupt input. |
| 1 | Uses the input signal of the LRxD0 pin as an external interrupt input. |

| ISC0 | Switching external interrupt (INTP0) input |
|---|---|
| 0 | Uses the input signal of the INTP0 pin as an external interrupt (normal operation). |
| 1 | Uses the input signal of the RxD0 pin as an external interrupt (wakeup signal detection). |

**Caution Be sure to clear bits 7 to 4 and 1 to 0.**

### 15.3.16 Noise filter enable register 0 (NFEN0)

The NFEN0 register is used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for CSI or simplified $I^2C$ communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is enabled, CPU/peripheral hardware clock ($f_{CLK}$) is synchronized with 2-clock match detection.

When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock ($f_{MCK}$) **Note**.

Set the NFEN0 register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the NFEN0 register to 00H.

**Note** For details, see **6.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)** and **6.5.2 Start timing of counter**.

**Figure 15-19. Format of Noise Filter Enable Register 0 (NFEN0)**

Address: F0070H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| NFEN0 | 0 | 0 | 0 | 0 | 0 | SNFEN10 Note | 0 | SNFEN00 |

| SNFEN10 | Use of noise filter of RxD1 pin |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |
| Set the SNFEN10 bit to 1 to use the RxD1 pin.<br>Clear the SNFEN10 bit to 0 to use the other than RxD1 pin. | |

| SNFEN00 | Use of noise filter of RxD0 pin |
|---|---|
| 0 | Noise filter OFF |
| 1 | Noise filter ON |
| Set the SNFEN00 bit to 1 to use the RxD0 pin.<br>Clear the SNFEN00 bit to 0 to use the other than RxD0 pin. | |

**Note** Not provided in the Group A products.

**Caution Be sure to clear the following bits to 0.**

- **Bits 7 to 1 in the Group A products**
- **Bits 7 to 3 and 1 in the products other than above**

### 15.3.17 Port input mode registers 1, 3, 5 to 7, 12 (PIM1, PIM3, PIM5 to PIM7, PIM12)

These registers set the input buffer of ports 1, 3, 5 to 7, and 12 in 1-bit units.

Set the PIM1, PIM3, PIM5 to PIM7, and PIM12 registers by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the PIM1, PIM3, PIM5 to PIM7, and PIM12 registers to 00H.

**Figure 15-20. Format of Port Input Mode Registers 1, 3, 5 to 7, and 12 (PIM1, PIM3, PIM5 to PIM7, PIM12)**

Address F0041H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIM1 | PIM17 | PIM16 | 0 | PIM14 | PIM13 | 0 | PIM11 | PIM10 |

Address F0043H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIM3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PIM30 |

Address F0045H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIM5 | 0 | 0 | 0 | PIM54 | 0 | 0 | 0 | 0 |

Address F0046H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIM6 | 0 | 0 | 0 | 0 | PIM63 | PIM62 | 0 | 0 |

Address F0047H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIM7 | 0 | 0 | 0 | 0 | PIM73 | 0 | PIM71 | PIM70 |

Address F004CH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PIM12 | 0 | 0 | PIM125 | 0 | 0 | 0 | 0 | 0 |

| PIMmn | Pmn pin input buffer selection (m = 1, 3, 5 to 7, 12; n = 0, to 7) |
|---|---|
| 0 | Normal input buffer |
| 1 | TTL input buffer |

### 15.3.18 Port output mode registers 1, 6, 7, 12 (POM1, POM6, POM7, POM12)

These registers set the output mode of ports 1, 6, 7, and 12 in 1-bit units.

Set the POM1, POM6, POM7, and POM12 registers by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the POM1, POM6, POM7, and POM12 registers to 00H.

**Figure 15-21. Format of Port Output Mode Registers 1, 6, 7, and 12 (POM1, POM6, POM7, POM12)**

Address F0051H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| POM1 | POM17 | POM16 | POM15 | POM14 | POM13 | POM12 | POM11 | POM10 |

Address F0056H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| POM6 | 0 | 0 | 0 | 0 | POM63 | POM62 | POM61 | POM60 |

Address F0057H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| POM7 | 0 | 0 | 0 | 0 | 0 | POM72 | POM71 | POM70 |

Address F005CH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| POM12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | POM120 |

| POMmn | Pmn pin output buffer selection (m = 1, 6, 7, 12; n = 0 to 7) |
|---|---|
| 0 | Normal output mode |
| 1 | N-ch open-drain output ($EV_{DD0}$ tolerance) mode |

**Caution Be sure to set bits for pins that are not present to their initial values.**

### 15.3.19 Port mode registers 1, 3 to 7, 12 (PM1, PM3 to PM7, PM12)

These registers set input/output of ports1, 3 to 7, and 12 in 1-bit units.

When using the port (P12) to be shared with the serial data output pin or serial clock output pin for serial data output or serial clock output, set the bit in the port mode register (PMxx) corresponding to each port to 0. And set the bit in the port register (Pxx) corresponding to each port to 1

Example: When using P12/TI11/TO11/(TRDI0D0)/INTP5/SO10/TXD1/SNZOUT3 for serial data output or serial clock output

Set the PM12 bit of the port mode register 1 to 0.

Set the P12 bit of the port register 1 to 1.

When using the ports to be shared with the serial data input pin or serial clock input pin for serial data input or serial clock input, set the bit in the port mode register (PMxx) corresponding to each port to 1. At this time, the bit in the port register (Pxx) may be 0 or 1.

Example: When using P16/TI02/TO02/TRDI0C1/SI00/SDA00/RXD0/TOOLRXD for serial data input or serial clock input

Set the PM16 bit of port mode register 1 to 1.

Set the P16 bit of port register 1 to 0 or 1.

Set the PM1, PM3 to PM7, and PM12 registers by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets the PM1, PM3 to PM7, and PM12 registers to FFH.

**Figure 15-22. Format of Port Mode Registers 1, 3 to 7, 12 (PM1, PM3 to PIM7, PIM12)**

Address: FFF21H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM1 | OM17 | PM16 | PM15 | PM14 | PM13 | PM12 | PM11 | PM10 |

Address: FFF23H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM3 | 1 | 1 | 1 | PM34 | PM33 | PM32 | PM31 | PM30 |

Address: FFF24H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM4 | PM47 | PM46 | PM45 | PM44 | PM43 | PM42 | PM41 | PM40 |

Address: FFF25H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM5 | PM57 | PM56 | PM55 | PM54 | PM53 | PM52 | PM51 | PM50 |

Address: FFF26H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM6 | PM67 | PM66 | PM65 | PM64 | PM63 | PM62 | PM61 | PM60 |

Address: FFF27H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM7 | PM77 | PM76 | PM75 | PM74 | PM73 | PM72 | PM71 | PM70 |

Address: FFF27H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM12 | PM127 | PM126 | PM125 | 1 | 1 | 1 | 1 | PM120 |

| PMmn | Pmn pin I/O mode selection (m = 1, 3 to 7, 12; n = 0 to 7) |
|---|---|
| 0 | Output mode (output buffer on) |
| 1 | Input mode (output buffer off) |

## 15.4 Operation stop mode

Each serial interface of serial array unit has the operation stop mode.

In this mode, serial communication cannot be executed, thus reducing the power consumption.

In addition, the pin for serial interface can be used as port function pins in this mode.

### 15.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

The PER0 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0.

To stop the operation of serial array unit 1, set bit 3 (SAU1EN) to 0.

**Figure 15-23. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units**

**(a) Peripheral enable register 0 (PER0) … Set only the bit of SAUm to be stopped to 0.**

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PER0 | RTCEN | 0 | ADCEN | IICA0EN | SAU1EN | SAU0EN | TAU1EN | TAU0EN |
| | × | | × | × | 0/1 | 0/1 | × | × |

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

**Cautions 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read**
**Note that this does not apply to the following registers.**
- **Input switch control register (ISC)**
- **Noise filter enable register 0 (NFEN0)**
- **Port input mode registers 1, 3, 5 to 7, 12 (PIM1, PIM3, PIM5 to PIM7, PIM12)**
- **Port output mode registers 1, 6, 7, 12 (POM1, POM6, POM7, POM12)**
- **Port mode registers 1, 3 to 7, 12 (PM1, PM3 to PM7, PM12)**
- **Port registers 1, 3 to 7, 12 (P1, P3 to P7, P12)**

**2. Be sure to clear the following bits to 0.**
**Bits 1, 3, 4, and 6 in the RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory**
**Bits 4 and 6 in 30-pin products of the RL78/F13 (CAN and LIN incorporated) and in 30-pin products of the RL78/F14**
**Bit 6 in the products other than above**

**Remark** ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user
m: Unit number (m = 0, 1)

### 15.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

**Figure 15-24. Each Register Setting When Stopping the Operation by Channels (1/2)**

**(a) Serial channel stop register m (STm) … This register is a trigger register that is used to enable stopping communication/count by each channel.**

| STm | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | STm1 0/1 | STm0 0/1 |

1: Clears the SEmn bit to 0 and stops the communication operation

* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

**(b) Serial channel enable status register m (SEm) … This register indicates whether data transmission/reception operation of each channel is enabled or stopped.**

| SEm | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SEm1 0/1 | SEm0 0/1 |

0: Operation stops

* The SEmn register is a read-only status register, whose operation is stopped by using the STm register. With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

**(c) Serial output enable register m (SOEm) … This register is a register that is used to enable or stop output of the serial communication operation of each channel.**

| SOEm | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 0/1 | SOEm0 0/1 |

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software.

**(d) Serial output register m (SOm) …This register is a buffer register for serial output of each channel.**

| SOm | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | CKOm1 0/1 | CKOm0 0/1 | 0 | 0 | 0 | 0 | 0 | 0 | SOm1 0/1 | SOm0 0/1 |

1: Serial clock output value is 1 — 1: Serial data output value is 1

* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to 1.

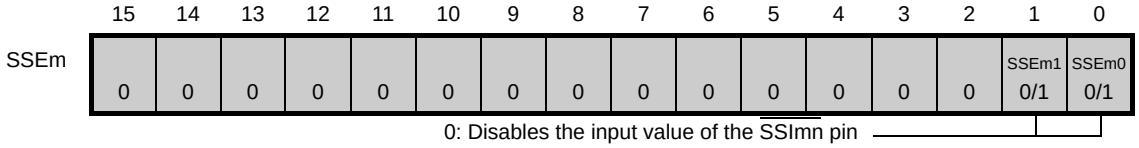
**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

**2.** ☐ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-24. Each Register Setting When Stopping the Operation by Channels (2/2)**

**(e) Serial slave select enable register (SSEm) … This register controls the $\overline{\text{SSImn}}$ pin in each slave channel.**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

**2.** ☐ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

## 15.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) Communication

This is a clocked communication function that uses three lines: serial clock ($\overline{SCK}$) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 to 16 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- Level setting of transmit/receive data

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate
  - During master communication: Max. $f_{MCK}/4$ **Note**
  - During slave communication: Max. $f_{MCK}/6$ **Note**

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00m, CSI01, CSI10, and CSI11 support the slave select input function. For details, refer to **15.6 Clock Synchronous Serial Communication with SPI Function**.

**Note** Use the clocks within a range satisfying the $\overline{SCK}$ cycle time ($t_{KCY}$) characteristics (see **CHAPTER 34** to **CHAPTER 36 ELECTRICAL SPECIFICATIONS**).

The channels supporting 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11) are channels 0 and 1 of SAU0 and channels 0 and 1 of SAU1.

- Group A products

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note2 | | IIC01 |

- Products of Groups C-1 and D-1

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note2 | | IIC01 |
| 1 | 0 | CSI10 (supporting SPI function) Note1, 2 | UART1 | IIC10 |
| | 1 | - | | - |

- Products of Groups B, C-2, D-2, and E

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note2 | | IIC01 |
| 1 | 0 | CSI10 (supporting SPI function) Note1, 2 | UART1 | IIC10 |
| | 1 | CSI11 (supporting SPI function) Note2 | | IIC11 |

**Notes 1.** 48-pin, 32-pin and 30-pin products do not have $\overline{SSI10}$ pin.

<R> **2.** Set CKPmn bit of SCRmn register to 1, when SSEmn = 1 (Enables $\overline{SSImn}$ pin input).
(m = 0, 1, n = 0, 1)

**Remark** Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory
Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory
Group C-1: RL78/F13 (CAN and LIN incorporated) products with 30 or 32 pins
Group C-2: RL78/F13 (CAN and LIN incorporated) products with 48, 64, or 80 pins
Group D-1: RL78/F14 products with 30 or 32 pins
Group D-2: RL78/F14 products with 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory
Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

3-wire serial I/O (CSI00, CSI01, CSI10, CSI11) performs the following six types of communication operations.

- Master transmission (See **15.5.1 Master transmission**.)
- Master reception (See **15.5.2 Master reception**.)
- Master transmission/reception (See **15.5.3 Master transmission/reception**.)
- Slave transmission (See **15.5.4 Slave transmission**.)
- Slave reception (See **15.5.5 Slave reception**.)
- Slave transmission/reception (See **15.5.6 Slave transmission/reception**.)

### 15.5.1 Master transmission

Master transmission is an operation wherein this MCU outputs a transfer clock and transmits data to another device.

| 3-Wire Serial I/O | CSI00 | CSI01 | CSI10 | CSI11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | $\overline{\text{SCK00}}$, SO00 | $\overline{\text{SCK01}}$, SO01 | $\overline{\text{SCK10}}$, SO10 | $\overline{\text{SCK11}}$, SO11 |
| Interrupt | INTCSI00 | INTCSI01 | INTCSI10 | INTCSI11 |
| | Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected. | | | |
| Error detection flag | None | | | |
| Transfer data length | 7 to 16 bits | | | |
| Transfer rate | Max. $f_{MCK}/4$ [Hz]<br>Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] **Note** $f_{CLK}$: System clock frequency | | | |
| Data phase | Selectable by the DAPmn bit of the SCRmn register<br>• DAPmn = 0: Data output starts from the start of the serial clock operation.<br>• DAPmn = 1: Data output starts half a clock before the start of the serial clock operation. | | | |
| Clock phase | Selectable by the CKPmn bit of the SCRmn register<br>• CKPmn = 0: Forward<br>• CKPmn = 1: Reverse | | | |
| Data direction | MSB or LSB first | | | |

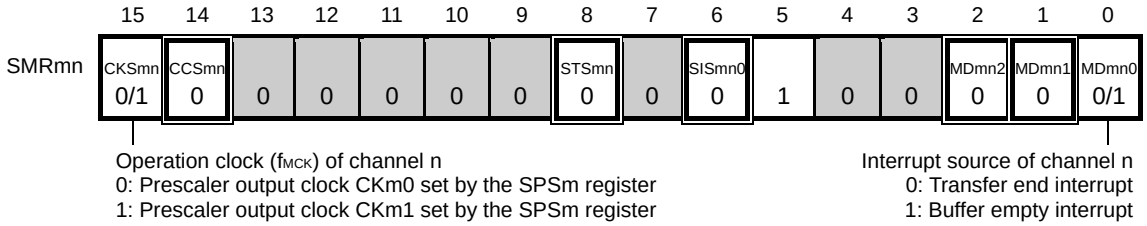
**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

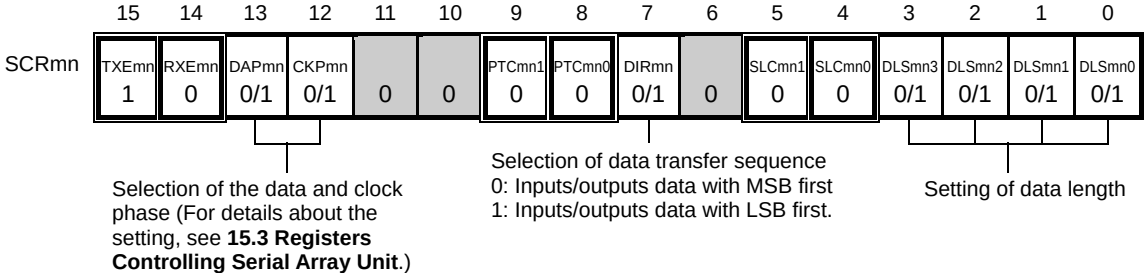
### (1) Register setting

**Figure 15-25. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

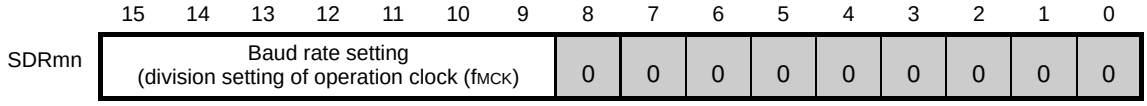



**(b) Serial communication operation setting register mn (SCRmn)**

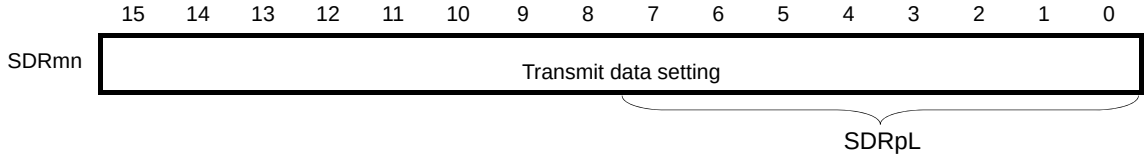



**(c) Serial data register mn (SDRmn)**

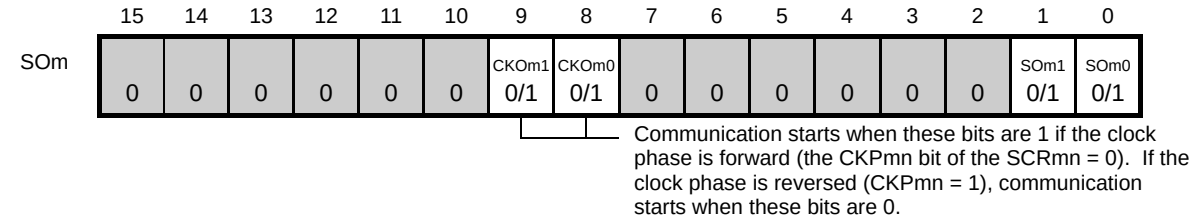
**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)**




**(d) Serial output register m (SOm) … Sets only the bits of the target channel.**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐ : Setting is fixed in the CSI master transmission mode, ▒ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-25. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (2/2)**

**(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 0/1 | SOEm0 0/1 |

**(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

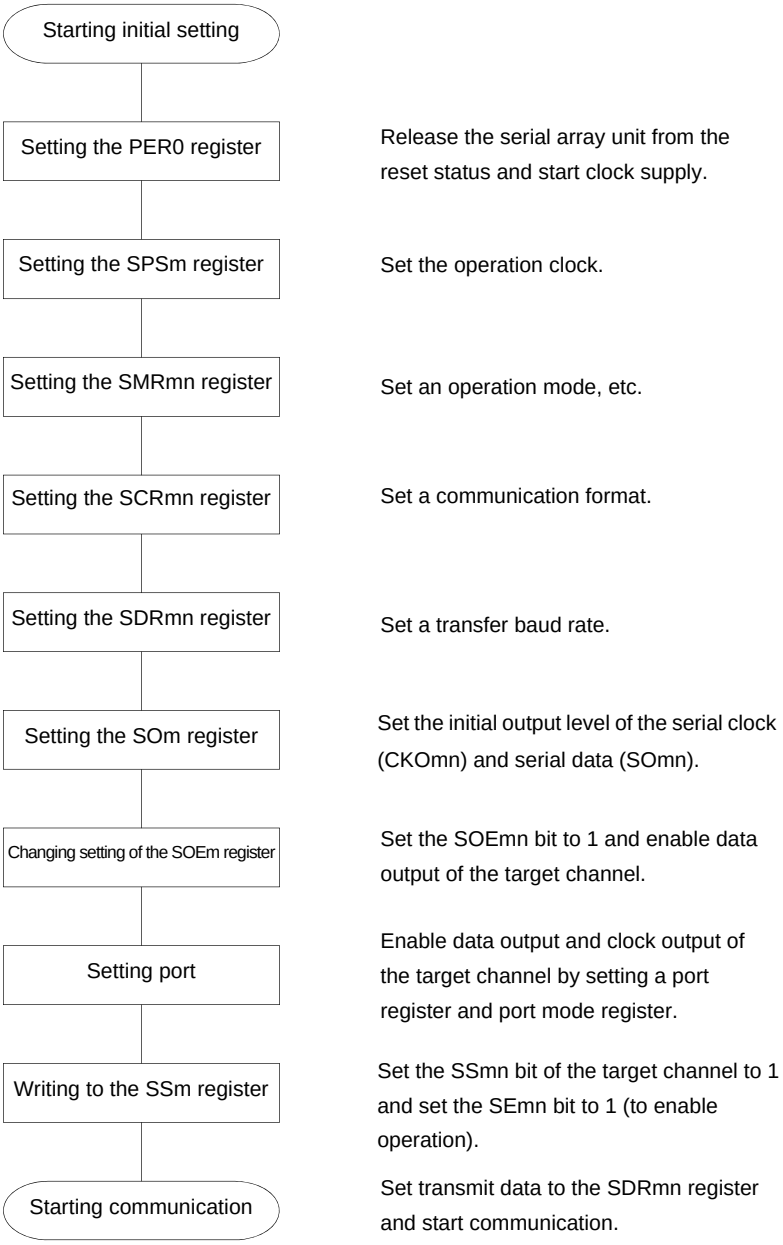
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1 0/1 | SSm0 0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

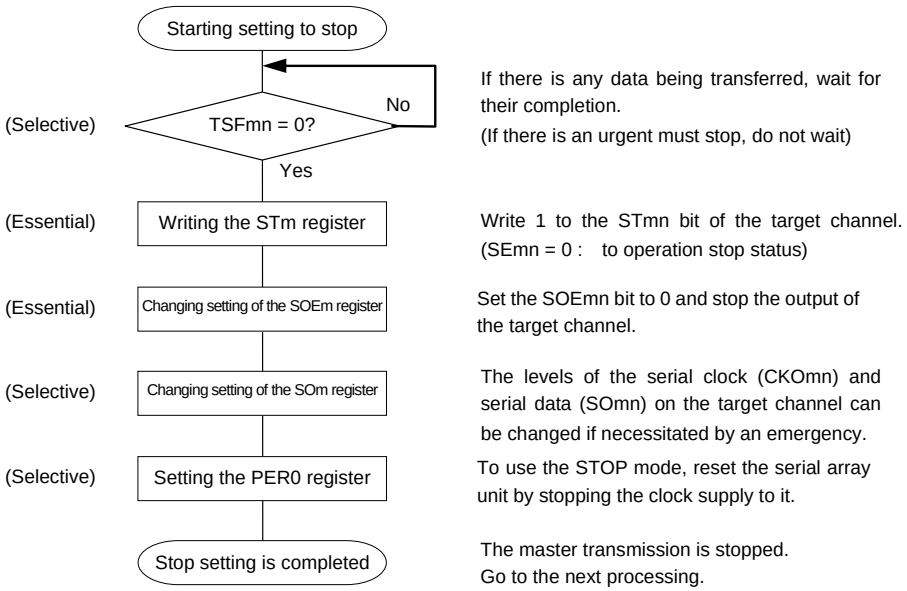
### (2) Operation procedure

**Figure 15-26. Initial Setting Procedure for Master Transmission**



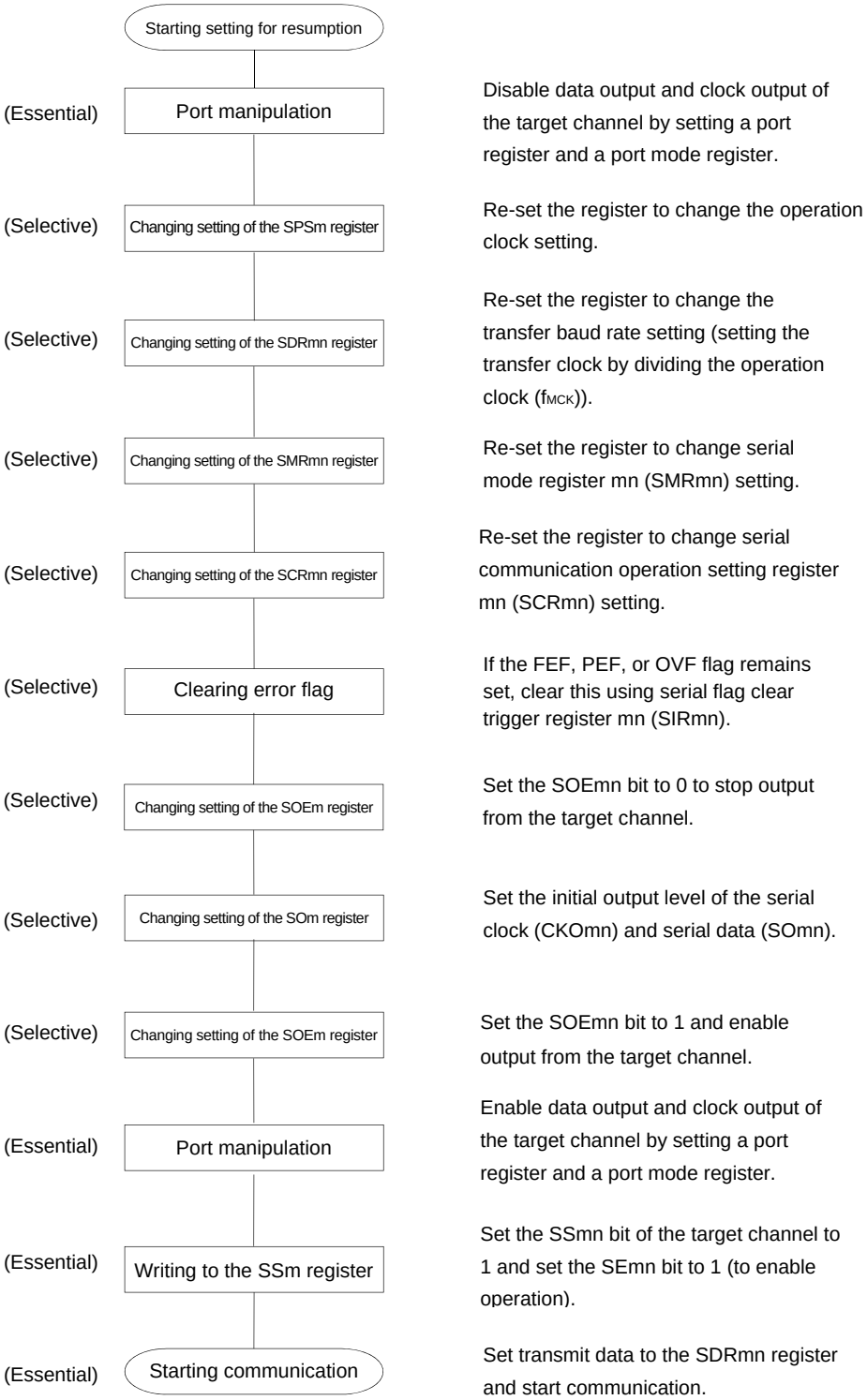

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

**Figure 15-27. Procedure for Stopping Master Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

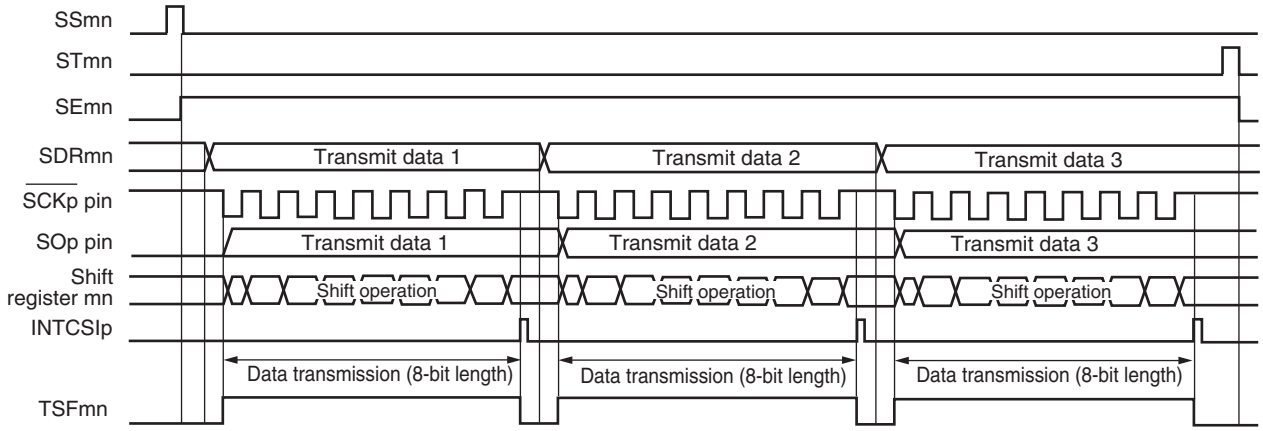
**Figure 15-28. Procedure for Resuming Master Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

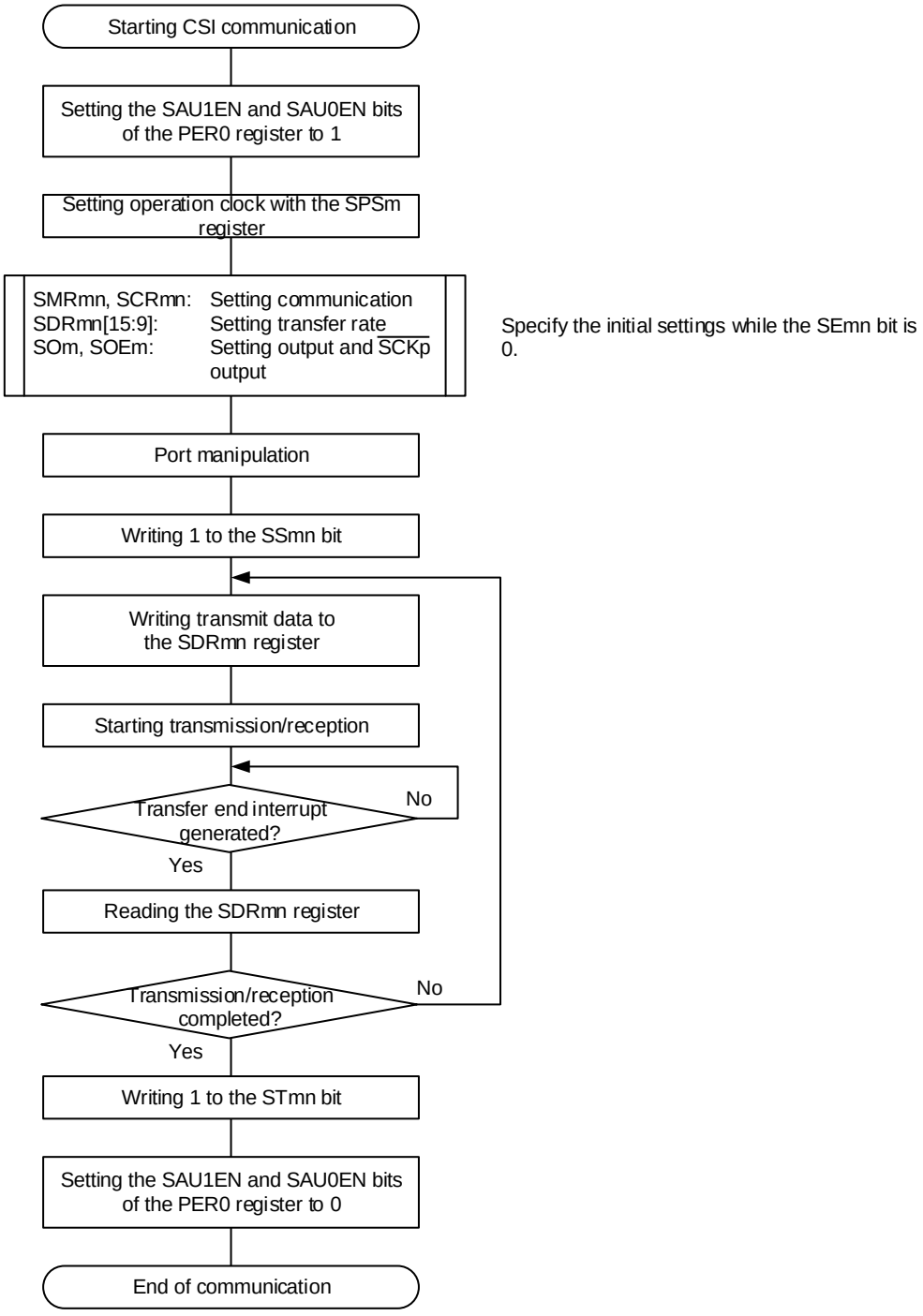
### (3) Processing flow (in single-transmission mode)

**Figure 15-29. Timing Chart of Master Transmission (in Single-Transmission Mode)**
**(Type 1: DAPmn = 0, CKPmn = 0)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

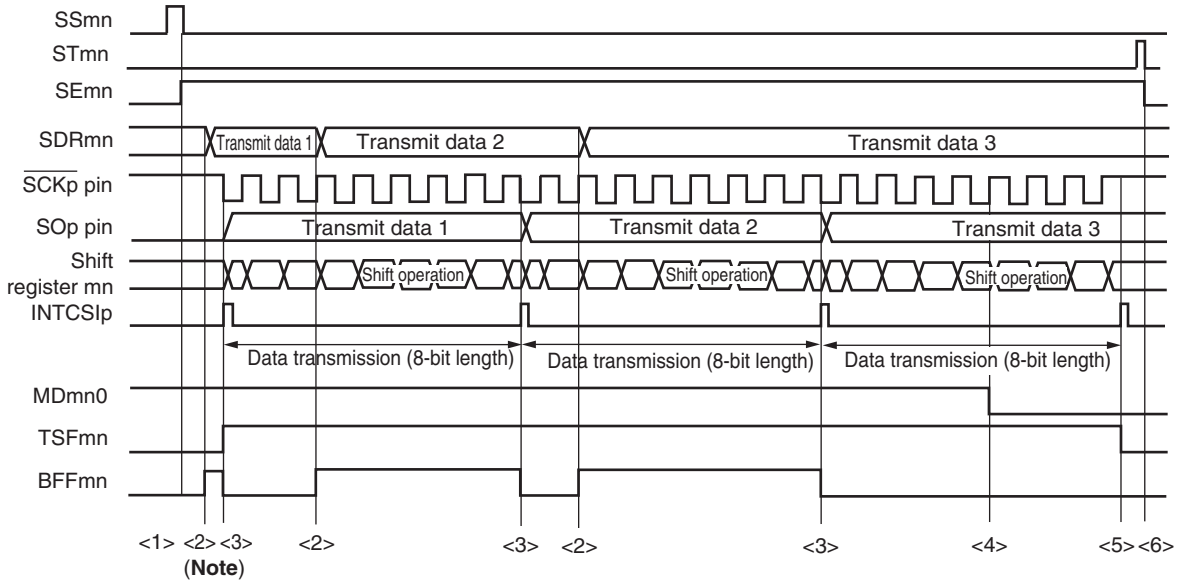
**Figure 15-30. Flowchart of Master Transmission (in Single-Transmission Mode)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### (4) Processing flow (in continuous transmission mode)

**Figure 15-31. Timing Chart of Master Transmission (in Continuous Transmission Mode)**
**(Type 1: DAPmn = 0, CKPmn = 0)**

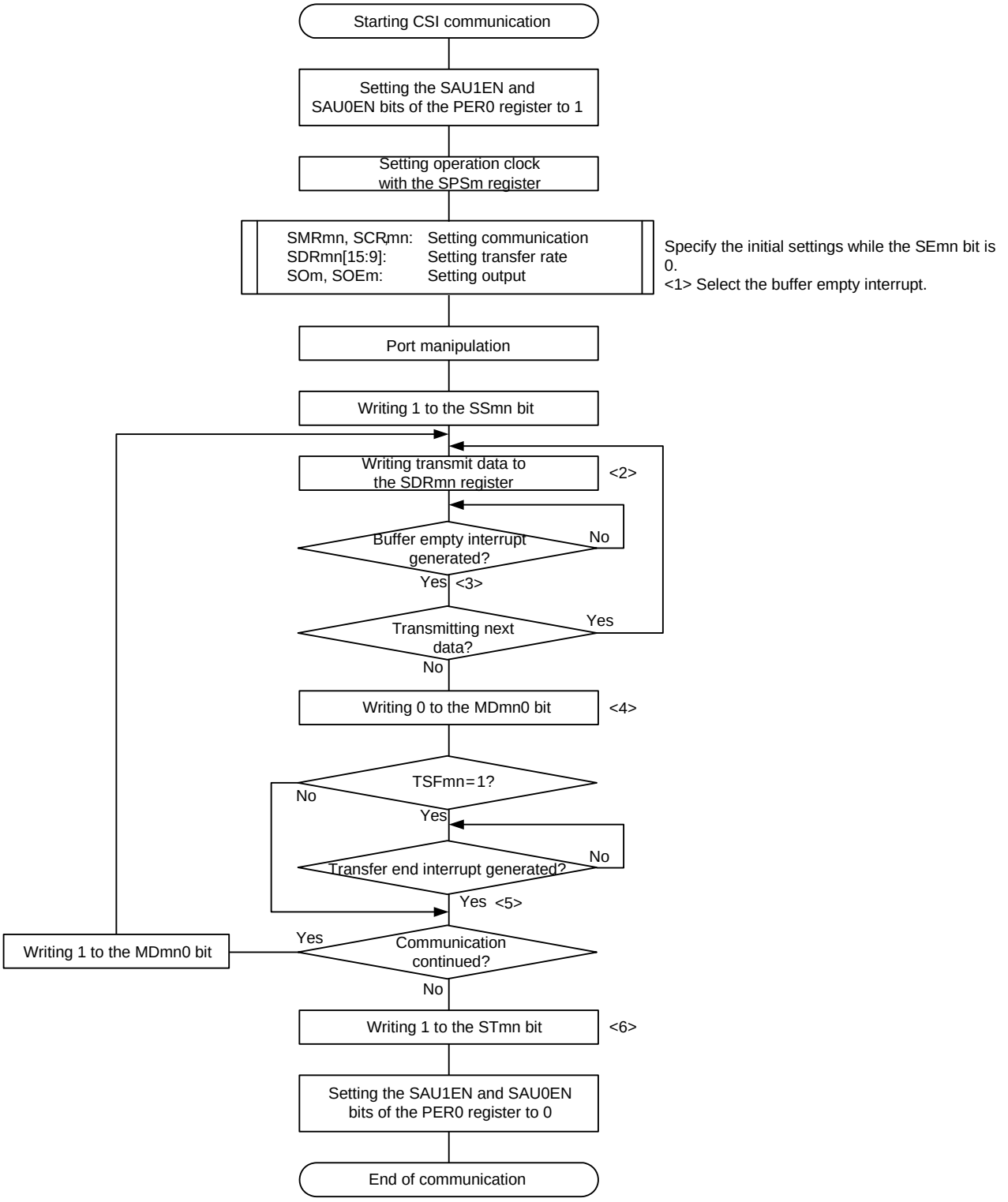



**Note** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

**Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-32. Flowchart of Master Transmission (in Continuous Transmission Mode)**




**Remarks 1.** <1> to <6> in the figure correspond to <1> to <6> in **Figure 15-31 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### 15.5.2 Master reception

Master reception is an operation wherein this MCU outputs a transfer clock and receives data from other device.

| 3-Wire Serial I/O | CSI00 | CSI01 | CSI10 | CSI11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | $\overline{\text{SCK00}}$, SI00 | $\overline{\text{SCK01}}$, SI01 | $\overline{\text{SCK10}}$, SI10 | $\overline{\text{SCK11}}$, SI11 |
| Interrupt | INTCSI00 | INTCSI01 | INTCSI10 | INTCSI11 |
| | Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected. | | | |
| Error detection flag | Overrun error detection flag (OVFmn) only | | | |
| Transfer data length | 7 to 16 bits | | | |
| Transfer rate | Max. $f_{MCK}/4$ [Hz]<br>Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz]**Note** $f_{CLK}$: System clock frequency | | | |
| Data phase | Selectable by the DAPmn bit of the SCRmn register<br>• DAPmn = 0: Data input starts from the start of the serial clock operation.<br>• DAPmn = 1: Data input starts half a clock before the start of the serial clock operation. | | | |
| Clock phase | Selectable by the CKPmn bit of the SCRmn register<br>• CKPmn = 0: Forward<br>• CKPmn = 1: Reverse | | | |
| Data direction | MSB or LSB first | | | |

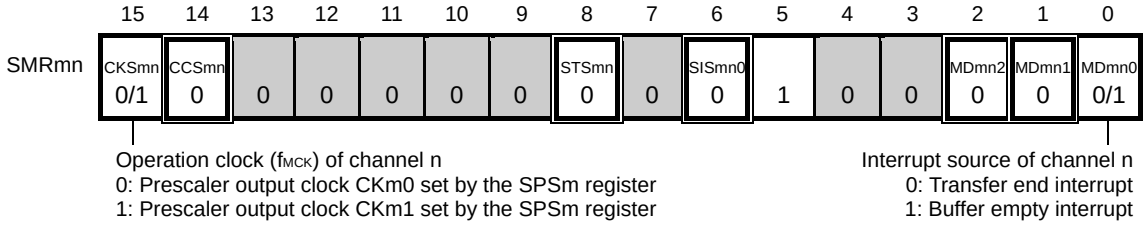
**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

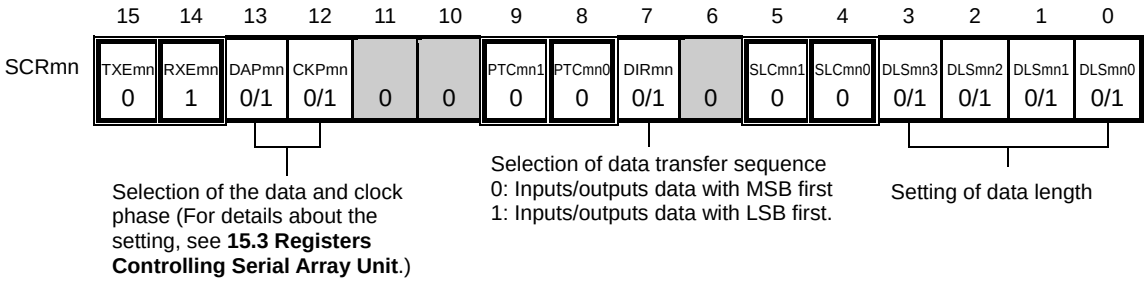
**(1) Register setting**

**Figure 15-33. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

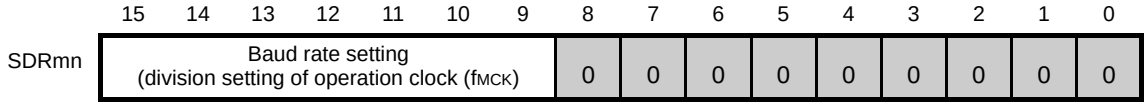



**(b) Serial communication operation setting register mn (SCRmn)**

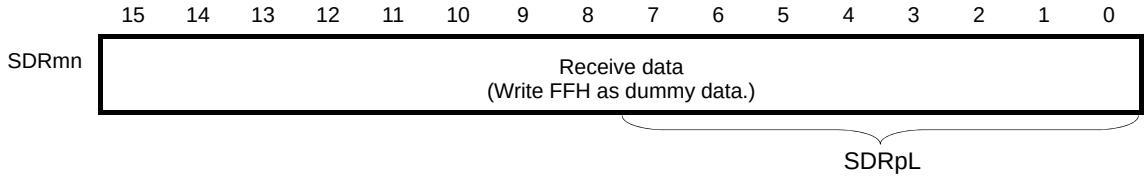



**(c) Serial data register mn (SDRmn)**

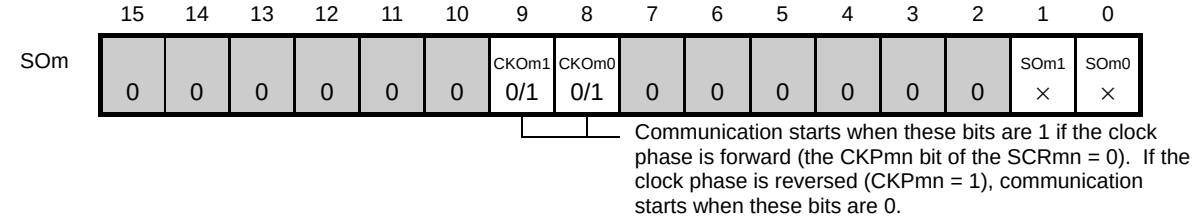
**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)**




**(d) Serial output register m (SOm) … Sets only the bits of the target channel.**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI master reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-33. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (2/2)**

**(e) Serial output enable register m (SOEm) …The register that not used in this mode.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 × | SOEm0 × |

**(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1 0/1 | SSm0 0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI master reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

### (2) Operation procedure

**Figure 15-34. Initial Setting Procedure for Master Reception**

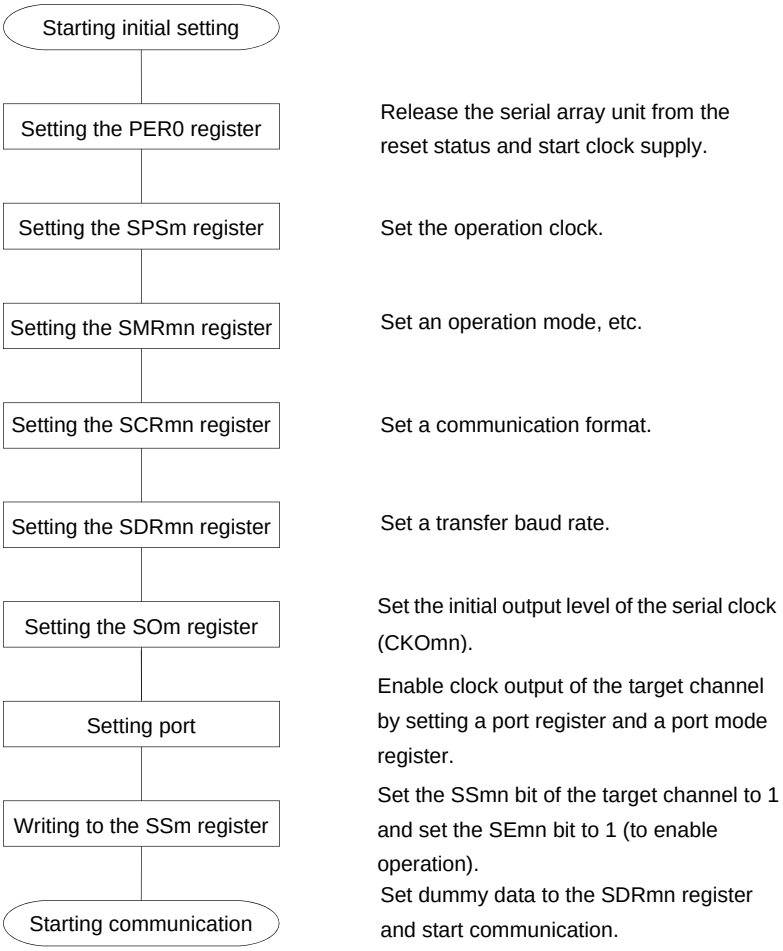
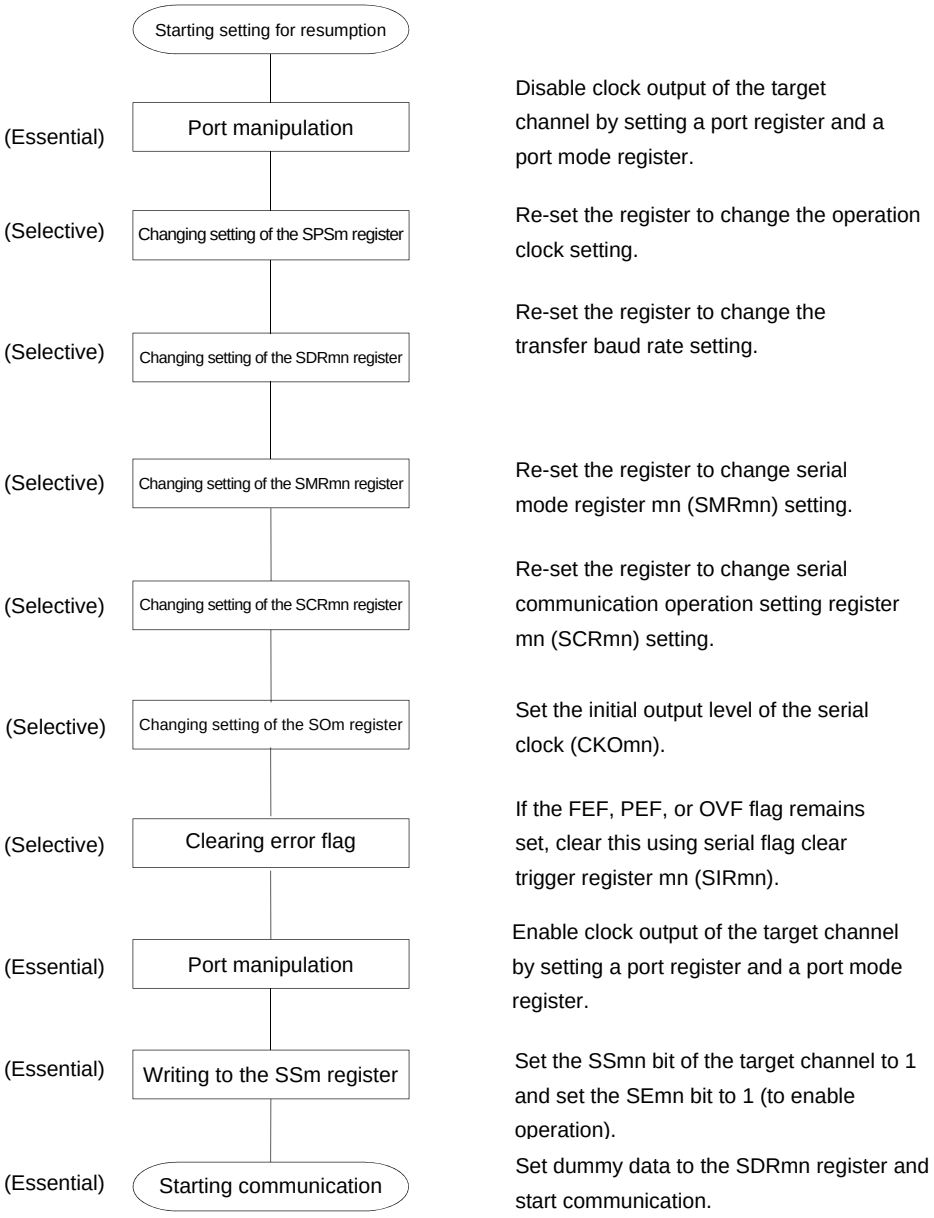



**Figure 15-35. Procedure for Stopping Master Reception**

**Remarks 1.** Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 15-36 Procedure for Resuming Master Reception**).

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

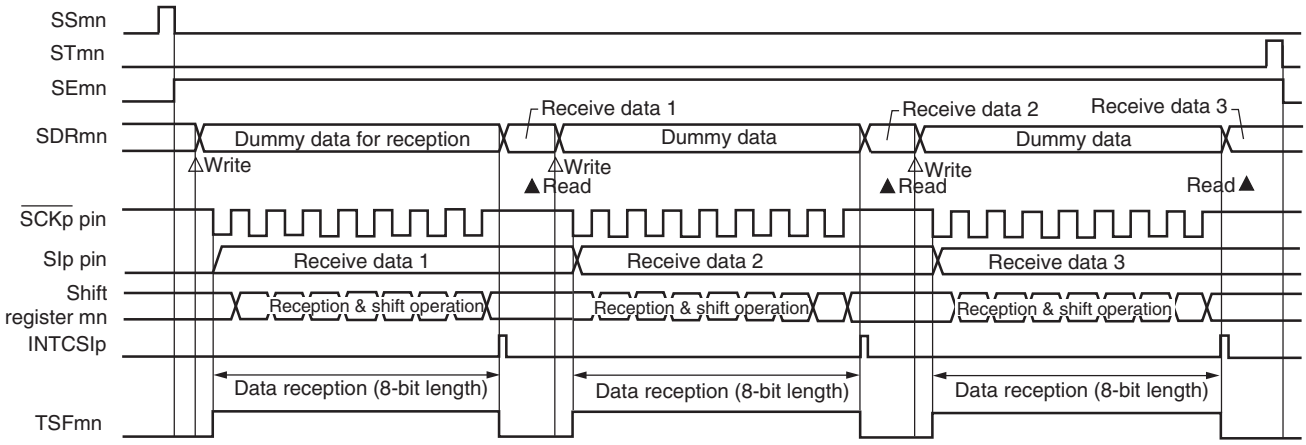
**Figure 15-36. Procedure for Resuming Master Reception**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

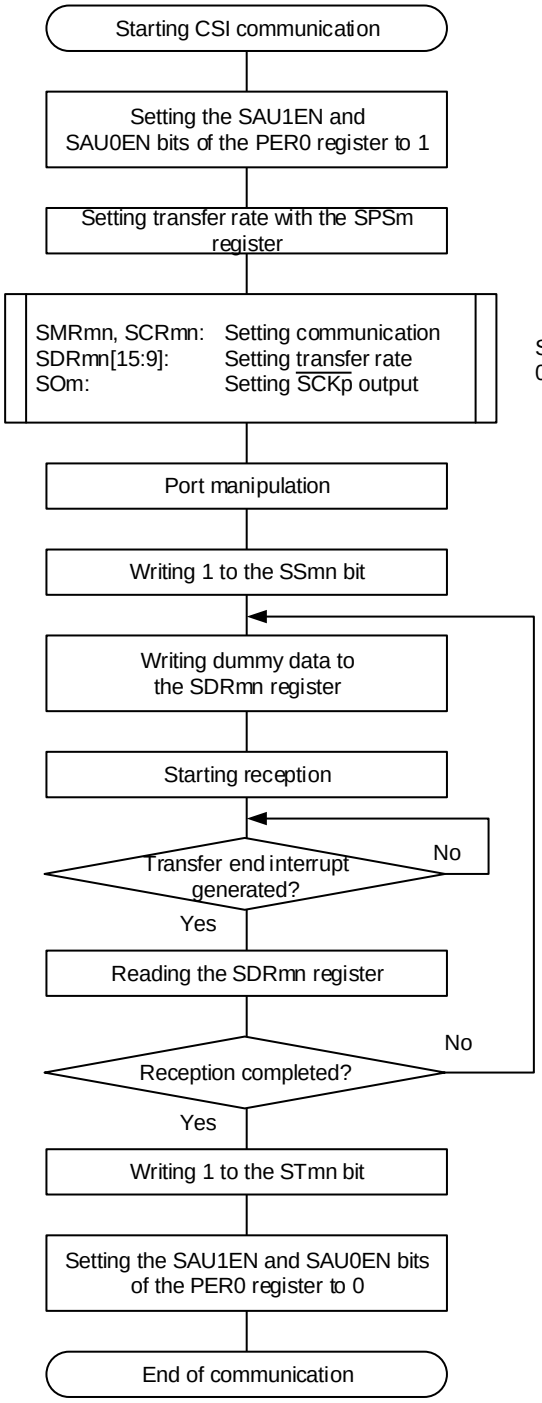
### (3) Processing flow (in single-reception mode)

**Figure 15-37. Timing Chart of Master Reception (in Single-Reception Mode)**
**(Type 1: DAPmn = 0, CKPmn = 0)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

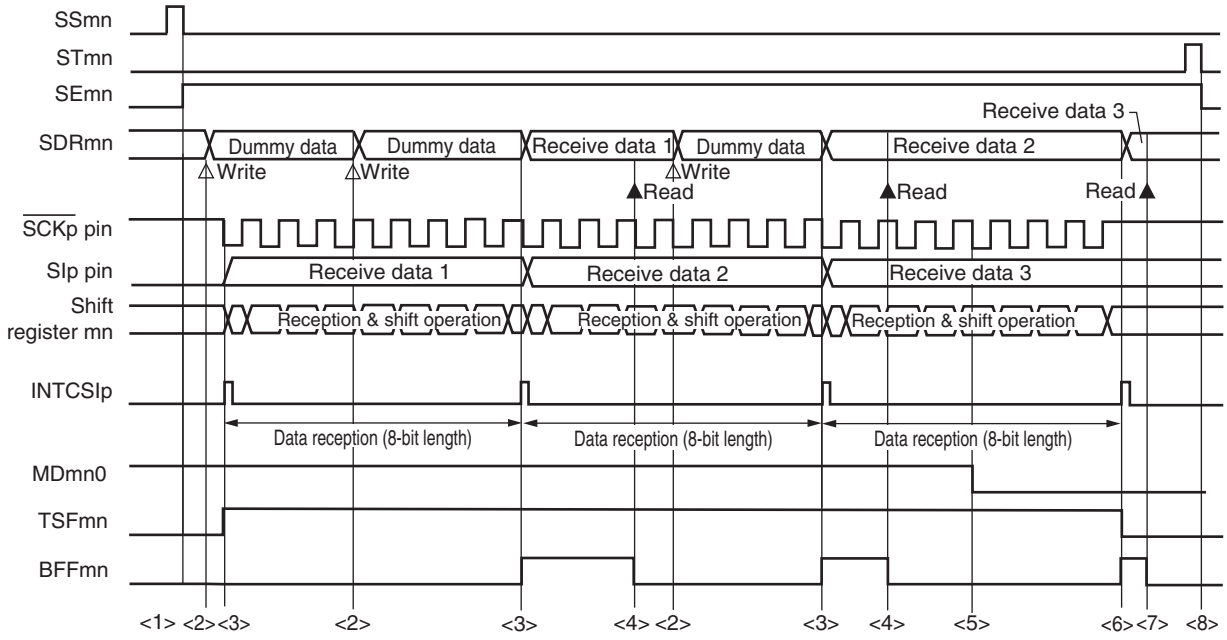
**Figure 15-38. Flowchart of Master Reception (in Single-Reception Mode)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

### (4) Processing flow (in continuous reception mode)

**Figure 15-39. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**

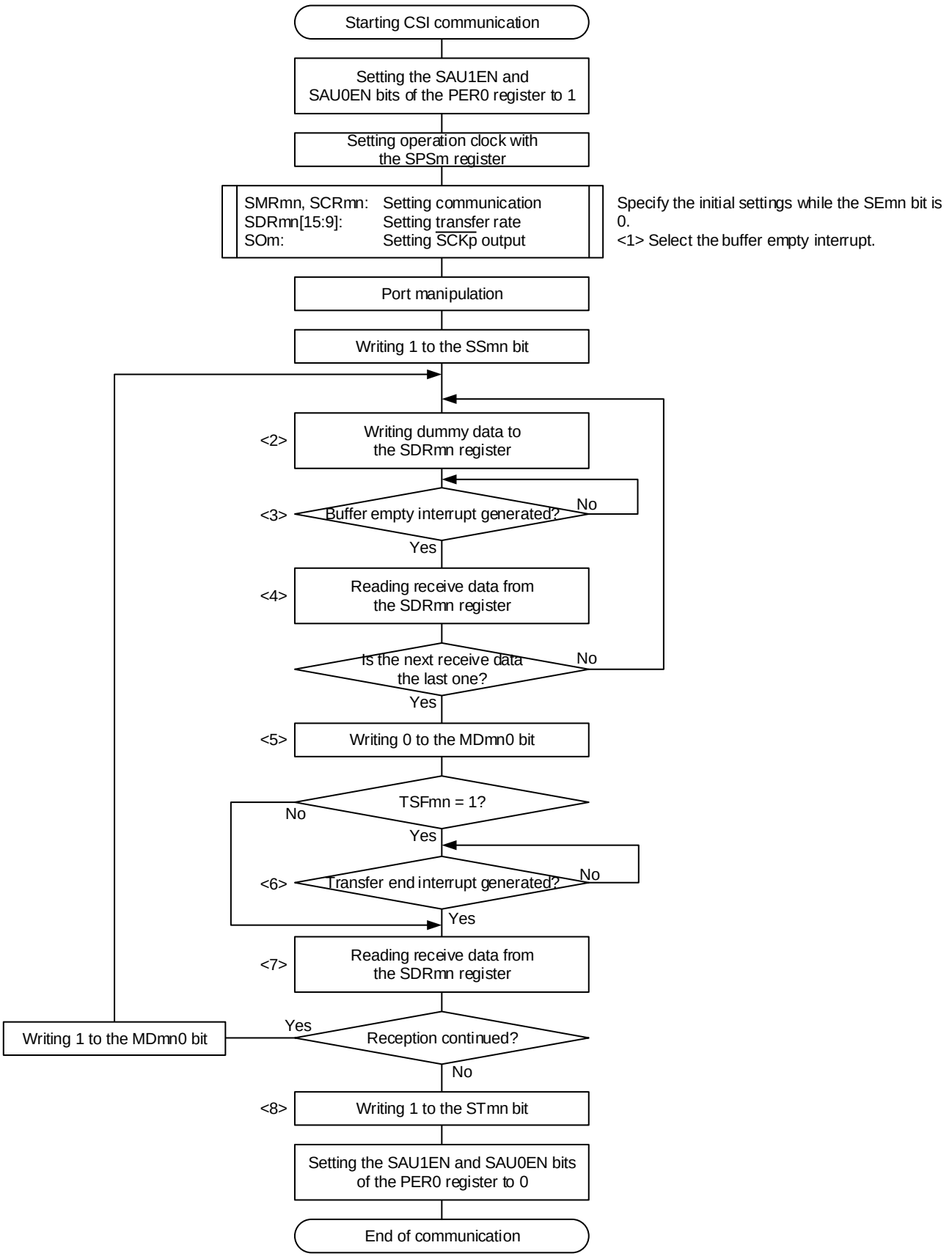



**Caution The MDmn0 bit can be rewritten even during operation.**
**However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.**

**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-40 Flowchart of Master Reception (in Continuous Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-40. Flowchart of Master Reception (in Continuous Reception Mode)**




(**Caution** and **Remarks** are listed on the next page.)

**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-39 Timing Chart of Master Reception (in Continuous Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

### 15.5.3 Master transmission/reception

Master transmission/reception is an operation wherein this MCU outputs a transfer clock and transmits/receives data to/from other device.

| 3-Wire Serial I/O | CSI00 | CSI01 | CSI10 | CSI11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | $\overline{\text{SCK00}}$, SI00, SO00 | $\overline{\text{SCK01}}$, SI01, SO01 | $\overline{\text{SCK10}}$, SI10, SO10 | $\overline{\text{SCK11}}$, SI11, SO11 |
| Interrupt | INTCSI00 | INTCSI01 | INTCSI10 | INTCSI11 |
| | Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected. | | | |
| Error detection flag | Overrun error detection flag (OVFmn) only | | | |
| Transfer data length | 7 to 16 bits | | | |
| Transfer rate | Max. $f_{MCK}/4$ [Hz]<br>Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] **Note** $f_{CLK}$: System clock frequency | | | |
| Data phase | Selectable by the DAPmn bit of the SCRmn register<br>• DAPmn = 0: Data I/O starts at the start of the serial clock operation.<br>• DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation. | | | |
| Clock phase | Selectable by the CKPmn bit of the SCRmn register<br>• CKPmn = 0: Forward<br>• CKPmn = 1: Reverse | | | |
| Data direction | MSB or LSB first | | | |

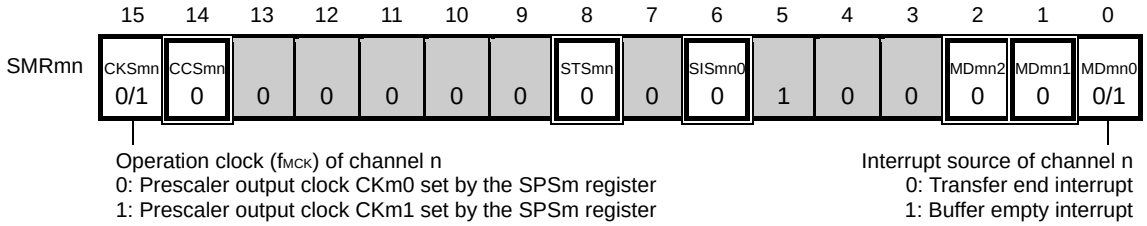
**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

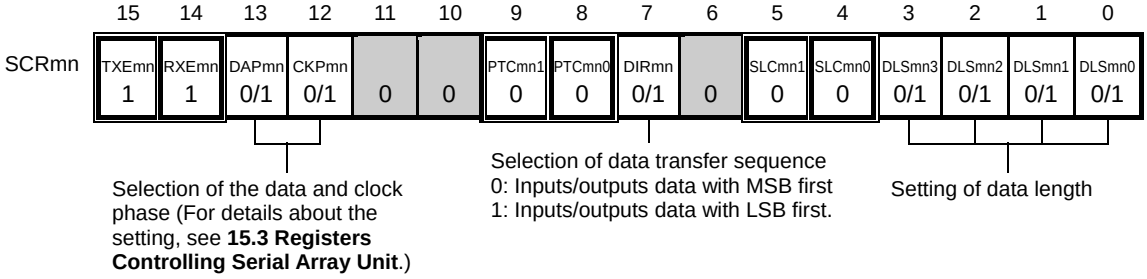
### (1) Register setting

**Figure 15-41. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

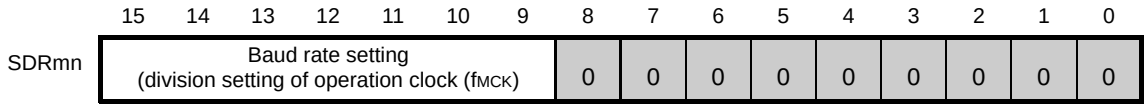



**(b) Serial communication operation setting register mn (SCRmn)**

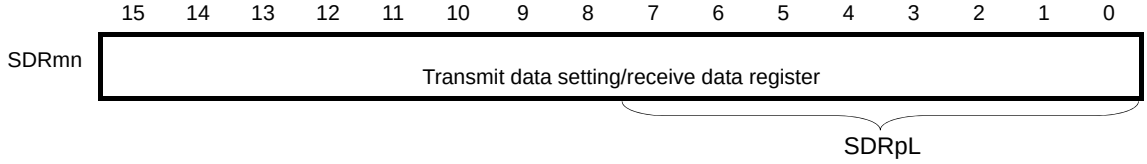



**(c) Serial data register mn (SDRmn)**

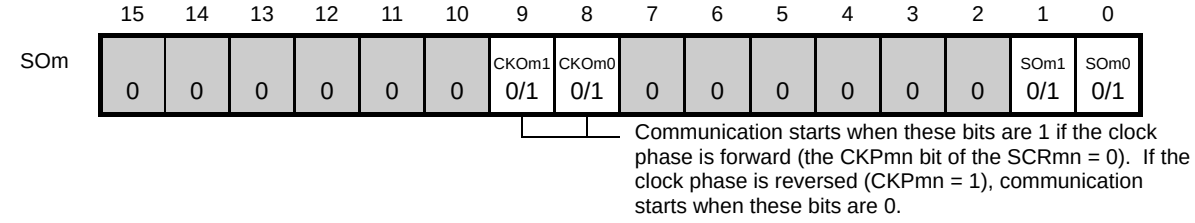
**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)**




**(d) Serial output register m (SOm) … Sets only the bits of the target channel.**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI master transmission/reception mode

▒: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-41. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (2/2)**

**(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.**

SOEm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1<br>0/1 | SOEm0<br>0/1 |

**(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

SSm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1<br>0/1 | SSm0<br>0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

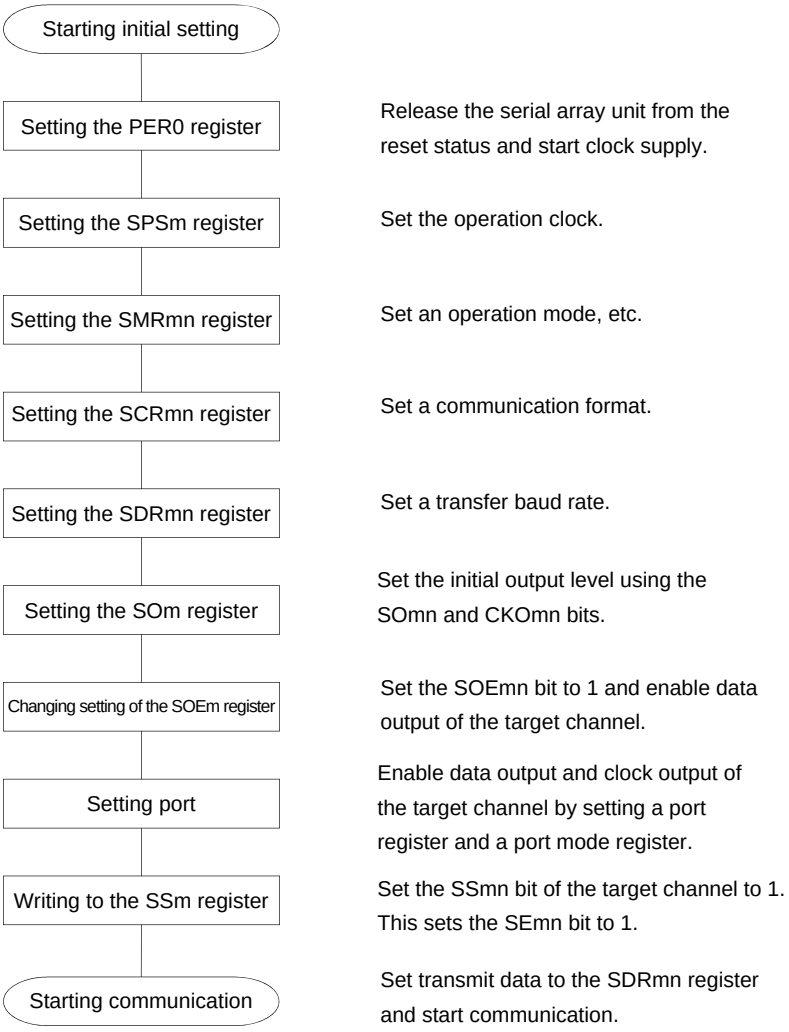
**2.** ☐: Setting is fixed in the CSI master transmission/reception mode

: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

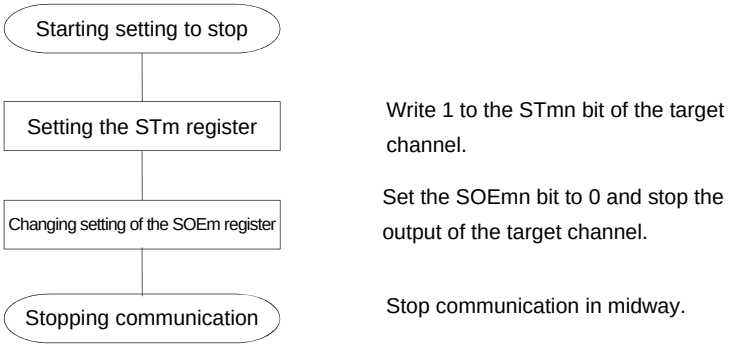
### (2) Operation procedure

**Figure 15-42. Initial Setting Procedure for Master Transmission/Reception**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

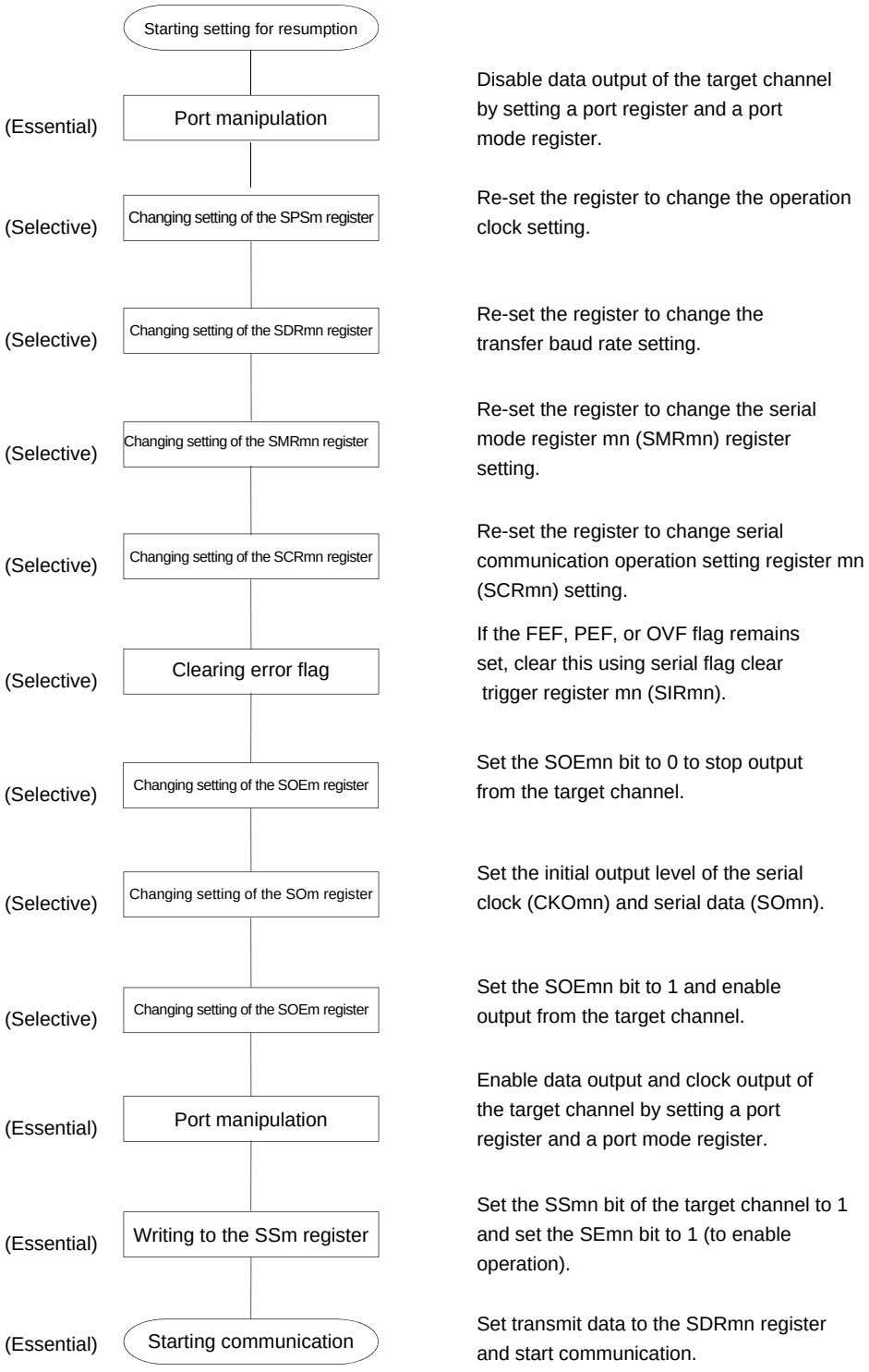
**Figure 15-43. Procedure for Stopping Master Transmission/Reception**




**Remarks 1.** Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 15-44 Procedure for Resuming Master Transmission/ Reception**).

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

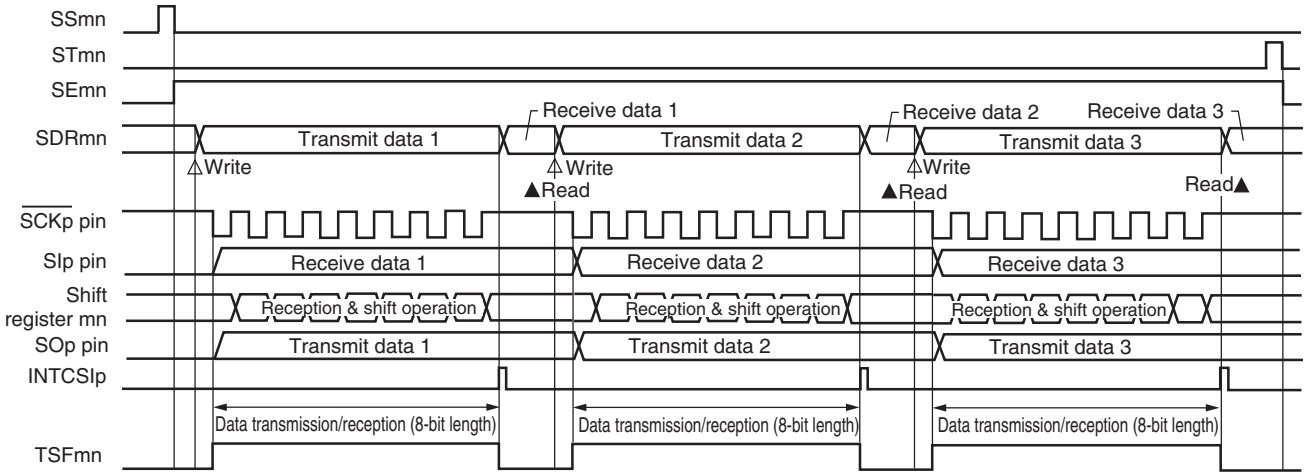
**Figure 15-44. Procedure for Resuming Master Transmission/Reception**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

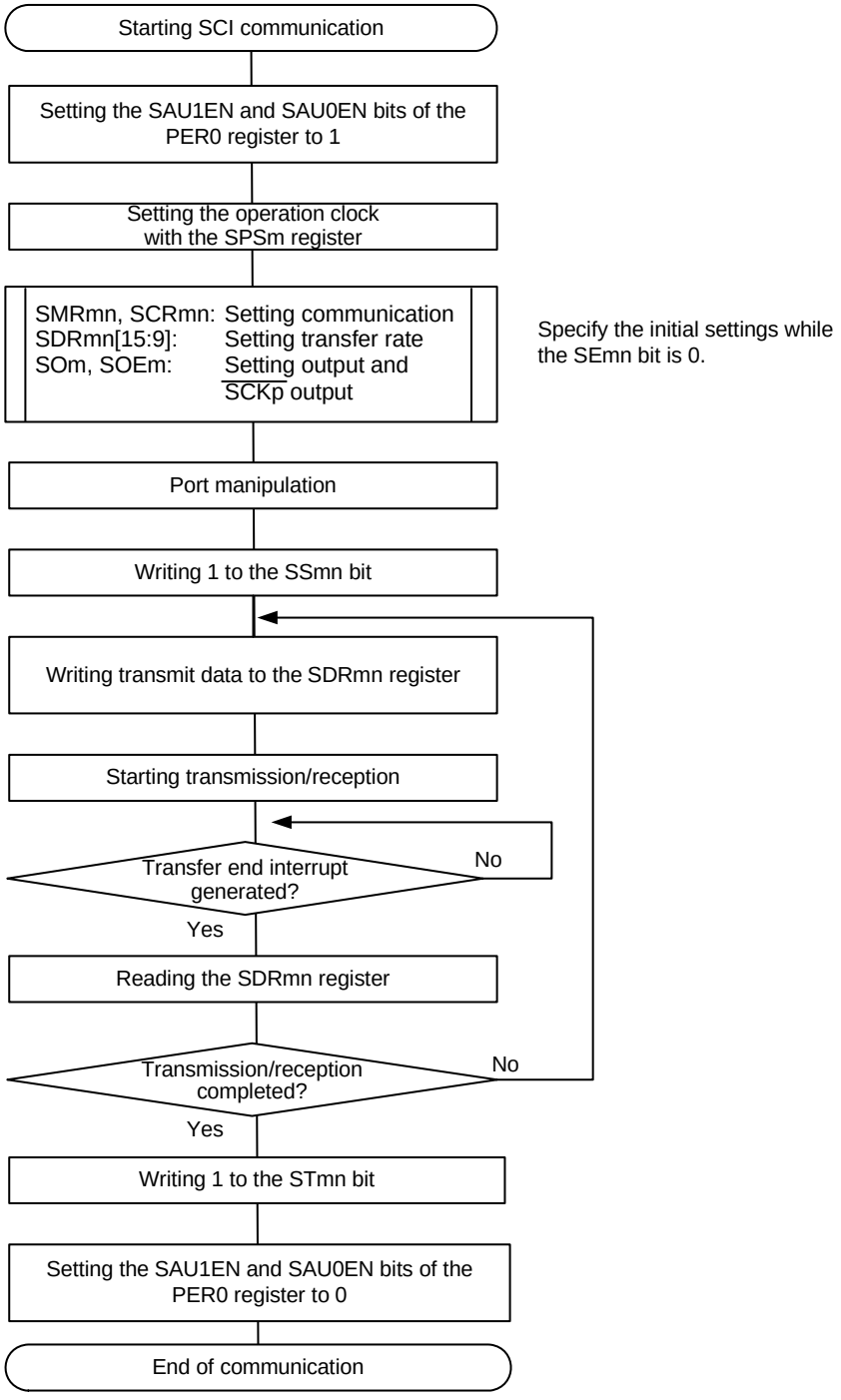
### (3) Processing flow (in single-transmission/reception mode)

**Figure 15-45. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

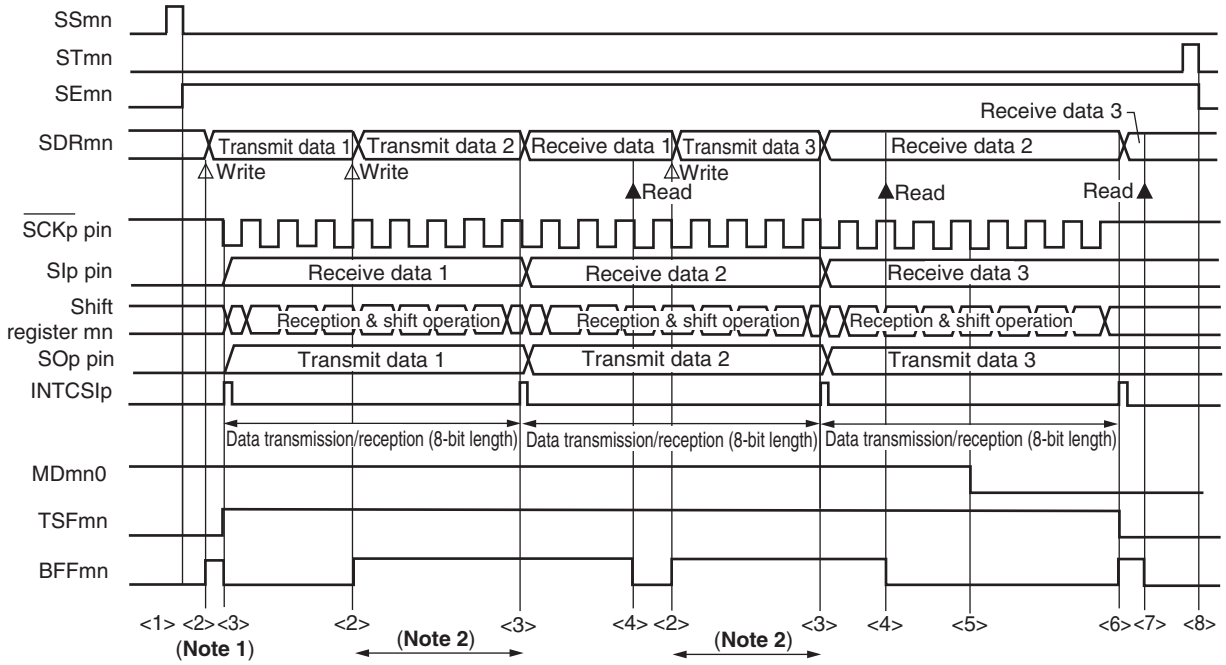
**Figure 15-46. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

### (4) Processing flow (in continuous transmission/reception mode)

**Figure 15-47. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**




**Notes 1.** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

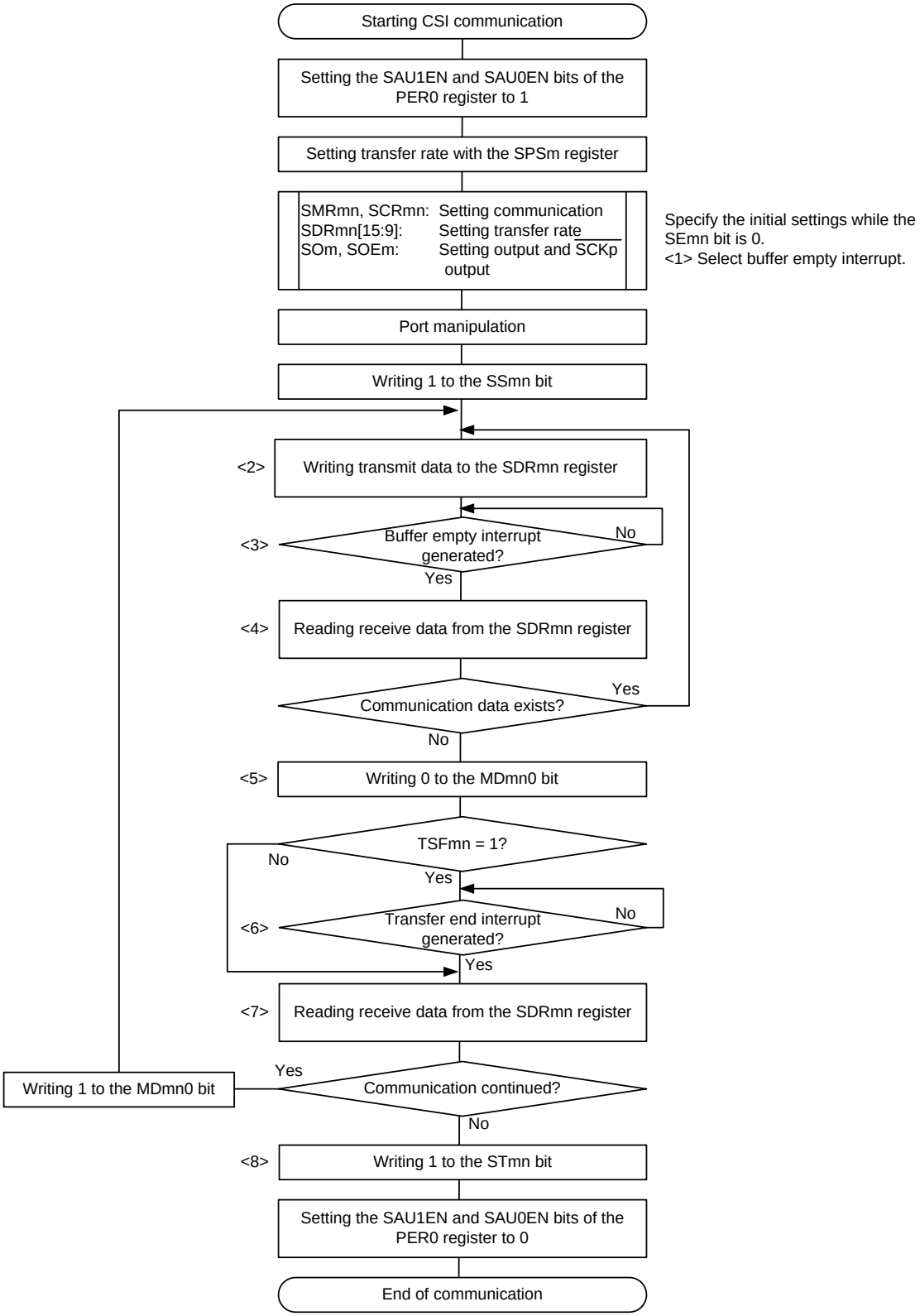
**2.** The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

**Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.**

**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-48 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-48. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**




**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-47 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### 15.5.4 Slave transmission

Slave transmission is an operation wherein this MCU transmits data to another device in the state of a transfer clock being input from another device.

| 3-Wire Serial I/O | CSI00 | CSI01 | CSI10 | CSI11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | $\overline{\text{SCK00}}$, SO00 | $\overline{\text{SCK01}}$, SO01 | $\overline{\text{SCK10}}$, SO10 | $\overline{\text{SCK11}}$, SO11 |
| Interrupt | INTCSI00 | INTCSI01 | INTCSI10 | INTCSI11 |
| | Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected. | | | |
| Error detection flag | Overrun error detection flag (OVFmn) only | | | |
| Transfer data length | 7 to 16 bits | | | |
| Transfer rate | Max. $f_{MCK}$/6 [Hz]Notes 1, 2. | | | |
| Data phase | Selectable by the DAPmn bit of the SCRmn register<br>• DAPmn = 0: Data output starts from the start of the serial clock operation.<br>• DAPmn = 1: Data output starts half a clock before the start of the serial clock operation. | | | |
| Clock phase | Selectable by the CKPmn bit of the SCRmn register<br>• CKPmn = 0: Forward<br>• CKPmn = 1: Reverse | | | |
| Data direction | MSB or LSB first | | | |

**Notes 1.** Because the external serial clock input to the $\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$, and $\overline{\text{SCK11}}$ pins is sampled internally and used, the fastest transfer rate is $f_{MCK}$/6 [Hz].

**2.** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

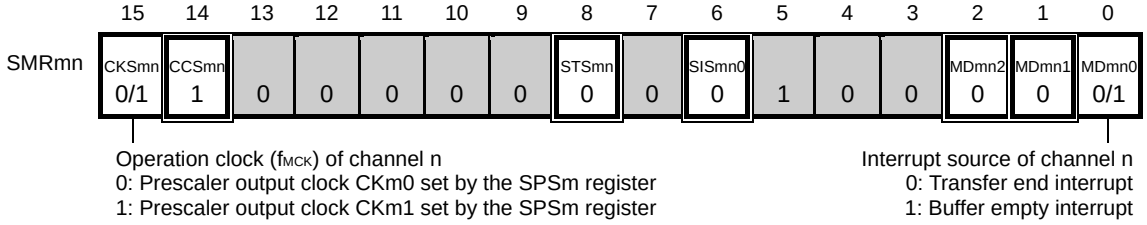
**Remarks 1.** $f_{MCK}$: Operation clock frequency of target channel

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

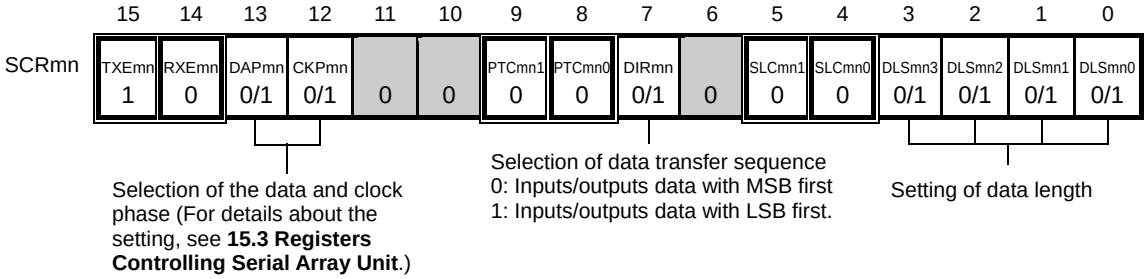
**(1) Register setting**

**Figure 15-49. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

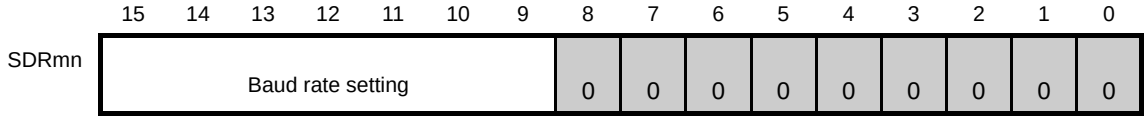



**(b) Serial communication operation setting register mn (SCRmn)**

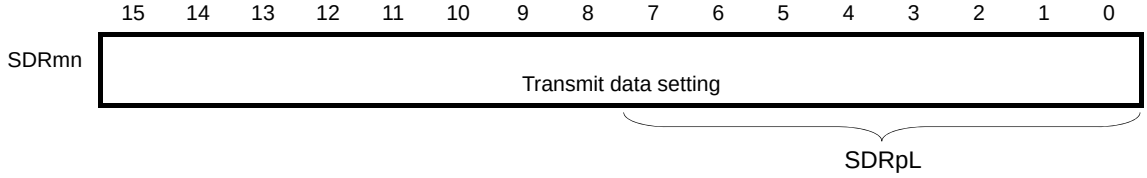



**(c) Serial data register mn (SDRmn)**

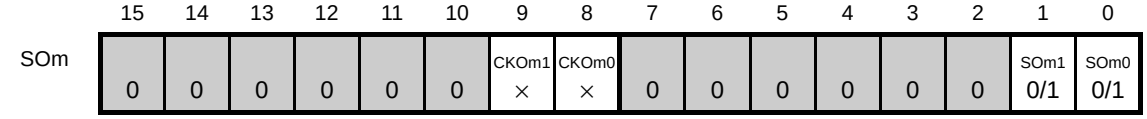
**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)**




**(d) Serial output register m (SOm) … Sets only the bits of the target channel.**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** □ : Setting is fixed in the CSI slave transmission mode

▒ : Setting disabled (set to the initial value)

× : Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1 : Set to 0 or 1 depending on the usage of the user

**Figure 15-49. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (2/2)**

**(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 0/1 | SOEm0 0/1 |

**(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1 0/1 | SSm0 0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** □ : Setting is fixed in the CSI slave transmission mode

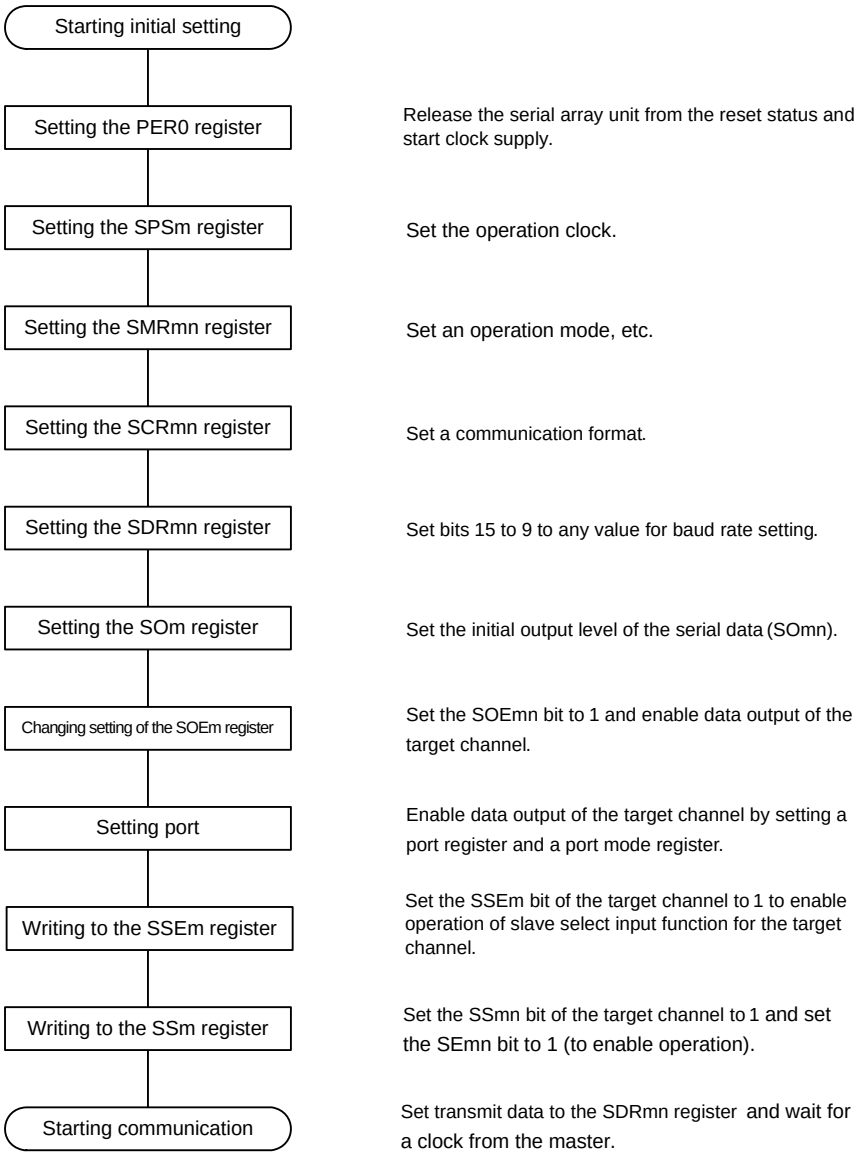
■ : Setting disabled (set to the initial value)

× : Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1 : Set to 0 or 1 depending on the usage of the user

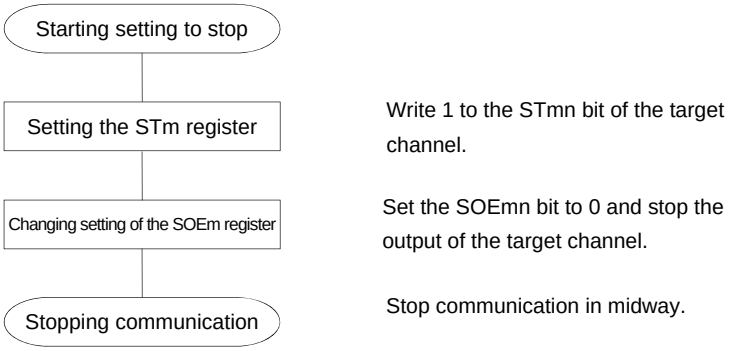
### (2) Operation procedure

**Figure 15-50. Initial Setting Procedure for Slave Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

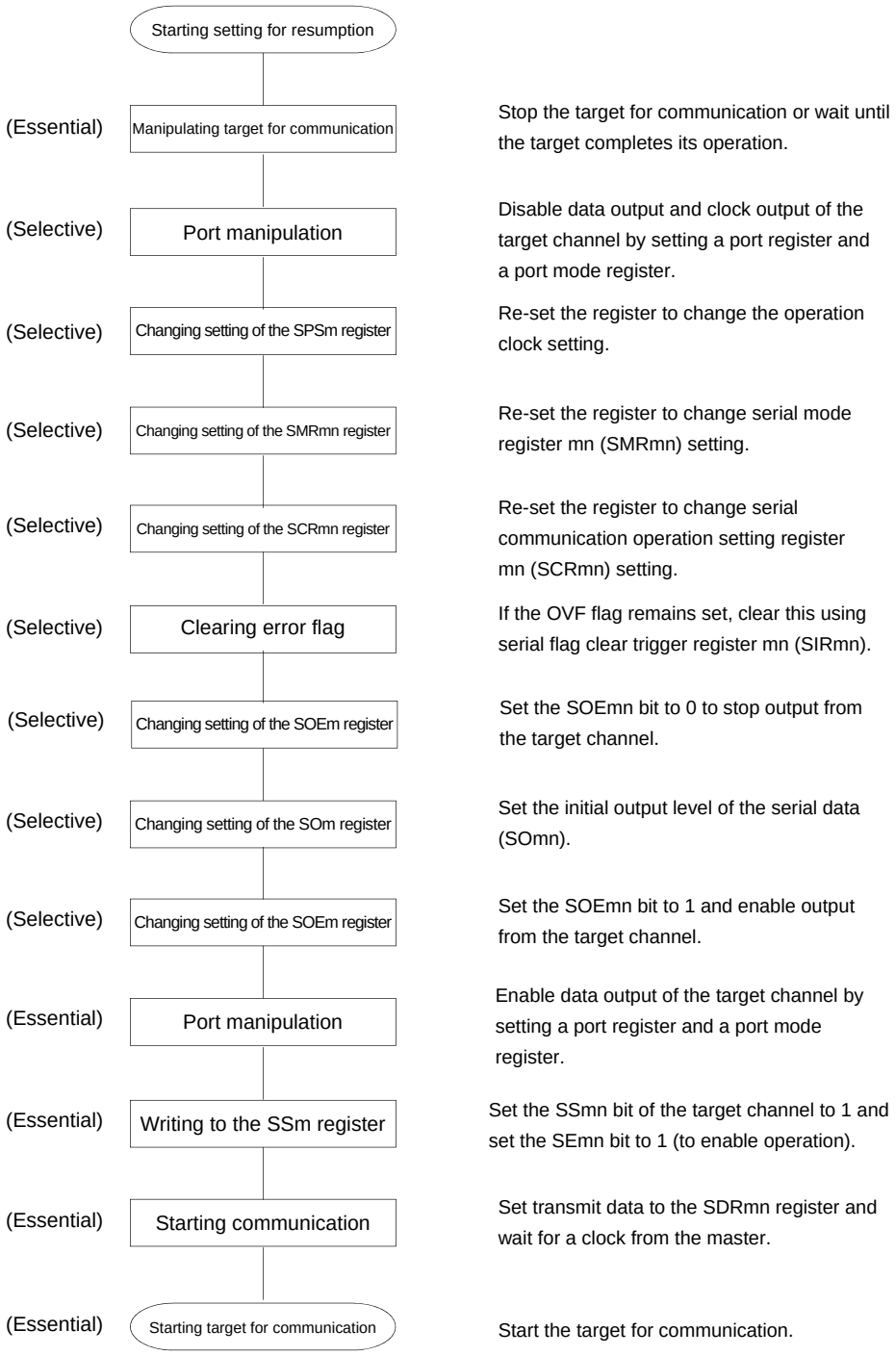
**Figure 15-51. Procedure for Stopping Slave Transmission**




**Remarks 1.** Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 15-52 Procedure for Resuming Slave Transmission**).

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

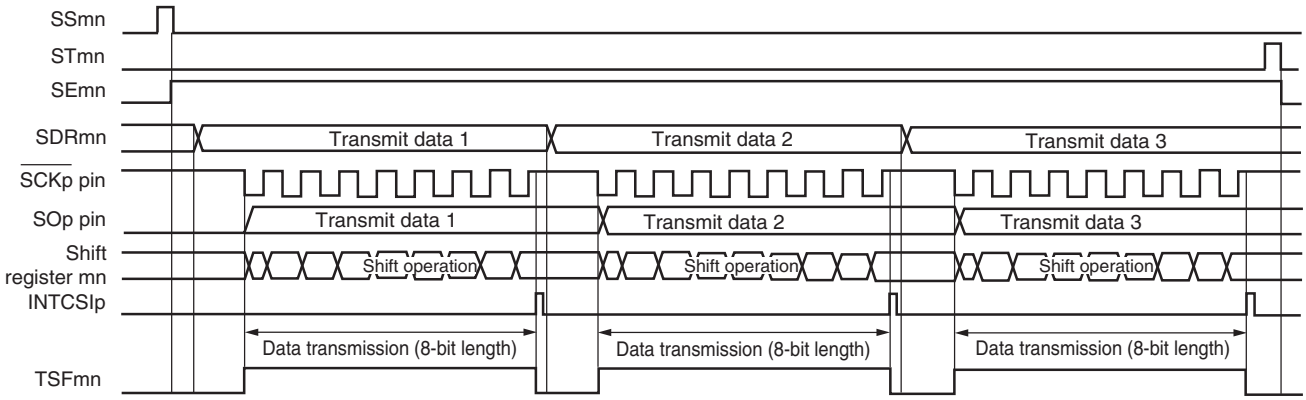
**Figure 15-52. Procedure for Resuming Slave Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

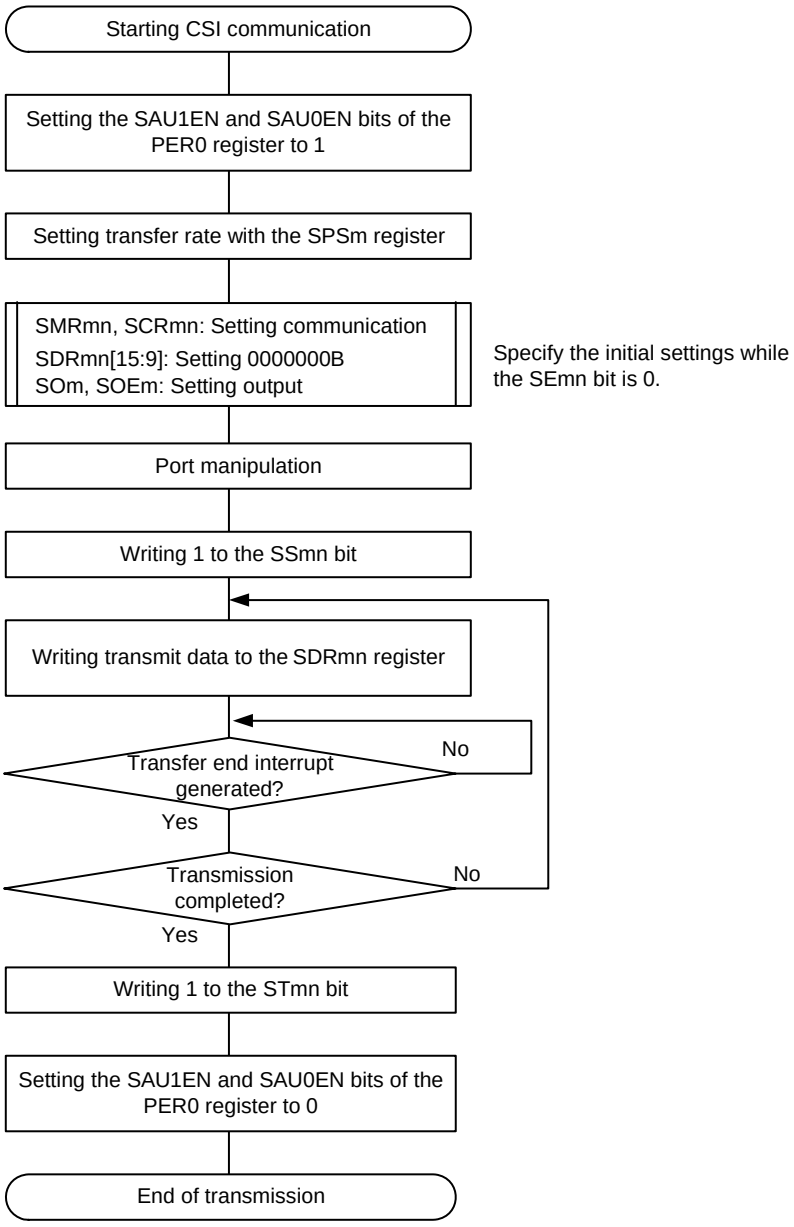
### (3) Processing flow (in single-transmission mode)

**Figure 15-53. Timing Chart of Slave Transmission (in Single-Transmission Mode)**
**(Type 1: DAPmn = 0, CKPmn = 0)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

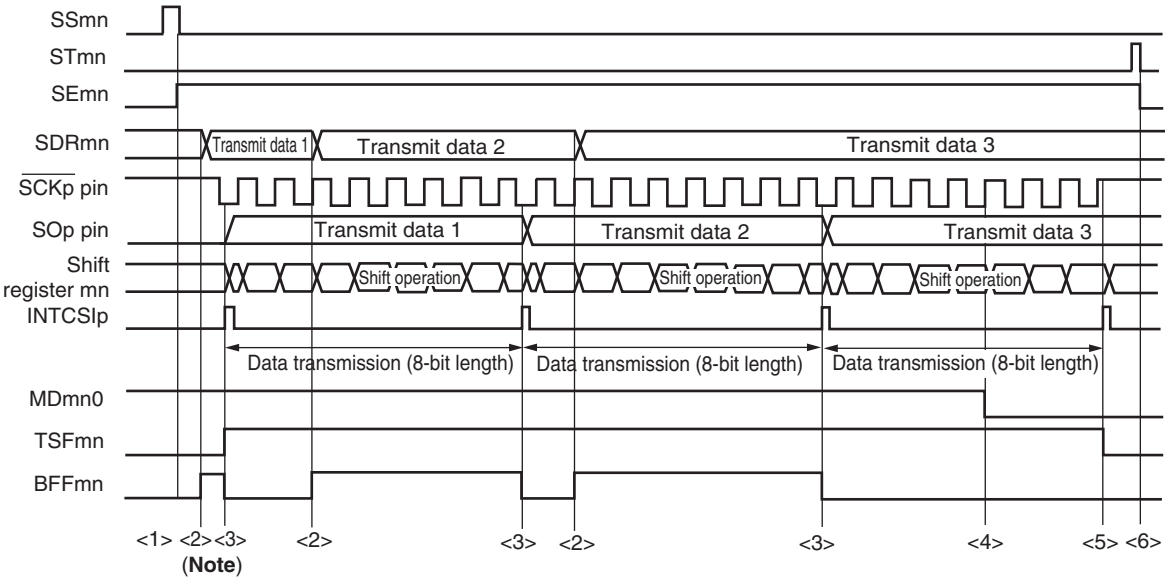
**Figure 15-54. Flowchart of Slave Transmission (in Single-Transmission Mode)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

### (4) Processing flow (in continuous transmission mode)

**Figure 15-55. Timing Chart of Slave Transmission (in Continuous Transmission Mode)**
**(Type 1: DAPmn = 0, CKPmn = 0)**

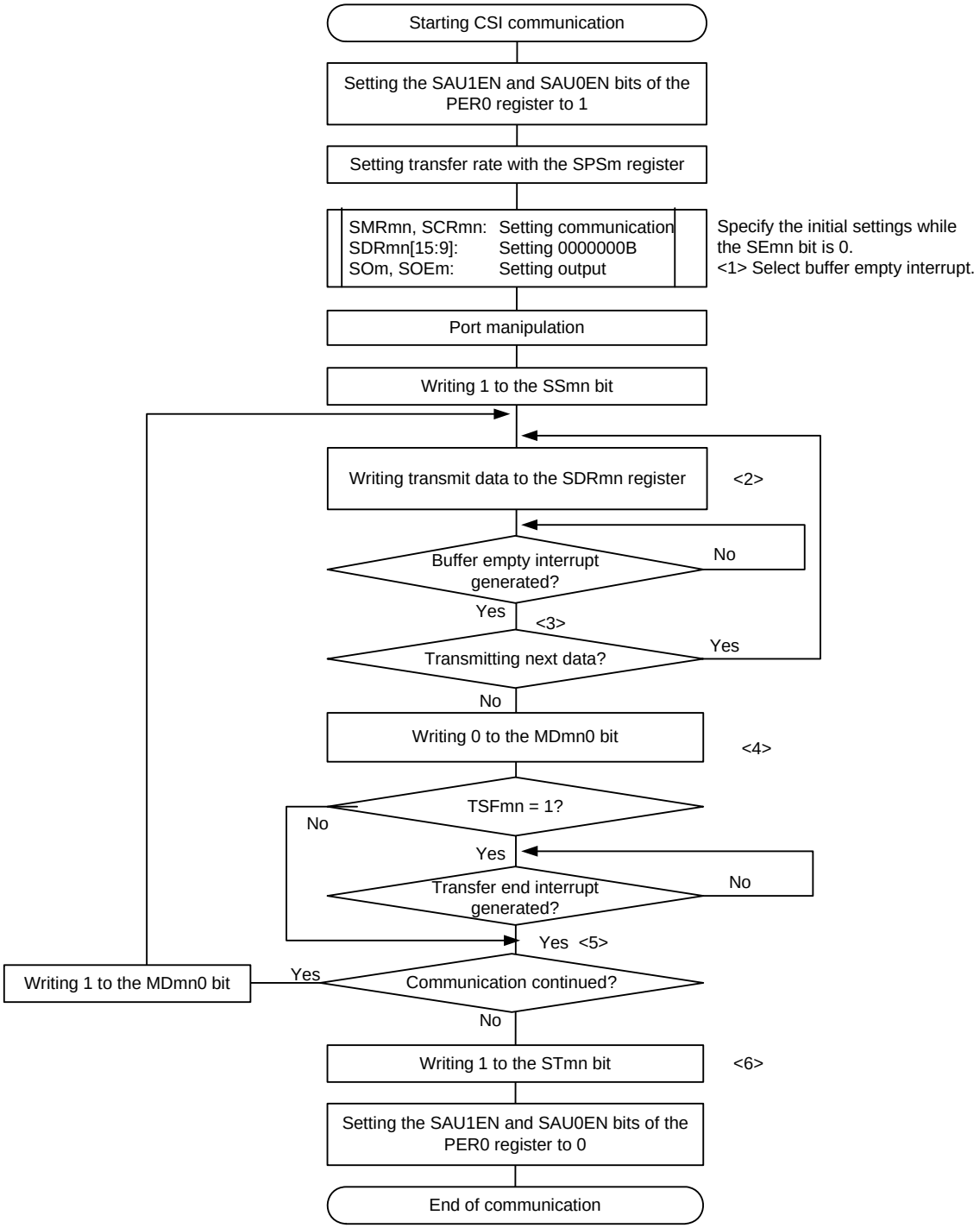



**Note** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

**Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-56. Flowchart of Slave Transmission (in Continuous Transmission Mode)**




**Remarks 1.** <1> to <6> in the figure correspond to <1> to <6> in **Figure 15-55 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

### 15.5.5 Slave reception

Slave reception is an operation wherein this MCU receives data from another device in the state of a transfer clock being input from another device.

| 3-Wire Serial I/O | CSI00 | CSI01 | CSI10 | CSI11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | $\overline{SCK00}$, SI00 | $\overline{SCK01}$, SI01 | $\overline{SCK10}$, SI10 | $\overline{SCK11}$, SI11 |
| Interrupt | INTCSI00 | INTCSI01 | INTCSI10 | INTCSI11 |
| | Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.) | | | |
| Error detection flag | Overrun error detection flag (OVFmn) only | | | |
| Transfer data length | 7 to 16 bits | | | |
| Transfer rate | Max. $f_{MCK}$/6 [Hz] Notes 1, 2 | | | |
| Data phase | Selectable by the DAPmn bit of the SCRmn register<br>• DAPmn = 0: Data input starts from the start of the serial clock operation.<br>• DAPmn = 1: Data input starts half a clock before the start of the serial clock operation. | | | |
| Clock phase | Selectable by the CKPmn bit of the SCRmn register<br>• CKPmn = 0: Forward<br>• CKPmn = 1: Reverse | | | |
| Data direction | MSB or LSB first | | | |

**Notes 1.** Because the external serial clock input to the $\overline{SCK00}$, $\overline{SCK01}$, $\overline{SCK10}$, and $\overline{SCK11}$ pins is sampled internally and used, the fastest transfer rate is $f_{MCK}$/6 [Hz].

**2.** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

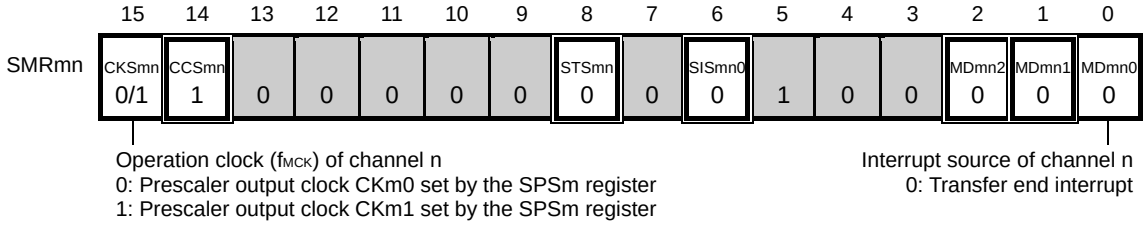
**Remarks 1.** $f_{MCK}$: Operation clock frequency of target channel

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

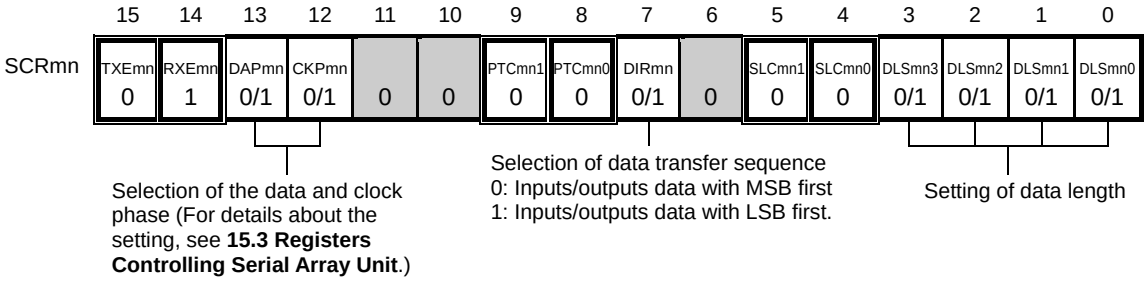
**(1) Register setting**

**Figure 15-57. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

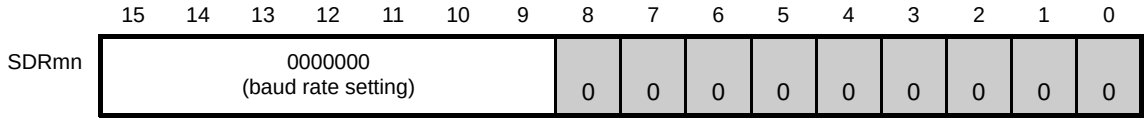



**(b) Serial communication operation setting register mn (SCRmn)**

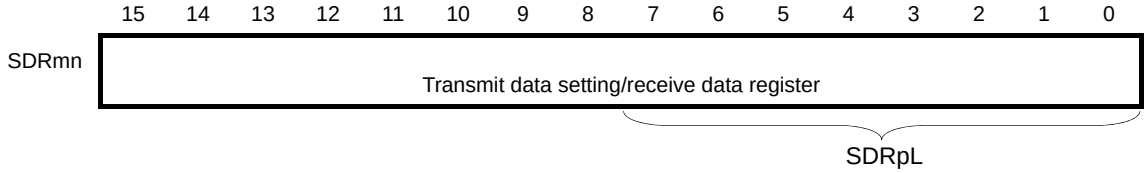



**(c) Serial data register mn (SDRmn)**

**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI slave reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-57. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (2/2)**

**(d) Serial output register m (SOm) …The register that not used in this mode.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOm | 0 | 0 | 0 | 0 | 0 | 0 | CKOm1 × | CKOm0 × | 0 | 0 | 0 | 0 | 0 | 0 | SOm1 × | SOm0 × |

**(e) Serial output enable register m (SOEm) …The register that not used in this mode.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 × | SOEm0 × |

**(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1 0/1 | SSm0 0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** □: Setting is fixed in the CSI slave reception mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

### (2) Operation procedure

**Figure 15-58. Initial Setting Procedure for Slave Reception**

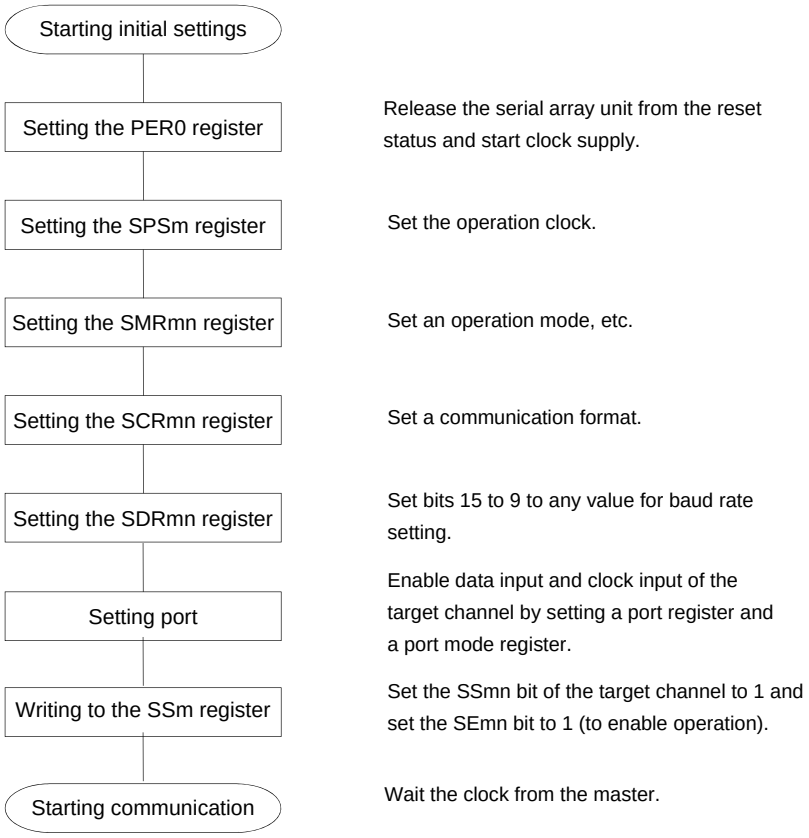
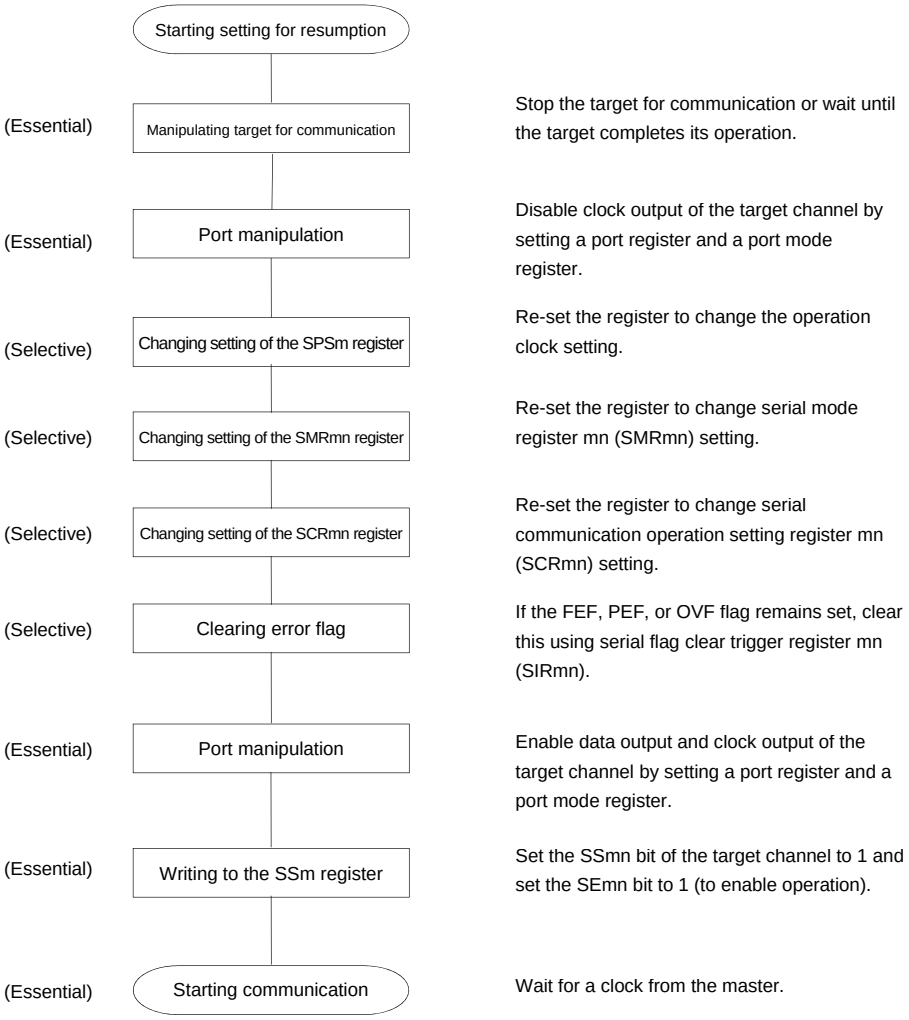



**Figure 15-59. Procedure for Stopping Slave Reception**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

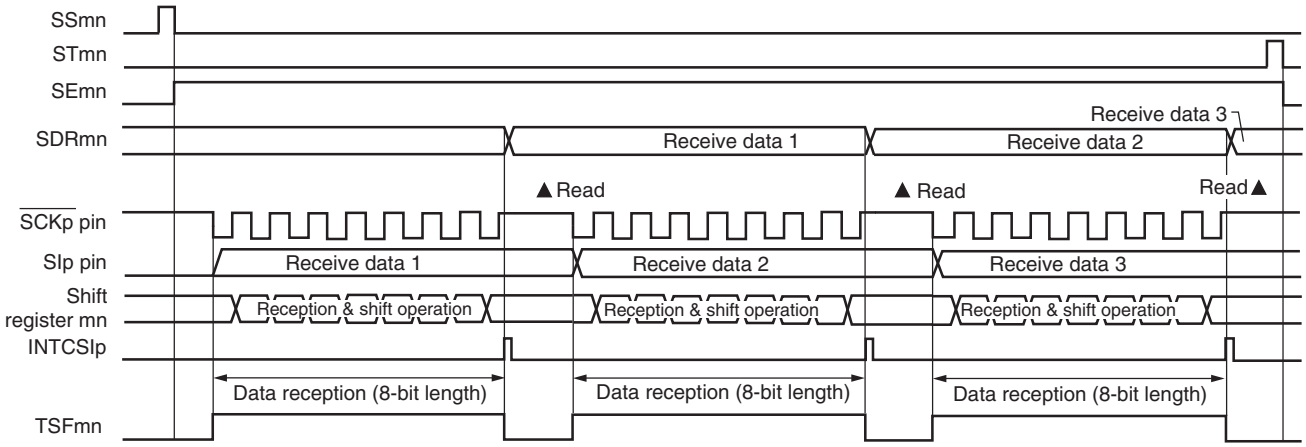
**Figure 15-60. Procedure for Resuming Slave Reception**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

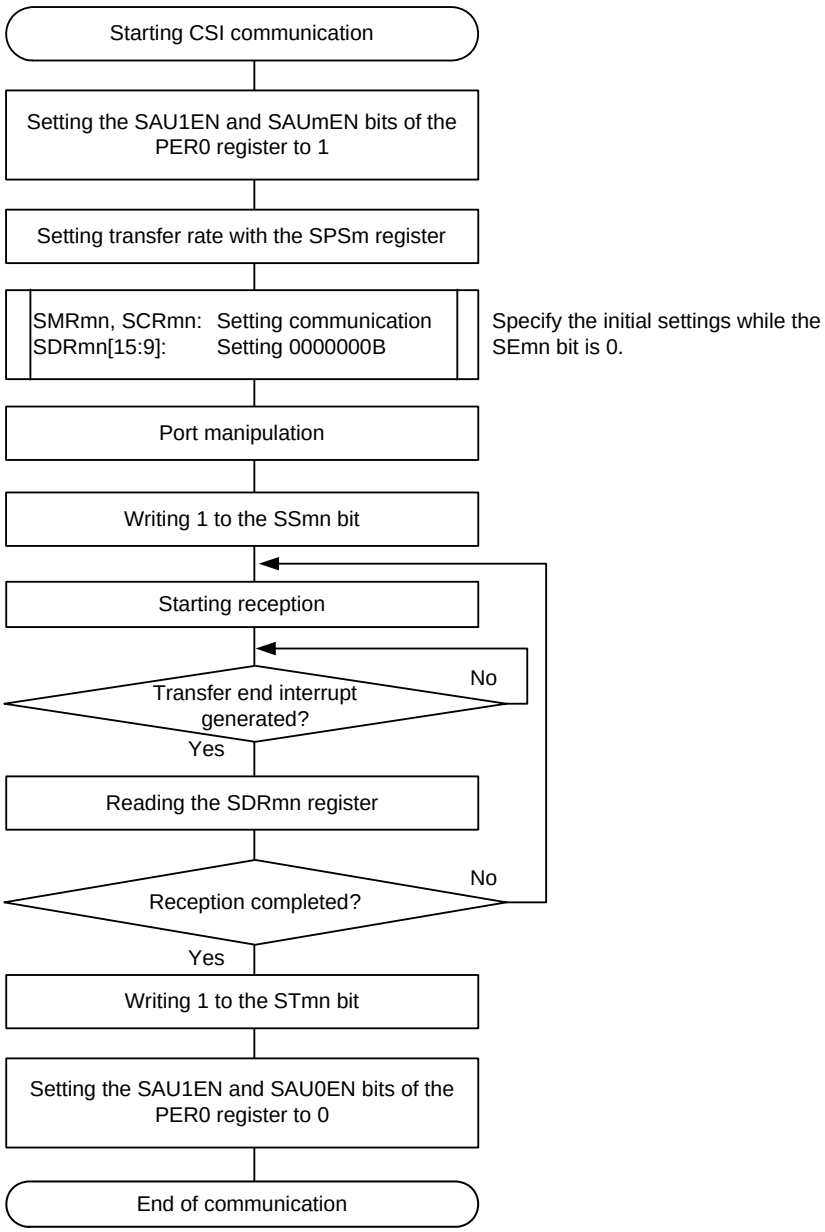
### (3) Processing flow (in single-reception mode)

**Figure 15-61. Timing Chart of Slave Reception (in Single-Reception Mode)**
**(Type 1: DAPmn = 0, CKPmn = 0)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-62. Flowchart of Slave Reception (in Single-Reception Mode)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### 15.5.6 Slave transmission/reception

Slave transmission/reception is an operation wherein this MCU transmits/receives data to/from another device in the state of a transfer clock being input from another device.

| 3-Wire Serial I/O | CSI00 | CSI01 | CSI10 | CSI11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | $\overline{\text{SCK00}}$, SI00, SO00 | $\overline{\text{SCK01}}$, SI01, SO01 | $\overline{\text{SCK10}}$, SI10, SO10 | $\overline{\text{SCK11}}$, SI11, SO11 |
| Interrupt | INTCSI00 | INTCSI01 | INTCSI10 | INTCSI11 |
| | Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected. | | | |
| Error detection flag | Overrun error detection flag (OVFmn) only | | | |
| Transfer data length | 7 to 16 bits | | | |
| Transfer rate | Max. $f_{MCK}$/6 [Hz]Notes 1, 2. | | | |
| Data phase | Selectable by the DAPmn bit of the SCRmn register<br>• DAPmn = 0: Data I/O starts from the start of the serial clock operation.<br>• DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation. | | | |
| Clock phase | Selectable by the CKPmn bit of the SCRmn register<br>• CKPmn = 0: Forward<br>• CKPmn = 1: Reverse | | | |
| Data direction | MSB or LSB first | | | |

**Notes 1.** Because the external serial clock input to the $\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$, and $\overline{\text{SCK11}}$ pins is sampled internally and used, the fastest transfer rate is $f_{MCK}$/6 [Hz].

**2.** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

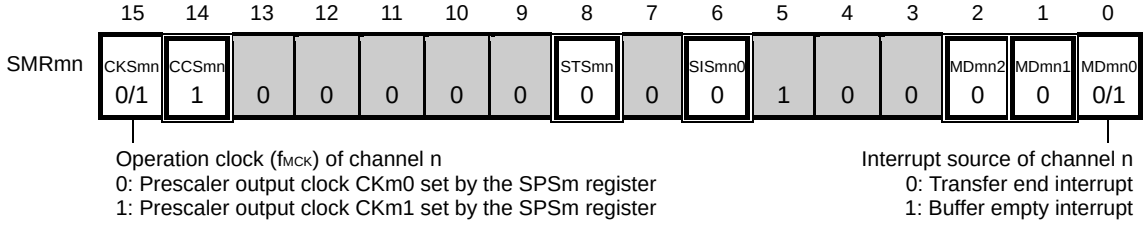
**Remarks 1.** $f_{MCK}$: Operation clock frequency of target channel

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

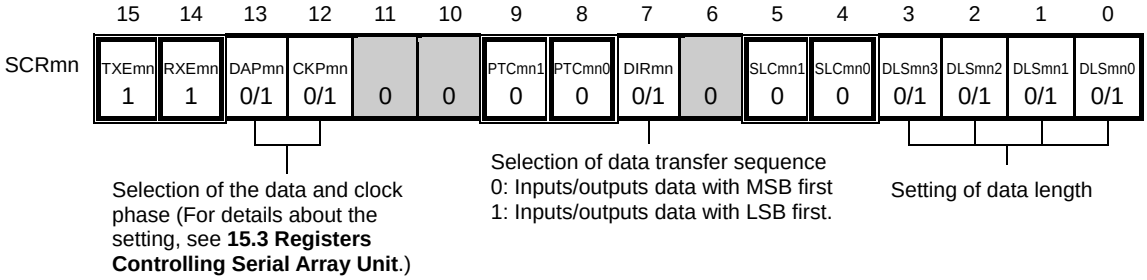
**(1) Register setting**

**Figure 15-63. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

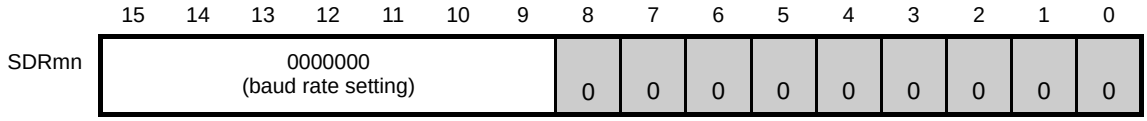



**(b) Serial communication operation setting register mn (SCRmn)**

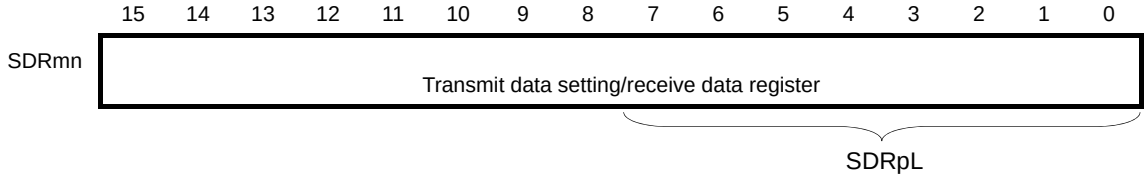



**(c) Serial data register mn (SDRmn)**

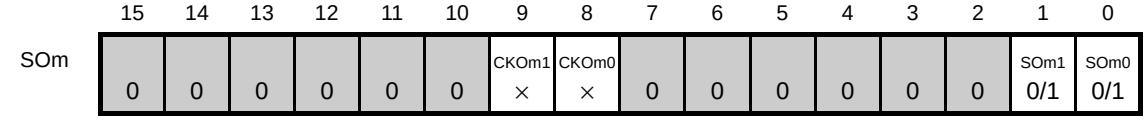
**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)**




**(d) Serial output register m (SOm) … Sets only the bits of the target channel.**




**Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** □: Setting is fixed in the CSI slave transmission/reception mode

■: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-63. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1<br>0/1 | SOEm0<br>0/1 |

**(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1<br>0/1 | SSm0<br>0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** □: Setting is fixed in the CSI slave transmission/reception mode

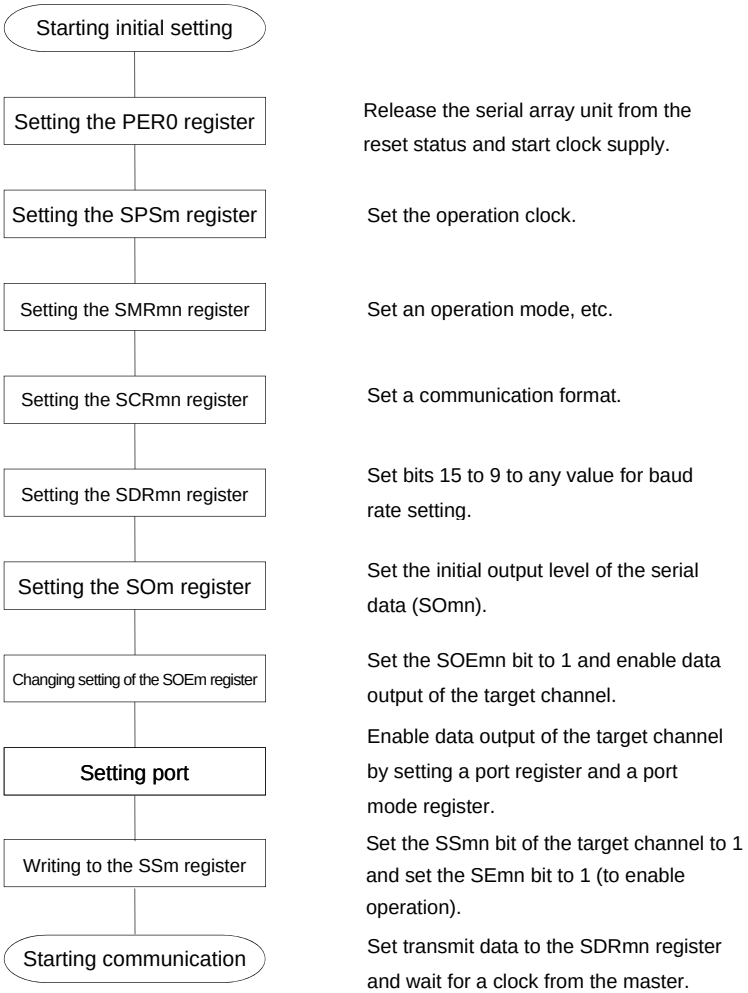
■: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

### (2) Operation procedure

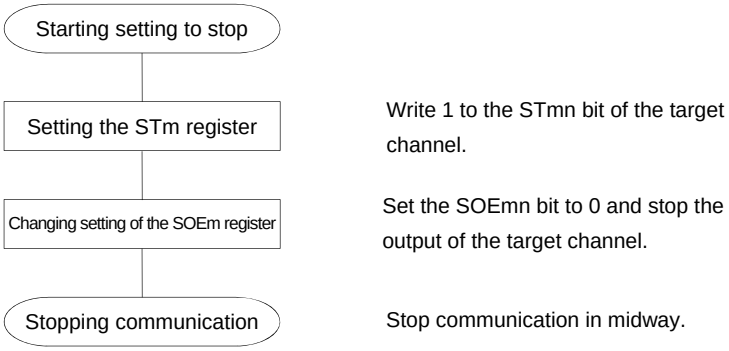
**Figure 15-64. Initial Setting Procedure for Slave Transmission/Reception**




**Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

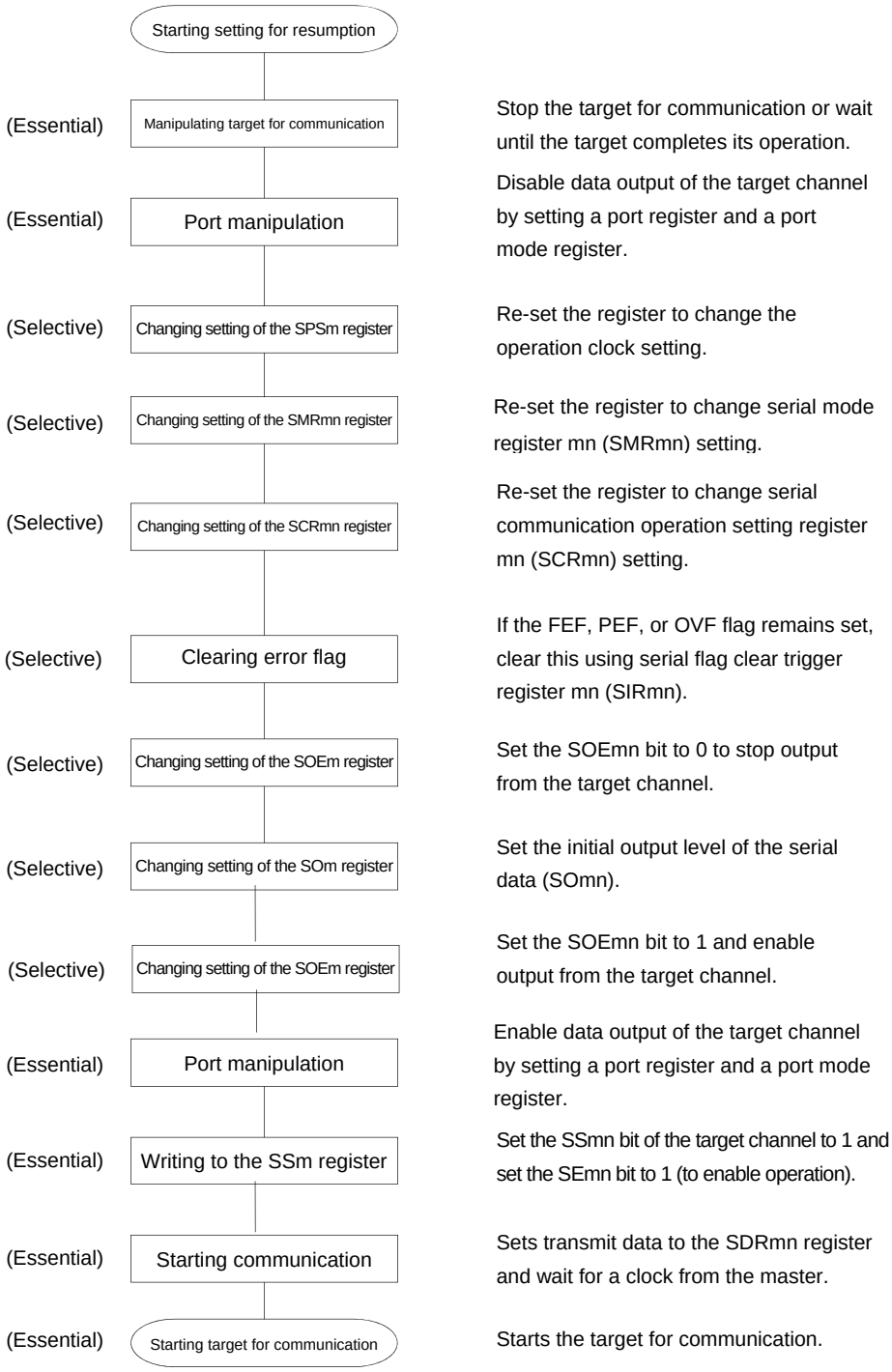
**Figure 15-65. Procedure for Stopping Slave Transmission/Reception**




**Remarks 1.** Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 15-66 Procedure for Resuming Slave Transmission/Reception**).

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-66. Procedure for Resuming Slave Transmission/Reception**

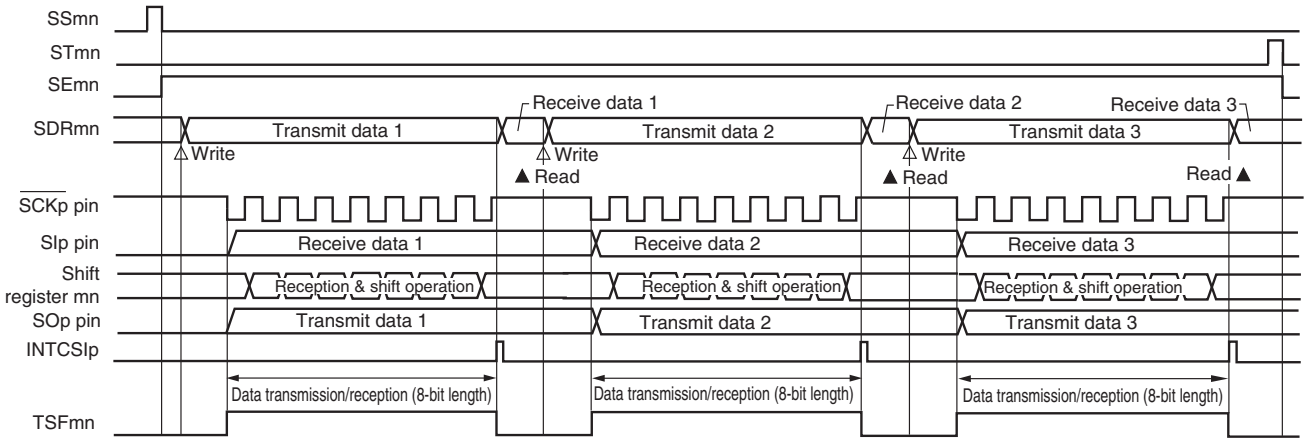



**Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

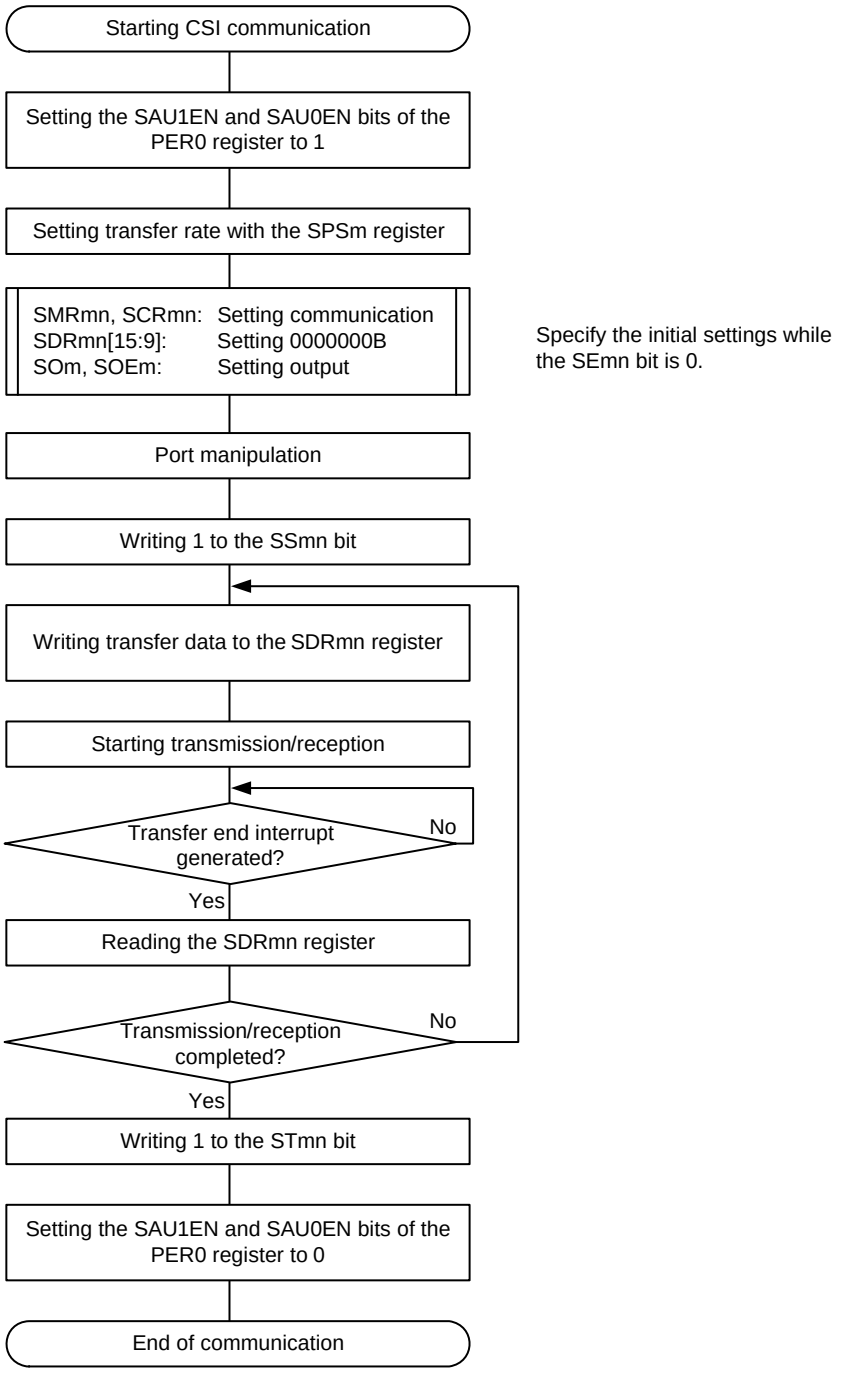
### (3) Processing flow (in single-transmission/reception mode)

**Figure 15-67. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-68. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)**

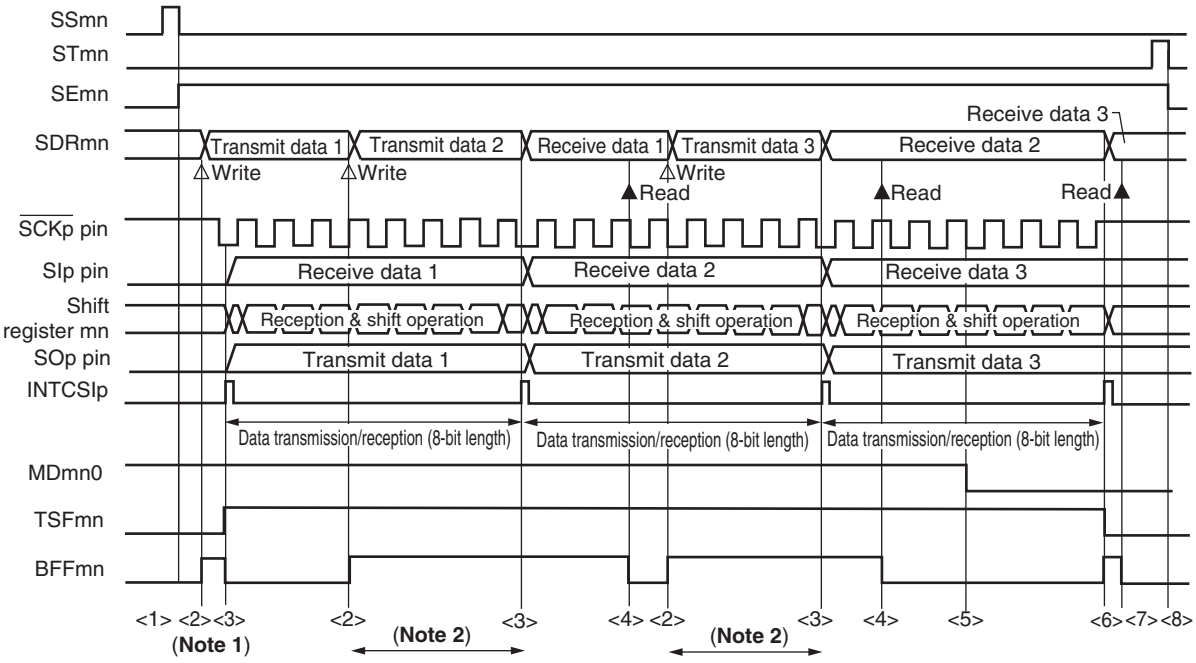



**Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

### (4) Processing flow (in continuous transmission/reception mode)

**Figure 15-69. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**




**Notes 1.** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

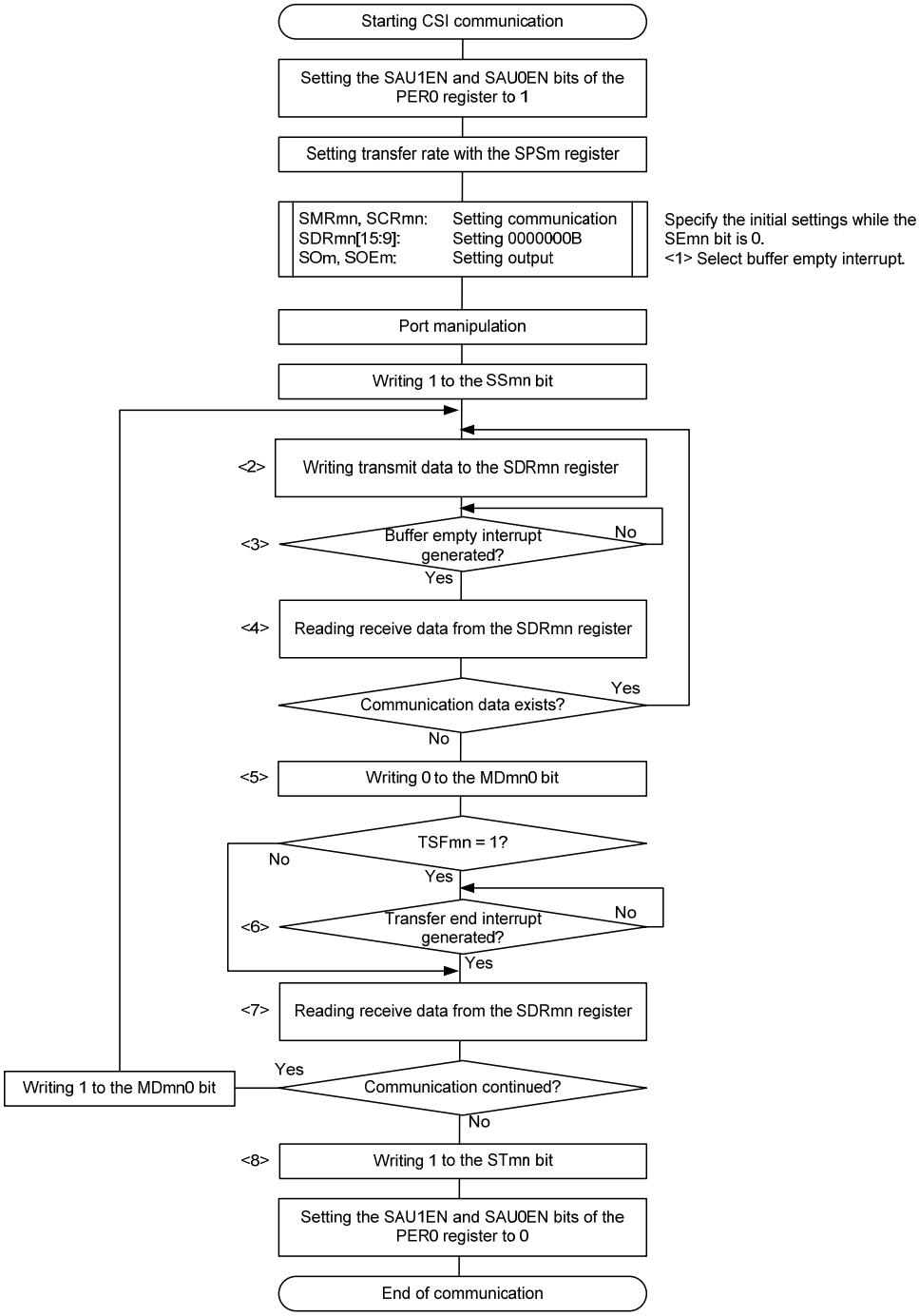
**2.** The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

**Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.**

**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-70 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-70. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**




**Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.**

**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-69 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

### 15.5.7 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11) communication can be calculated by the following expressions.

**(1) Master**

(Transfer clock frequency) = {Operation clock ($f_{MCK}$) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

**(2) Slave**

(Transfer clock frequency) = {Frequency of serial clock ($f_{SCK}$) supplied by master}[Note] [Hz]

**Note** The permissible maximum transfer clock frequency is $f_{MCK}$/6.

**Remark** The value of SDRmn[15:9] is the value of bits 15 to 9 of serial data register mn (SDRmn) (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock ($f_{MCK}$) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

**Table 15-2. Selection of Operation Clock For 3-Wire Serial I/O**

| SMRmn Register | SPSm Register | | | | | | | | Operation Clock ($f_{MCK}$) Note | |
|---|---|---|---|---|---|---|---|---|---|---|
| CKSmn | PRS m13 | PRS m12 | PRS m11 | PRS m10 | PRS m03 | PRS m02 | PRS m01 | PRS m00 | | $f_{CLK}$ = 32 MHz |
| 0 | X | X | X | X | 0 | 0 | 0 | 0 | $f_{CLK}$ | 32 MHz |
| | X | X | X | X | 0 | 0 | 0 | 1 | $f_{CLK}/2$ | 16 MHz |
| | X | X | X | X | 0 | 0 | 1 | 0 | $f_{CLK}/2^2$ | 8 MHz |
| | X | X | X | X | 0 | 0 | 1 | 1 | $f_{CLK}/2^3$ | 4 MHz |
| | X | X | X | X | 0 | 1 | 0 | 0 | $f_{CLK}/2^4$ | 2 MHz |
| | X | X | X | X | 0 | 1 | 0 | 1 | $f_{CLK}/2^5$ | 1 MHz |
| | X | X | X | X | 0 | 1 | 1 | 0 | $f_{CLK}/2^6$ | 500 kHz |
| | X | X | X | X | 0 | 1 | 1 | 1 | $f_{CLK}/2^7$ | 250 kHz |
| | X | X | X | X | 1 | 0 | 0 | 0 | $f_{CLK}/2^8$ | 125 kHz |
| | X | X | X | X | 1 | 0 | 0 | 1 | $f_{CLK}/2^9$ | 62.5 kHz |
| | X | X | X | X | 1 | 0 | 1 | 0 | $f_{CLK}/2^{10}$ | 31.25 kHz |
| | X | X | X | X | 1 | 0 | 1 | 1 | $f_{CLK}/2^{11}$ | 15.63 kHz |
| 1 | 0 | 0 | 0 | 0 | X | X | X | X | $f_{CLK}$ | 32 MHz |
| | 0 | 0 | 0 | 1 | X | X | X | X | $f_{CLK}/2$ | 16 MHz |
| | 0 | 0 | 1 | 0 | X | X | X | X | $f_{CLK}/2^2$ | 8 MHz |
| | 0 | 0 | 1 | 1 | X | X | X | X | $f_{CLK}/2^3$ | 4 MHz |
| | 0 | 1 | 0 | 0 | X | X | X | X | $f_{CLK}/2^4$ | 2 MHz |
| | 0 | 1 | 0 | 1 | X | X | X | X | $f_{CLK}/2^5$ | 1 MHz |
| | 0 | 1 | 1 | 0 | X | X | X | X | $f_{CLK}/2^6$ | 500 kHz |
| | 0 | 1 | 1 | 1 | X | X | X | X | $f_{CLK}/2^7$ | 250 kHz |
| | 1 | 0 | 0 | 0 | X | X | X | X | $f_{CLK}/2^8$ | 125 kHz |
| | 1 | 0 | 0 | 1 | X | X | X | X | $f_{CLK}/2^9$ | 62.5 kHz |
| | 1 | 0 | 1 | 0 | X | X | X | X | $f_{CLK}/2^{10}$ | 31.25 kHz |
| | 1 | 0 | 1 | 1 | X | X | X | X | $f_{CLK}/2^{11}$ | 15.63 kHz |
| Other than above | | | | | | | | | | Setting prohibited |

**Note** When changing the clock selected for $f_{CLK}$ (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 0003H) the operation of the serial array unit (SAU).

**Remarks 1.** X: Don't care

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### 15.5.8 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11) communication

The procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11) communication is described in Figure 15-71.

**Figure 15-71. Processing Procedure in Case of Overrun Error**

| Software Manipulation | Hardware Status | Remark |
|---|---|---|
| Reads serial data register mn (SDRmn). → | The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data. | This is to prevent an overrun error if the next reception is completed during error processing. |
| Reads serial status register mn (SSRmn). | | Error type is identified and the read value is used to clear error flag. |
| Writes 1 to serial flag clear trigger register mn (SIRmn). → | Error flag is cleared. | Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification. |

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

## 15.6 Clock Synchronous Serial Communication with SPI Function

All the channels (channels 0 and 1 of SAU0 and channels 0 and 1 of SAU1) correspond to the clock synchronous serial communication with SPI function.

[Data transmission/reception]

- Data length of 7 to 16 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- Level setting of transmit/receive data

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

[Expansion function]

- Slave select function

- Group A products

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note2 | | IIC01 |

- Products of Groups C-1 and D-1

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note2 | | IIC01 |
| 1 | 0 | CSI10 (supporting SPI function) Note1, 2 | UART1 | IIC10 |
| | 1 | - | | - |

- Products of Groups B, C-2, D-2, and E

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note2 | | IIC01 |
| 1 | 0 | CSI10 (supporting SPI function) Note1, 2 | UART1 | IIC10 |
| | 1 | CSI11 (supporting SPI function) Note2 | | IIC11 |

**Notes 1.** 48-pin, 32-pin and 30-pin products do not have $\overline{SSI10}$ pin.

<R> **2.** Set CKPmn bit of SCRmn register to 1, when SSEmn = 1 (Enables $\overline{SSImn}$ pin input).
(m = 0, 1, n = 0, 1)

**Remark** Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory
Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory
Group C-1: RL78/F13 (CAN and LIN incorporated) products with 30 or 32 pins
Group C-2: RL78/F13 (CAN and LIN incorporated) products with 48, 64, or 80 pins
Group D-1: RL78/F14 products with 30 or 32 pins
Group D-2: RL78/F14 products with 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory
Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

SPI function performs the following three types of communication operations.

- Master transmission (See **15.6.1 Master transmission**.)
- Master reception (See **15.6.2 Master reception**.)
- Master transmission/reception (See **15.6.3 Master transmission/reception**.)
- Slave transmission (See **15.6.4 Slave transmission**.)
- Slave reception (See **15.6.5 Slave reception**.)
- Slave transmission/reception (See **15.6.6 Slave transmission/reception**.)

Multiple slaves can be connected to a master and communication can be performed by using the slave select input function. The master outputs a slave select signal to the slave (one) that is the other party of communication, and each slave judges whether it has been selected as the other party of communication and controls the SO pin output. When a slave is selected, the SO pin is set to output state and transmit data can be communicated from the SO pin to the master. When a slave is not selected, the SO pin is set to Hi-Z state and prevents the short circuit with the output from the SO pin of other slaves. Therefore, in an environment where multiple slaves are connected, it is necessary set the SO pin to N-ch open-drain and pull up the node. Furthermore, when a slave is not selected, no transmission/reception operation is performed even if a serial clock is input from the master.

**Caution Output the slave select signal by port manipulation.**

**Figure 15-72. Example of Slave Select Input Function Configuration**

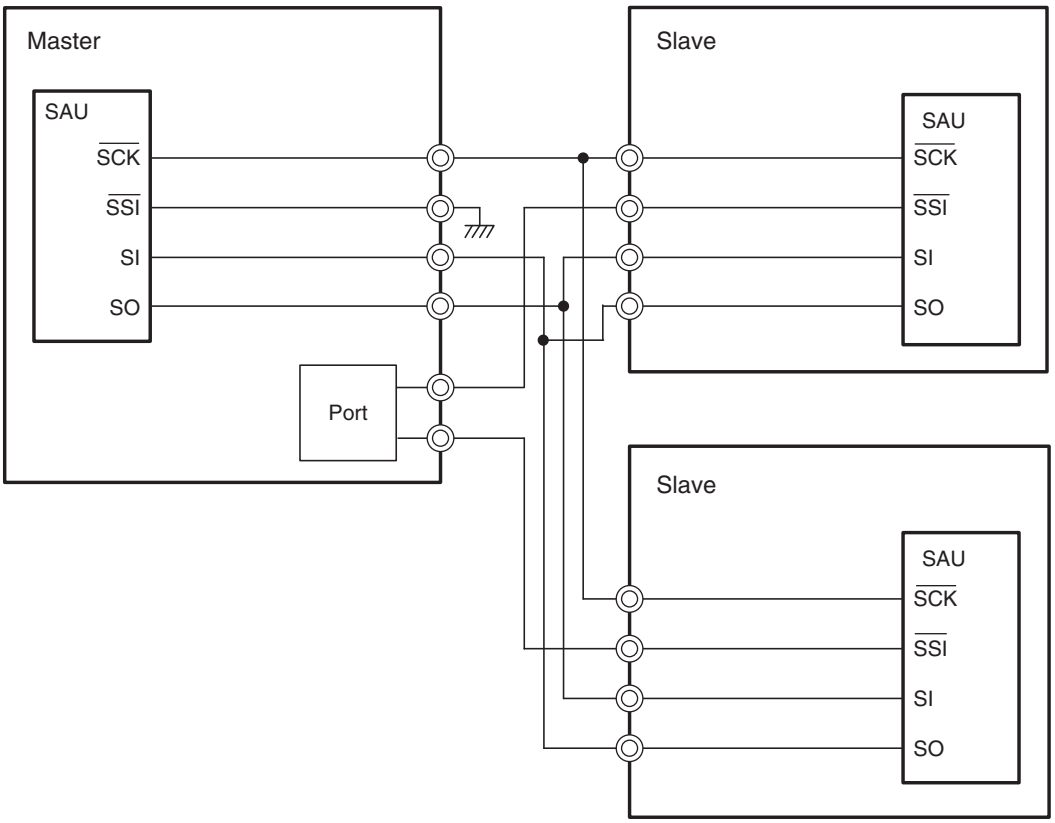
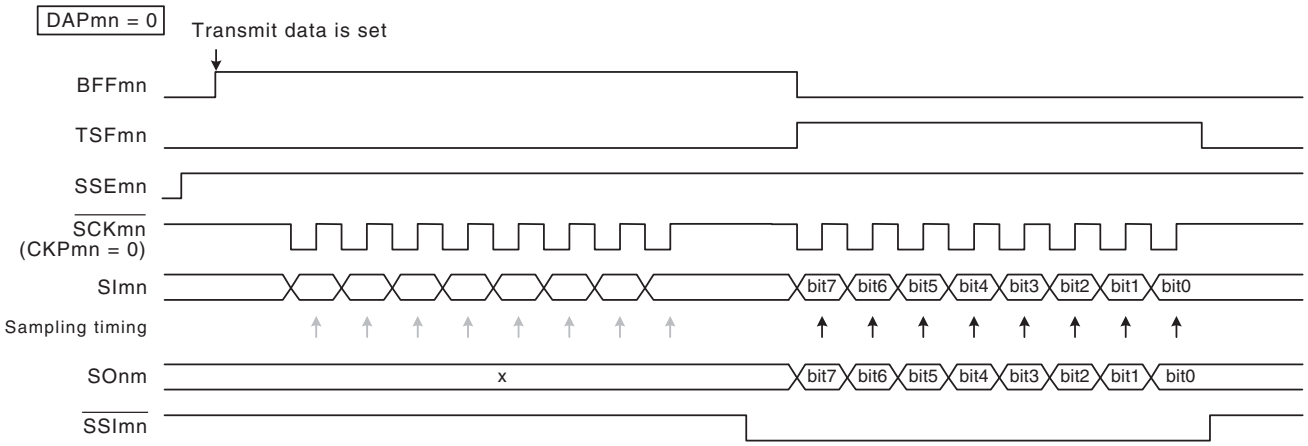
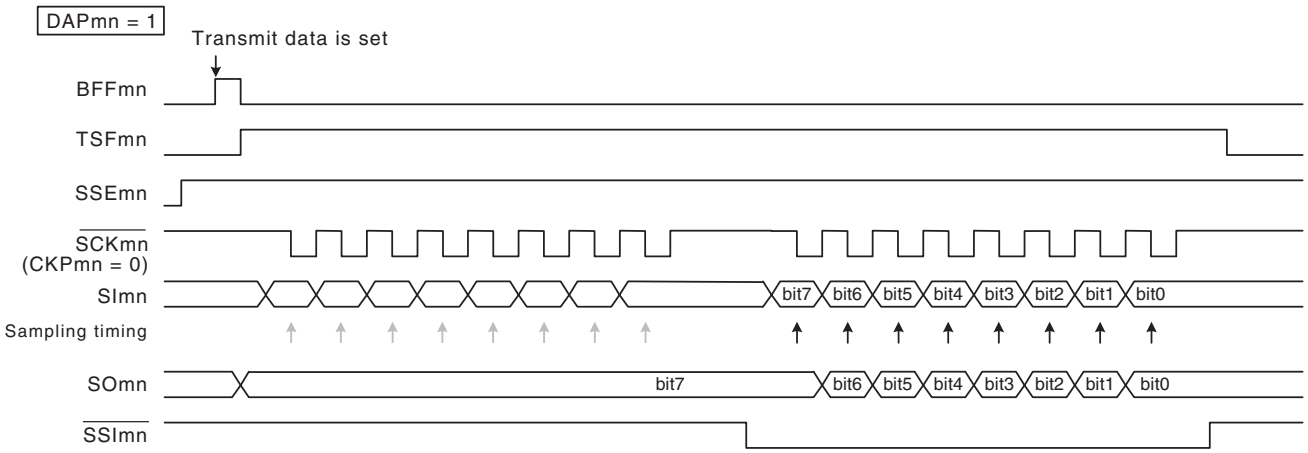


**Figure 15-73. Slave Select Input Function Timing Diagram**




While $\overline{\text{SSImn}}$ is at high level, transmission is not performed even if the falling edge of $\overline{\text{SCKmn}}$ (serial clock) arrives, and neither is receive data sampled in synchronization with the rising edge.
When $\overline{\text{SSImn}}$ goes to low level, data is output (shifted) in synchronization with the falling edge of the serial clock and a reception operation is performed in synchronization with the rising edge.




If DAPmn = 1, when transmit data is set while $\overline{\text{SSImn}}$ is at high level, the first data (bit 7) is output to the data output. However, no shift operation is performed even if the rising edge of $\overline{\text{SCKmn}}$ (serial clock) arrives, and neither is receive data sampled in synchronization with the falling edge. When $\overline{\text{SSImn}}$ goes to low level, data is output (shifted) in synchronization with the next rising edge and a reception operation is performed in synchronization with the falling edge.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

### 15.6.1 Master transmission

Master transmission is an operation wherein this MCU outputs a transfer clock and transmits data to another device.

| SPI Function | CSI00 | CSI01 | CSI10 | CSI11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | $\overline{\text{SCK00}}$, SO00 | $\overline{\text{SCK01}}$, SO01 | $\overline{\text{SCK10}}$, SO10 | $\overline{\text{SCK11}}$, SO11 |
| Interrupt | INTCSI00 | INTCSI01 | INTCSI10 | INTCSI11 |
| | Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected. | | | |
| Error detection flag | None | | | |
| Transfer data length | 7 to 16 bits | | | |
| Transfer rate | Max. $f_{MCK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] **Note** $f_{CLK}$: System clock frequency | | | |
| Data phase | Selectable by the DAPmn bit of the SCRmn register<br>• DAPmn = 0: Data output starts from the start of the serial clock operation.<br>• DAPmn = 1: Data output starts half a clock before the start of the serial clock operation. | | | |
| Clock phase | Selectable by the CKPmn bit of the SCRmn register<br>• CKPmn = 0: Forward<br>• CKPmn = 1: Reverse | | | |
| Data direction | MSB or LSB first | | | |

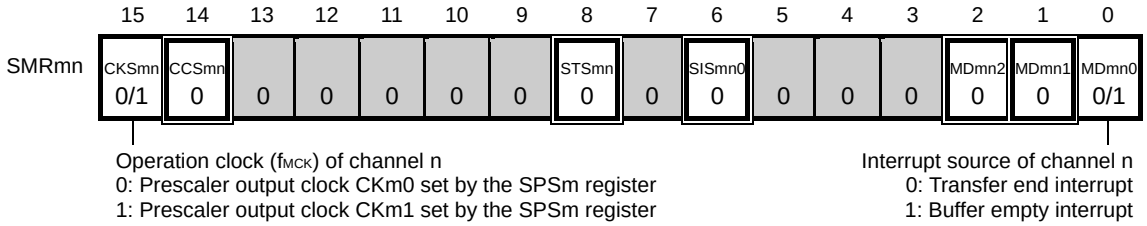
**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

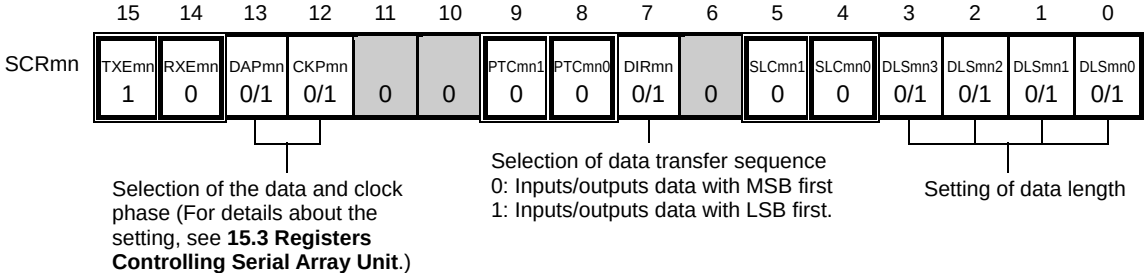
### (1) Register setting

**Figure 15-74. Example of Contents of Registers for Master Transmission of SPI Function (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

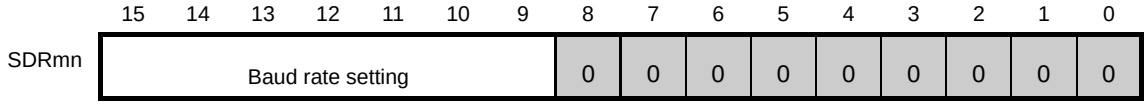



**(b) Serial communication operation setting register mn (SCRmn)**

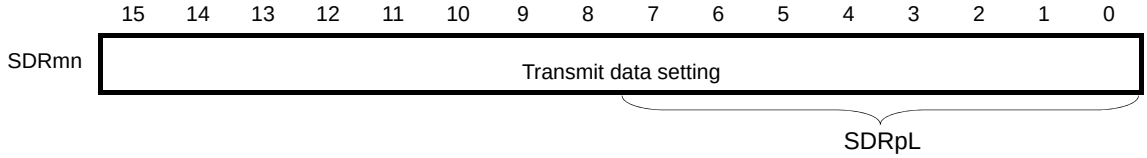



**(c) Serial data register mn (SDRmn)**

**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-74. Example of Contents of Registers for Master Transmission of SPI Function (CSI00, CSI01, CSI10, CSI11) (2/2)**

**(d) Serial output register m (SOm) … Sets only the bits of the target channel.**

SOm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | CKOm1 0/1 | CKOm0 0/1 | 0 | 0 | 0 | 0 | 0 | 0 | SOm1 0/1 | SOm0 0/1 |

Communication starts when these bits are 1 if the clock phase is forward (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

**(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.**

SOEm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 0/1 | SOEm0 0/1 |

**(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

SSm

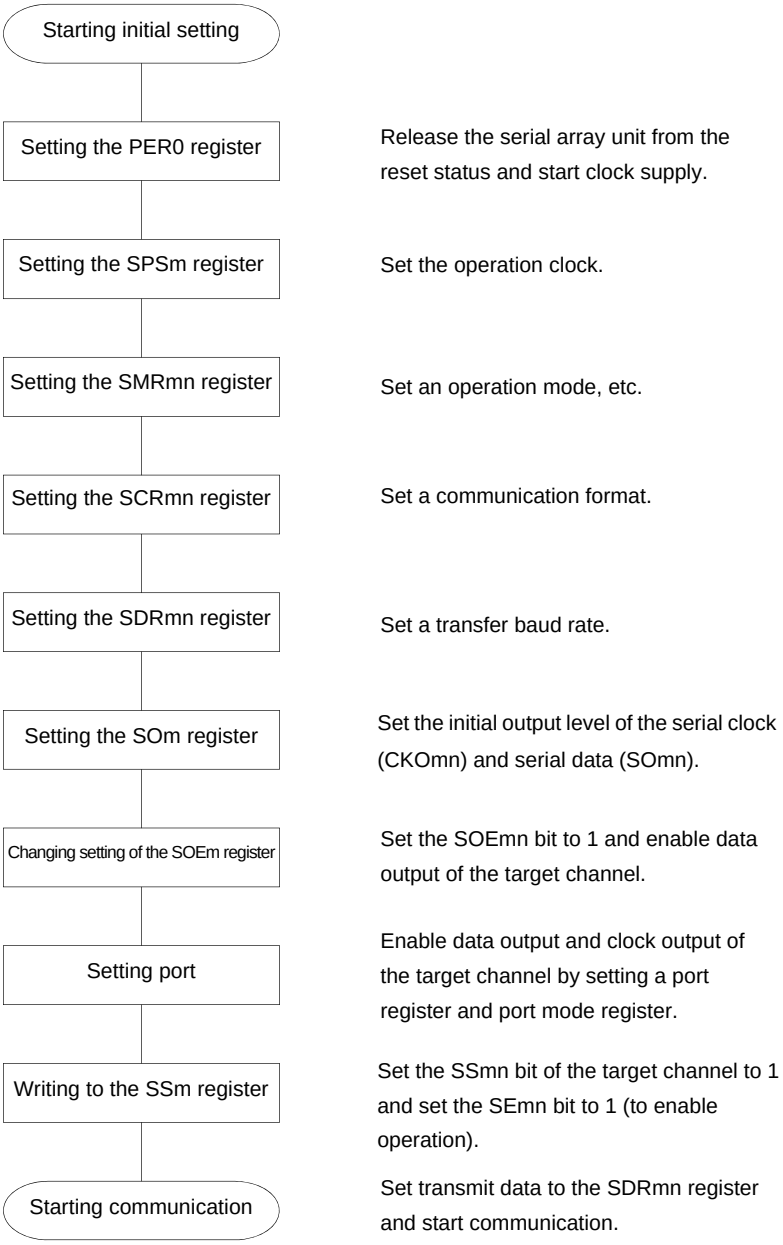
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1 0/1 | SSm0 0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

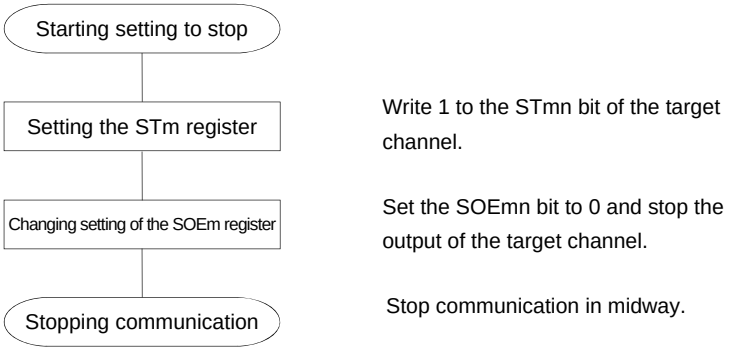
### (2) Operation procedure

**Figure 15-75. Initial Setting Procedure for Master Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

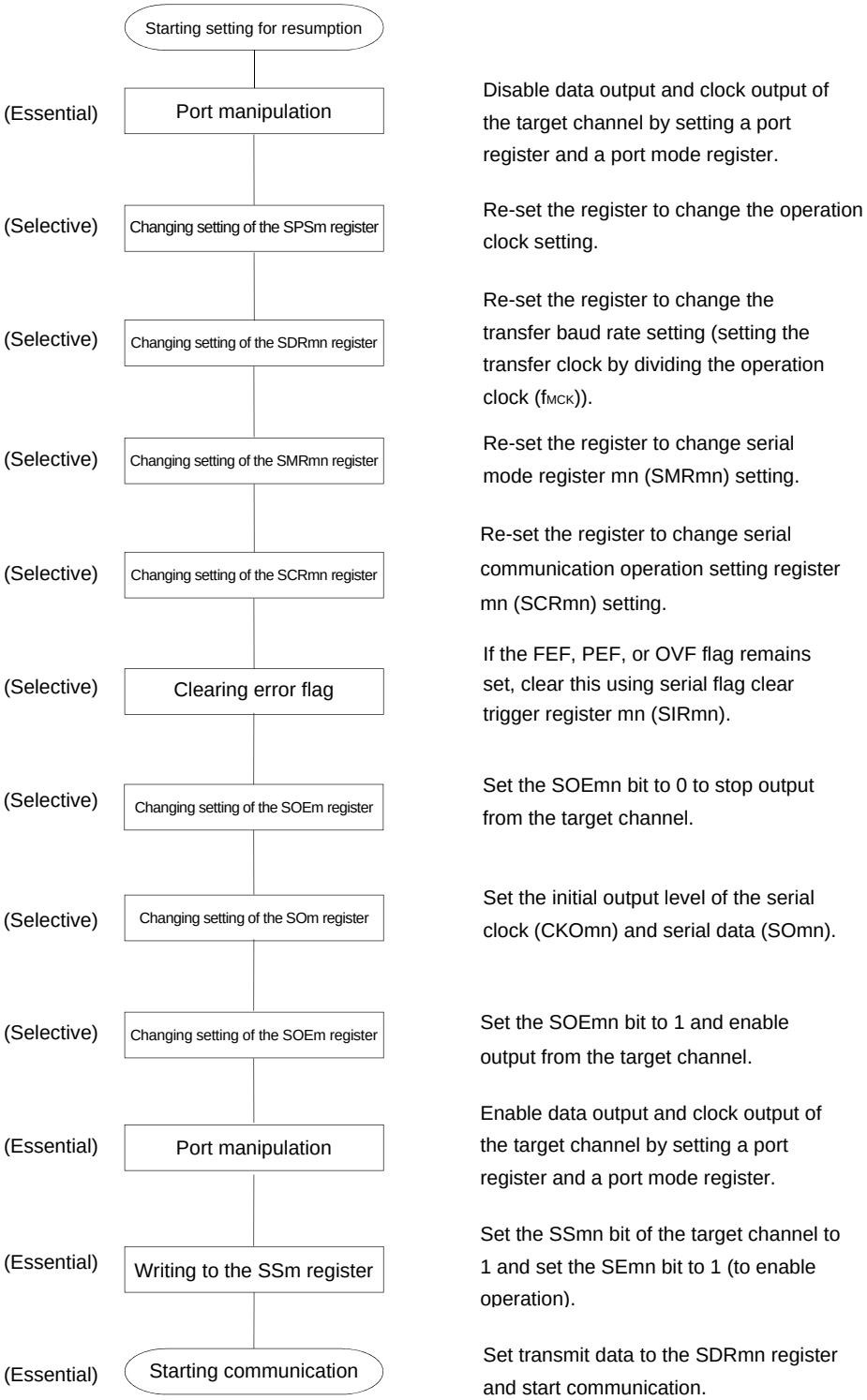
**Figure 15-76. Procedure for Stopping Master Transmission**




**Remarks 1.** Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 15-77 Procedure for Resuming Master Transmission**).

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

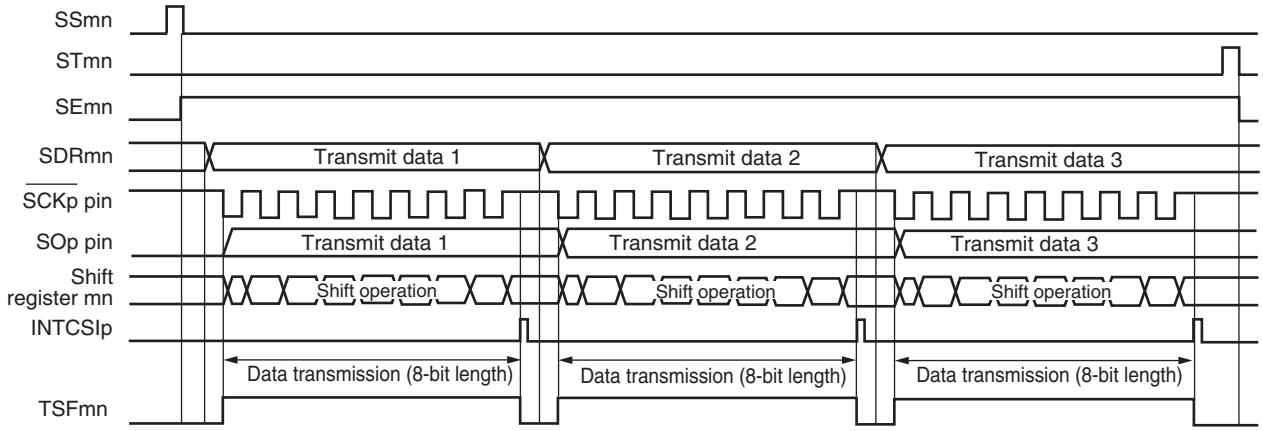
**Figure 15-77. Procedure for Resuming Master Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

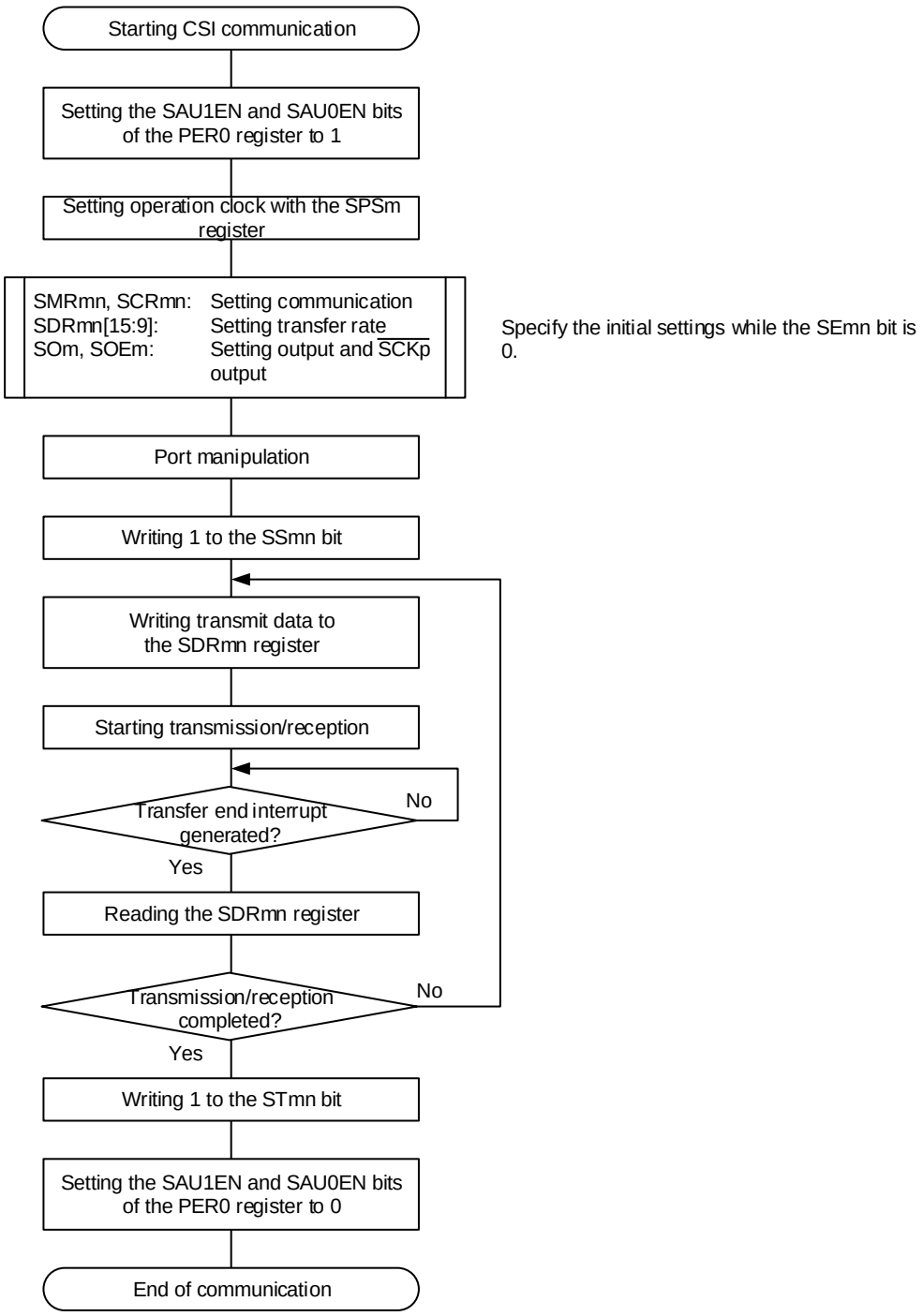
### (3) Processing flow (in single-transmission mode)

**Figure 15-78. Timing Chart of Master Transmission (in Single-Transmission Mode)**
**(Type 1: DAPmn = 0, CKPmn = 0)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

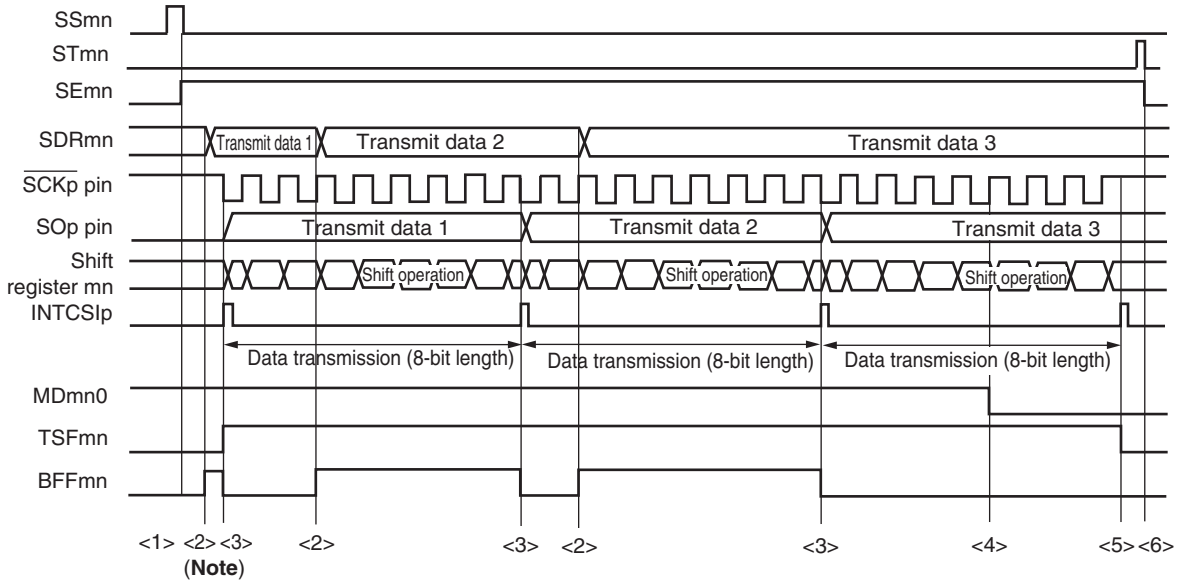
**Figure 15-79. Flowchart of Master Transmission (in Single-Transmission Mode)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### (4) Processing flow (in continuous transmission mode)

**Figure 15-80. Timing Chart of Master Transmission (in Continuous Transmission Mode)**
**(Type 1: DAPmn = 0, CKPmn = 0)**

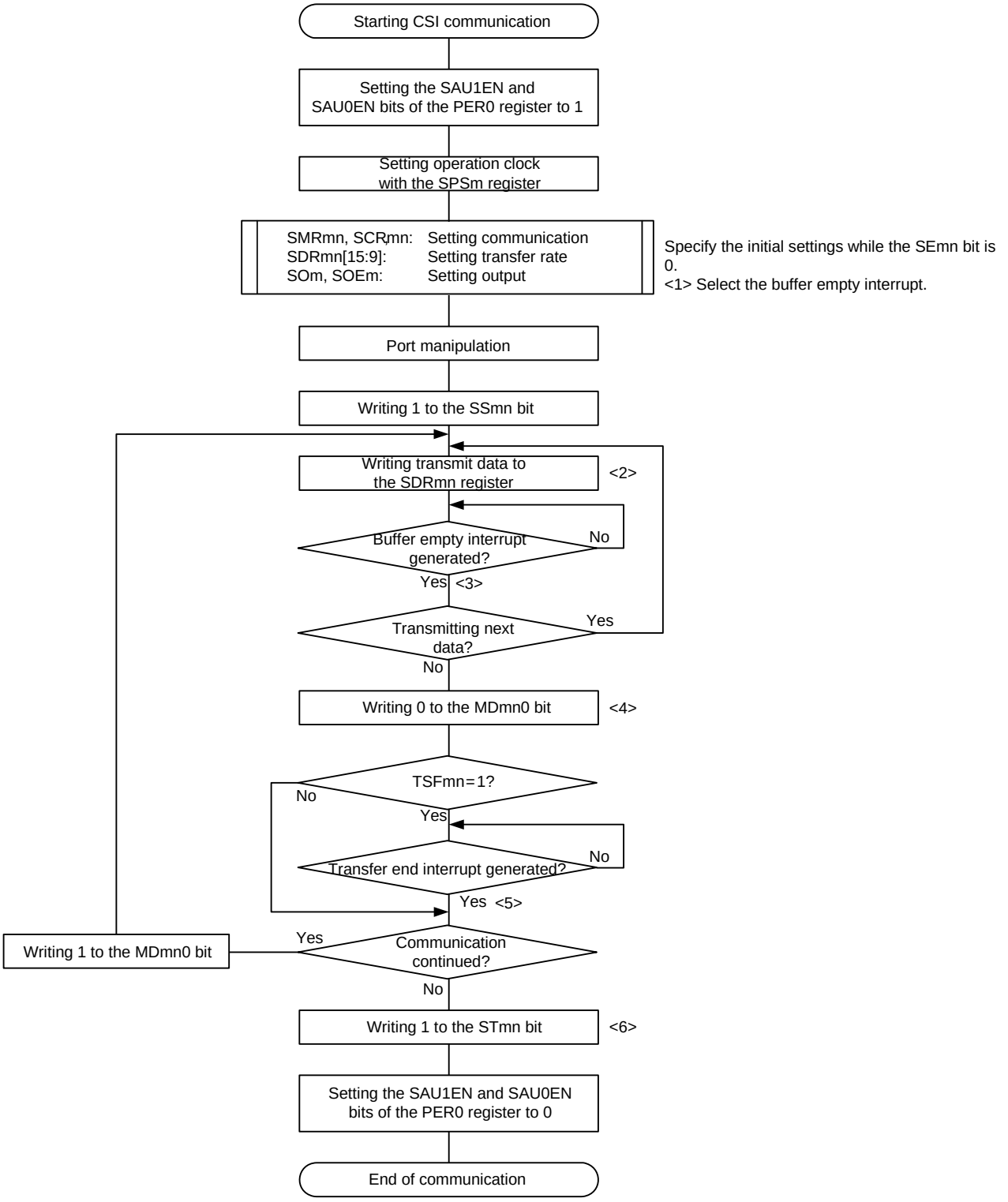



**Note** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

**Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-81. Flowchart of Master Transmission (in Continuous Transmission Mode)**




**Remarks 1.** <1> to <6> in the figure correspond to <1> to <6> in **Figure 15-80 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### 15.6.2 Master reception

Master reception is an operation wherein this MCU outputs a transfer clock and receives data from other device.

| SPI Function | CSI00 | CSI01 | CSI10 | CSI11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | $\overline{\text{SCK00}}$, SI00 | $\overline{\text{SCK01}}$, SI01 | $\overline{\text{SCK10}}$, SI10 | $\overline{\text{SCK11}}$, SI11 |
| Interrupt | INTCSI00 | INTCSI01 | INTCSI10 | INTCSI11 |
| | Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected. | | | |
| Error detection flag | Overrun error detection flag (OVFmn) only | | | |
| Transfer data length | 7 to 16 bits | | | |
| Transfer rate | Max. $f_{MCK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz]**Note** $f_{CLK}$: System clock frequency | | | |
| Data phase | Selectable by the DAPmn bit of the SCRmn register<br>• DAPmn = 0: Data input starts from the start of the serial clock operation.<br>• DAPmn = 1: Data input starts half a clock before the start of the serial clock operation. | | | |
| Clock phase | Selectable by the CKPmn bit of the SCRmn register<br>• CKPmn = 0: Forward<br>• CKPmn = 1: Reverse | | | |
| Data direction | MSB or LSB first | | | |

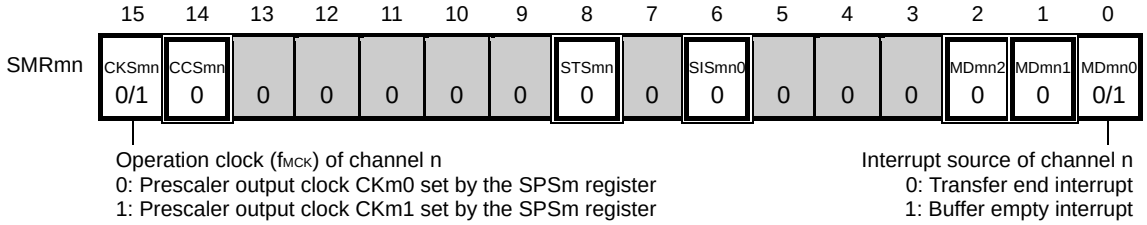
**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

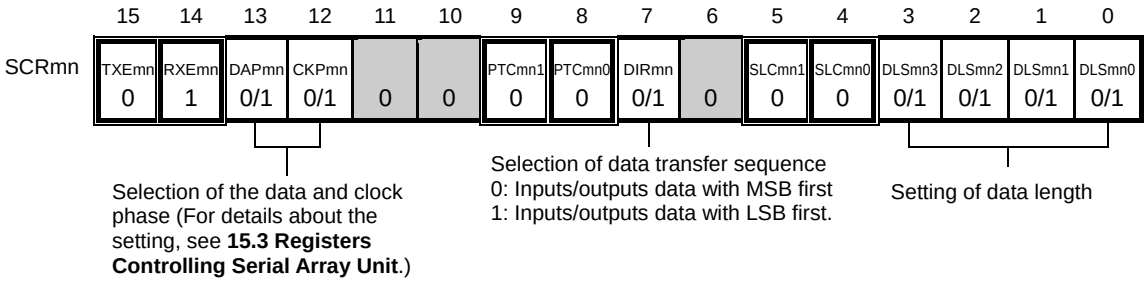
**(1) Register setting**

**Figure 15-82. Example of Contents of Registers for Master Reception of SPI Function (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

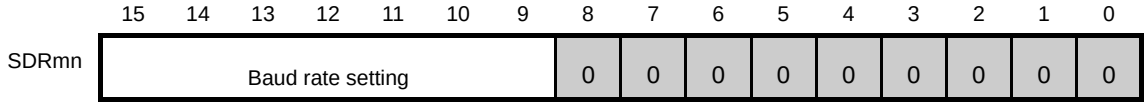



**(b) Serial communication operation setting register mn (SCRmn)**

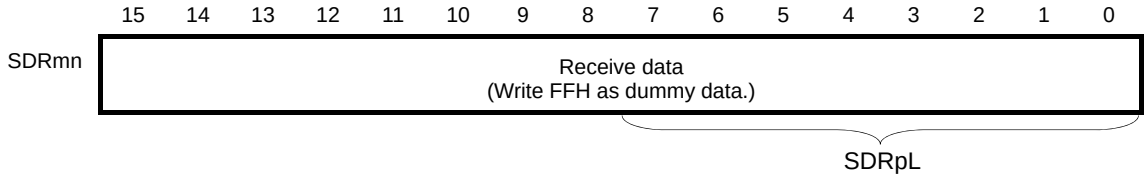



**(c) Serial data register mn (SDRmn)**

**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI master reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-82. Example of Contents of Registers for Master Reception of SPI Function (CSI00, CSI01, CSI10, CSI11) (2/2)**

**(d) Serial output register m (SOm) … Sets only the bits of the target channel.**

SOm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | CKOm1 0/1 | CKOm0 0/1 | 0 | 0 | 0 | 0 | 0 | 0 | SOm1 × | SOm0 × |

Communication starts when these bits are 1 if the clock phase is forward (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

**(e) Serial output enable register m (SOEm) …The register that not used in this mode.**

SOEm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 × | SOEm0 × |

**(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

SSm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1 0/1 | SSm0 0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI master reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

### (2) Operation procedure

**Figure 15-83. Initial Setting Procedure for Master Reception**

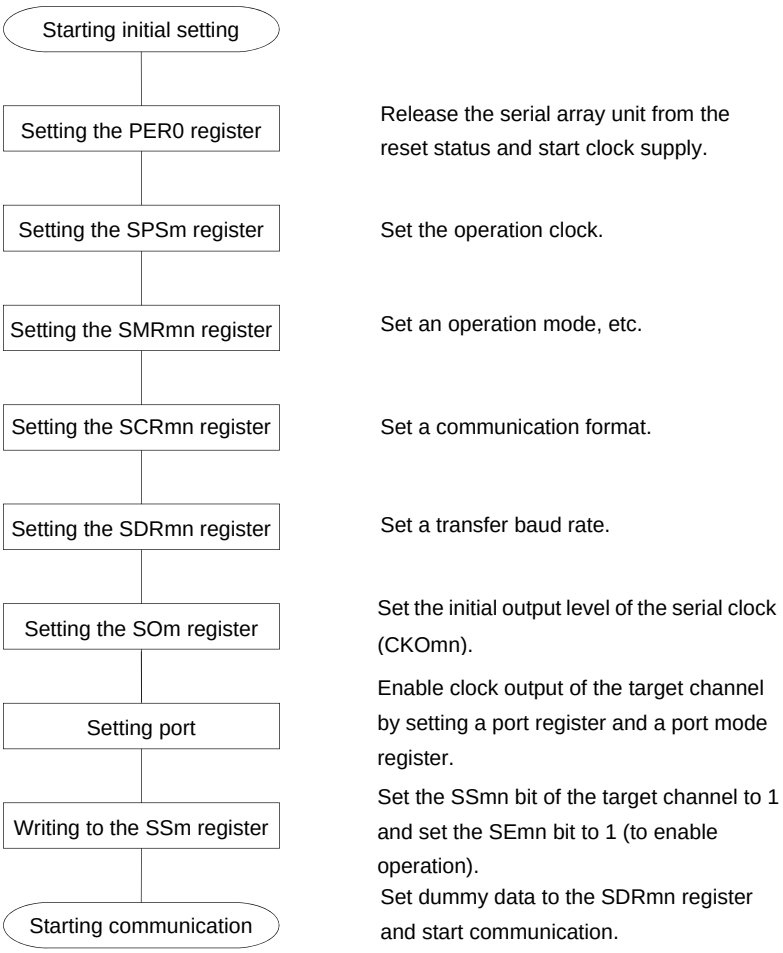
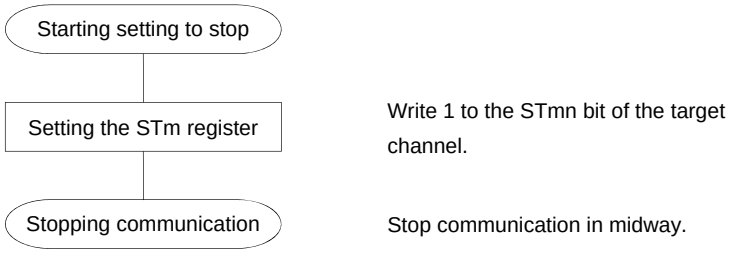


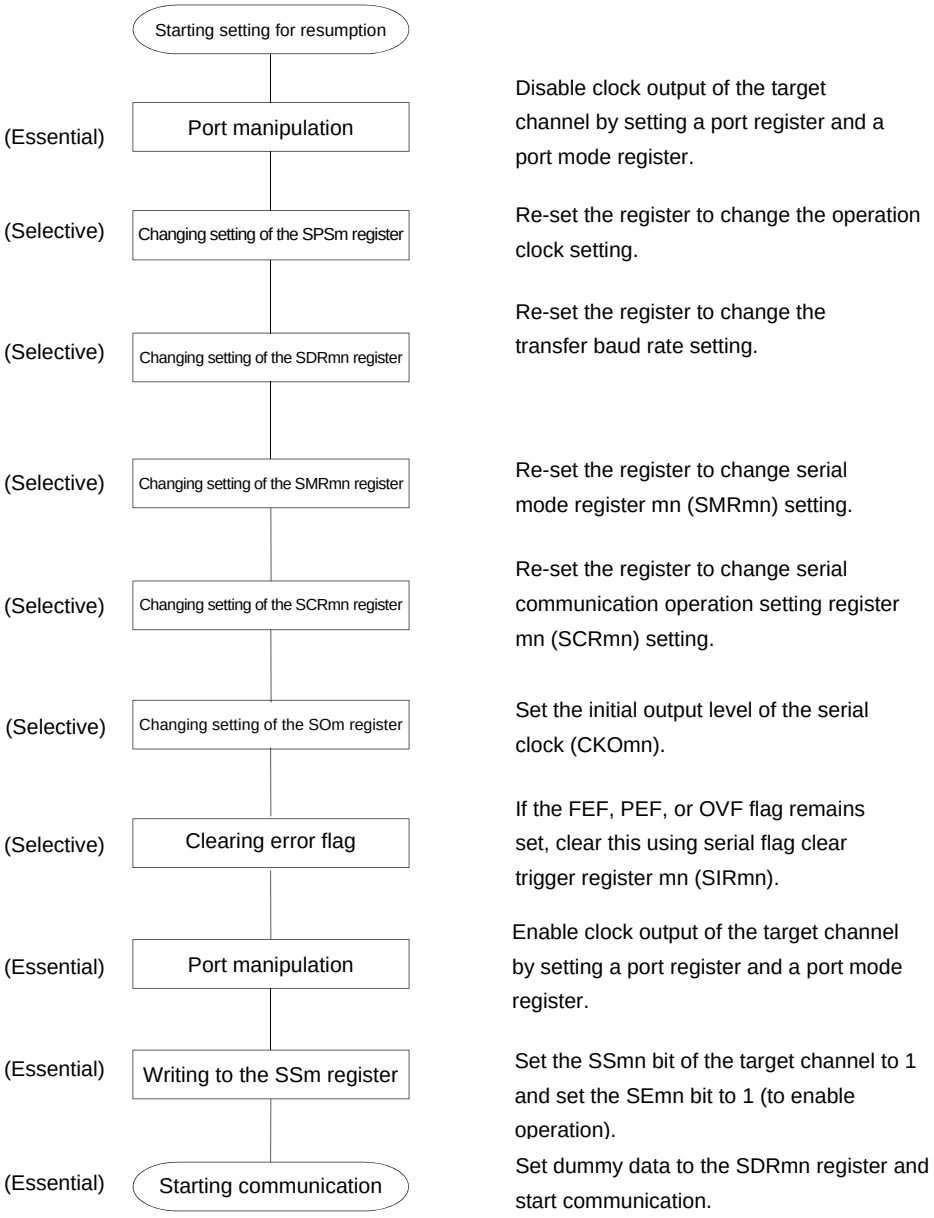

**Figure 15-84. Procedure for Stopping Master Reception**




**Remarks 1.** Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 15-85 Procedure for Resuming Master Reception**).

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

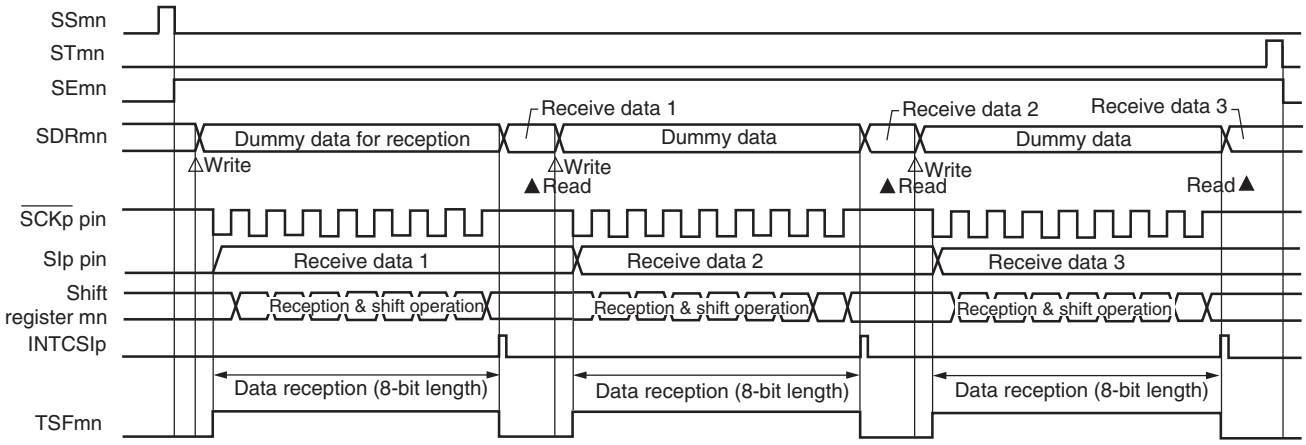
**Figure 15-85. Procedure for Resuming Master Reception**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

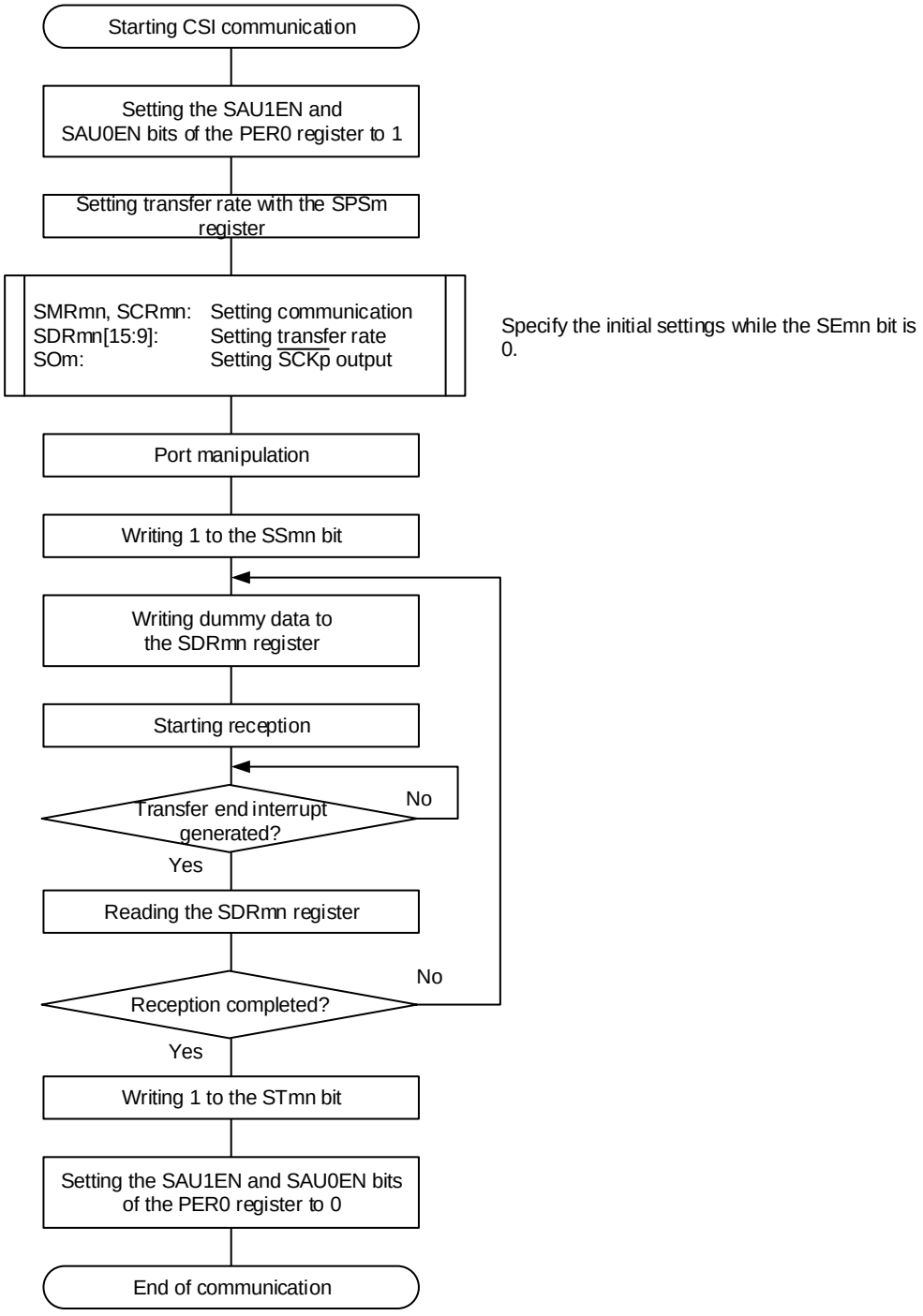
### (3) Processing flow (in single-reception mode)

**Figure 15-86. Timing Chart of Master Reception (in Single-Reception Mode)**
**(Type 1: DAPmn = 0, CKPmn = 0)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

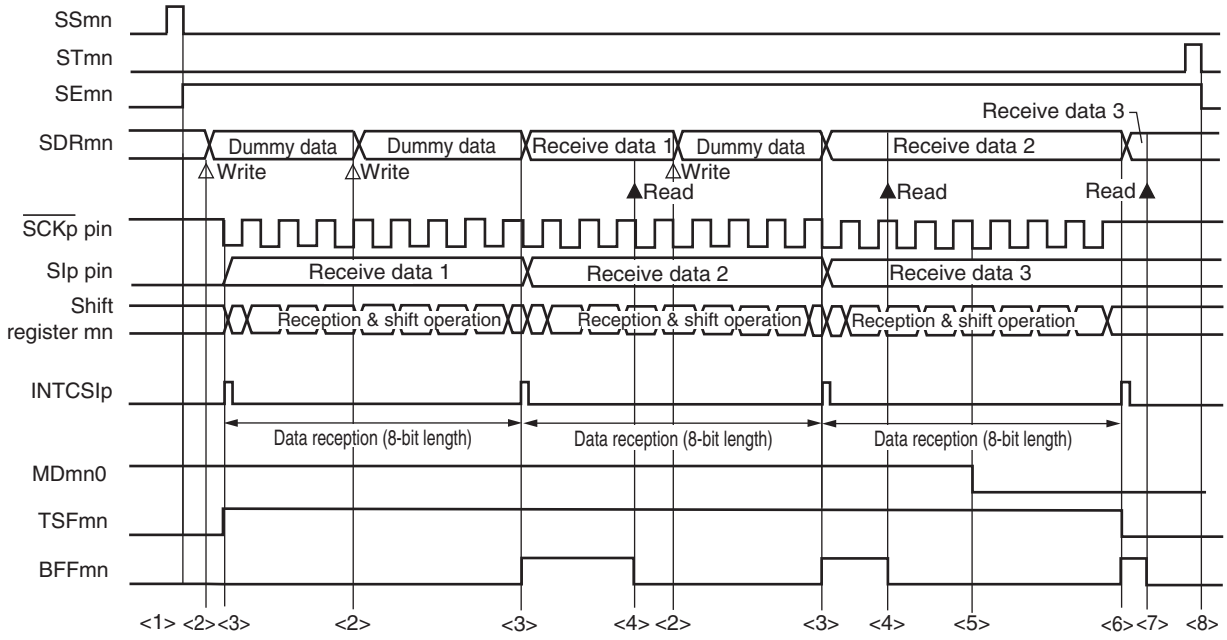
**Figure 15-87. Flowchart of Master Reception (in Single-Reception Mode)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### (4) Processing flow (in continuous reception mode)

**Figure 15-88. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**

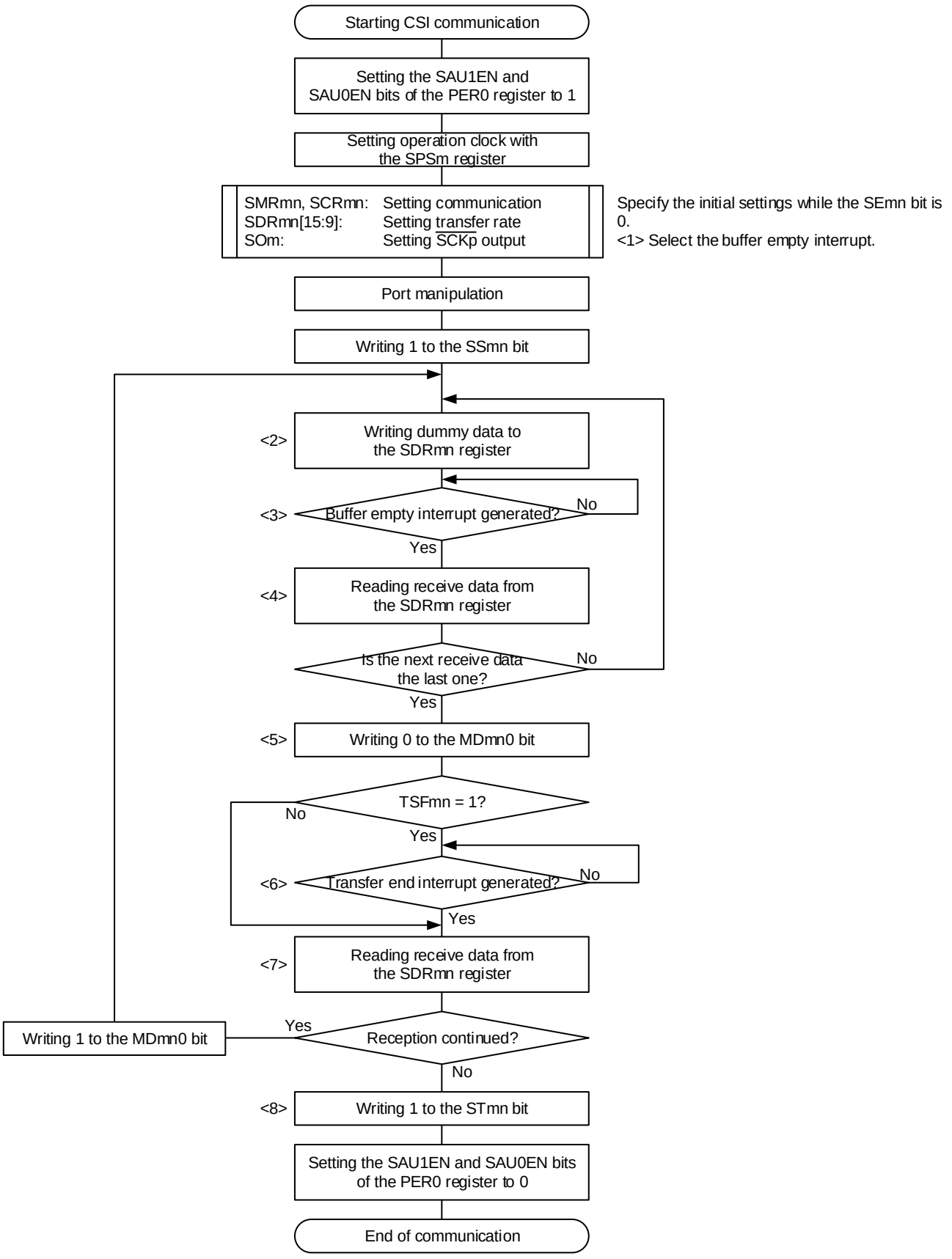



**Caution The MDmn0 bit can be rewritten even during operation.**
**However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.**

**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-89 Flowchart of Master Reception (in Continuous Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-89. Flowchart of Master Reception (in Continuous Reception Mode)**




(**Remarks** are listed on the next page.)

**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in Figure **15-88 Timing Chart of Master Reception (in Continuous Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

### 15.6.3 Master transmission/reception

Master transmission/reception is an operation wherein this MCU outputs a transfer clock and transmits/receives data to/from other device.

| SPI Function | CSI00 | CSI01 | CSI10 | CSI11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | $\overline{\text{SCK00}}$, SI00, SO00 | $\overline{\text{SCK01}}$, SI01, SO01 | $\overline{\text{SCK10}}$, SI10, SO10 | $\overline{\text{SCK11}}$, SI11, SO11 |
| Interrupt | INTCSI00 | INTCSI01 | INTCSI10 | INTCSI11 |
| | Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected. | | | |
| Error detection flag | Overrun error detection flag (OVFmn) only | | | |
| Transfer data length | 7 to 16 bits | | | |
| Transfer rate | Max. $f_{MCK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] **Note** $f_{CLK}$: System clock frequency | | | |
| Data phase | Selectable by the DAPmn bit of the SCRmn register<br>• DAPmn = 0: Data I/O starts at the start of the serial clock operation.<br>• DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation. | | | |
| Clock phase | Selectable by the CKPmn bit of the SCRmn register<br>• CKPmn = 0: Forward<br>• CKPmn = 1: Reverse | | | |
| Data direction | MSB or LSB first | | | |

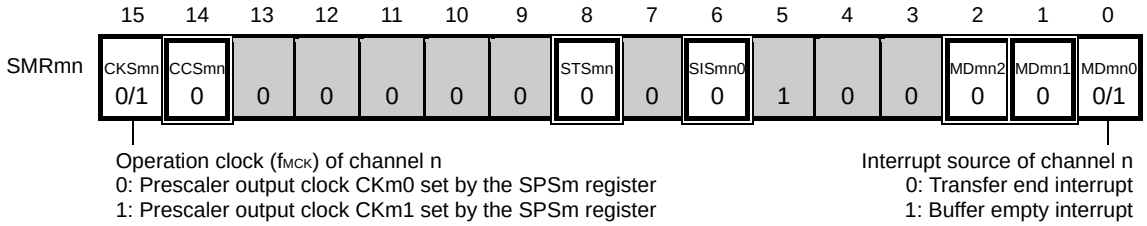
**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

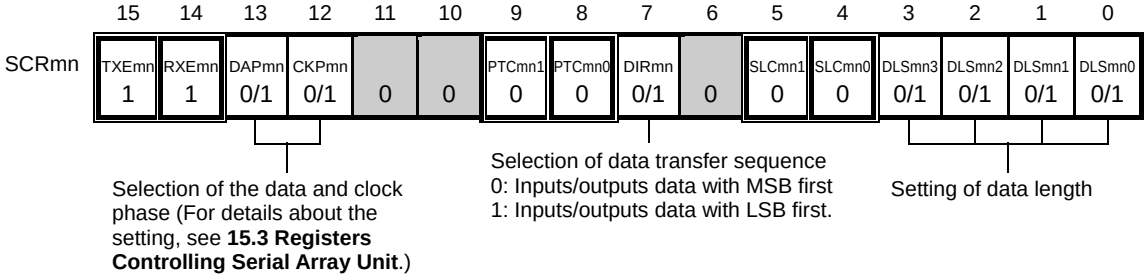
### (1) Register setting

**Figure 15-90. Example of Contents of Registers for Master Transmission/Reception of SPI Function (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

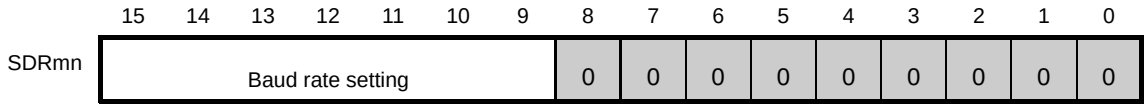



**(b) Serial communication operation setting register mn (SCRmn)**

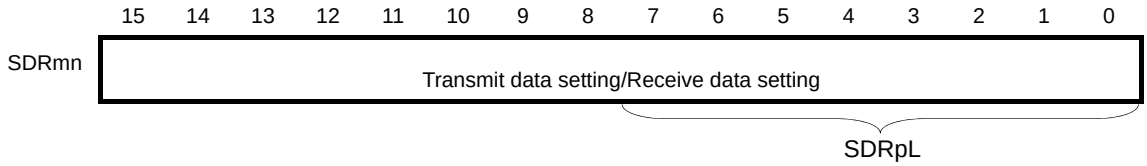



**(c) Serial data register mn (SDRmn)**

**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI master transmission/reception mode

▒: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-90. Example of Contents of Registers for Master Transmission/Reception of SPI Function (CSI00, CSI01, CSI10, CSI11) (2/2)**

**(d) Serial output register m (SOm) … Sets only the bits of the target channel.**

SOm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | CKOm1 0/1 | CKOm0 0/1 | 0 | 0 | 0 | 0 | 0 | 0 | SOm1 0/1 | SOm0 0/1 |

Communication starts when these bits are 1 if the clock phase is forward (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

**(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.**

SOEm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 0/1 | SOEm0 0/1 |

**(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

SSm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1 0/1 | SSm0 0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

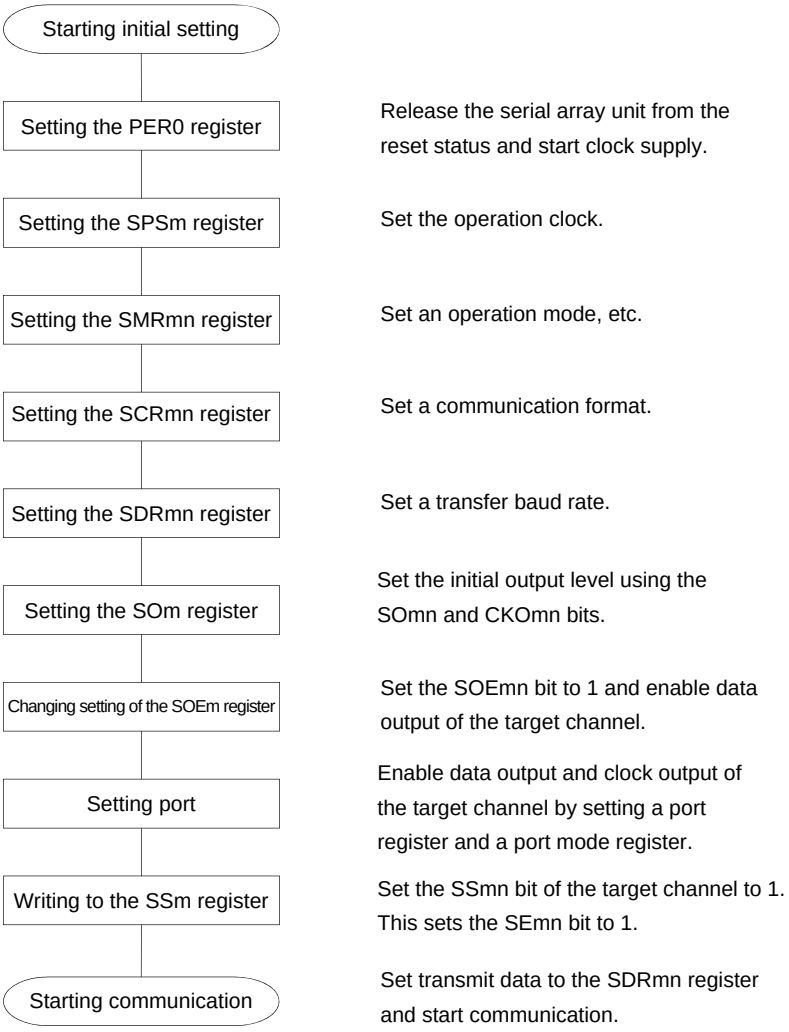
**2.** ☐: Setting is fixed in the CSI master transmission/reception mode

▒: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

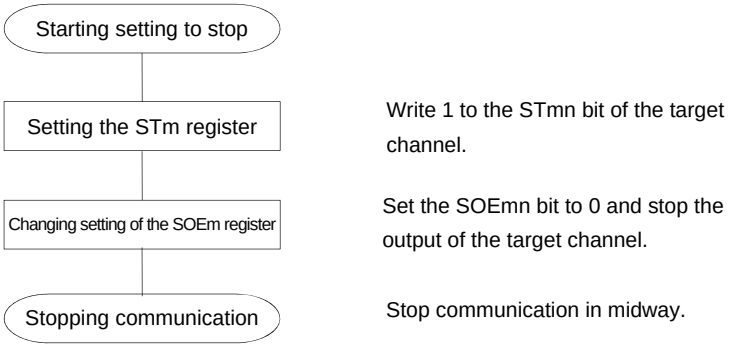
### (2) Operation procedure

**Figure 15-91. Initial Setting Procedure for Master Transmission/Reception**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

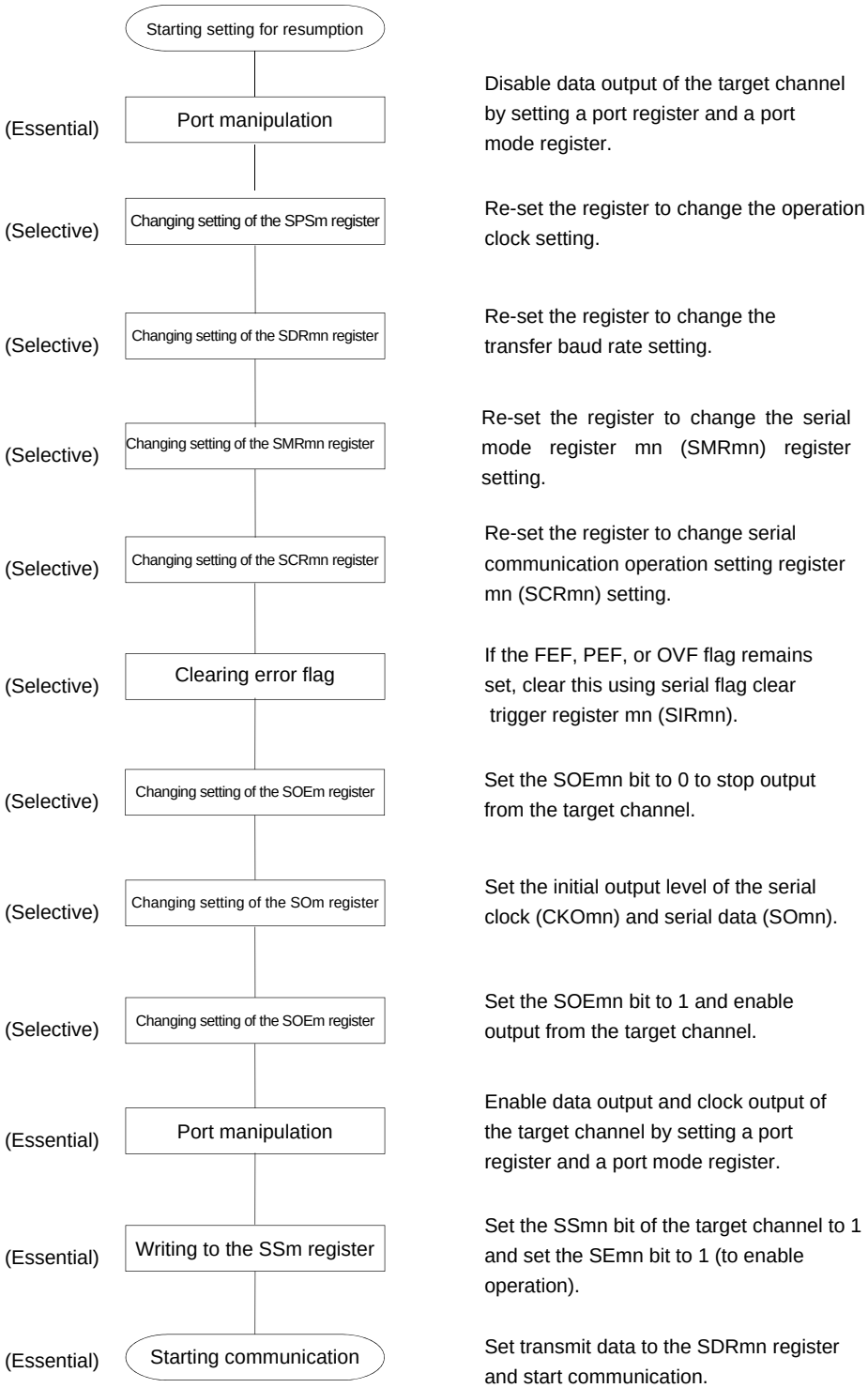
**Figure 15-92. Procedure for Stopping Master Transmission/Reception**




**Remarks 1.** Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 15-93 Procedure for Resuming Master Transmission/ Reception**).

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

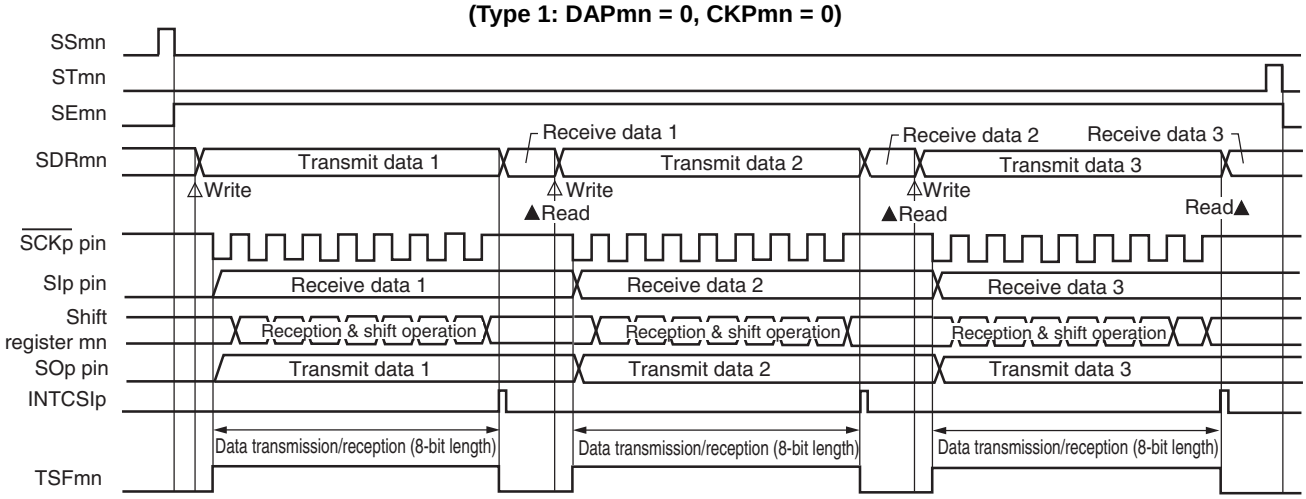
**Figure 15-93. Procedure for Resuming Master Transmission/Reception**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

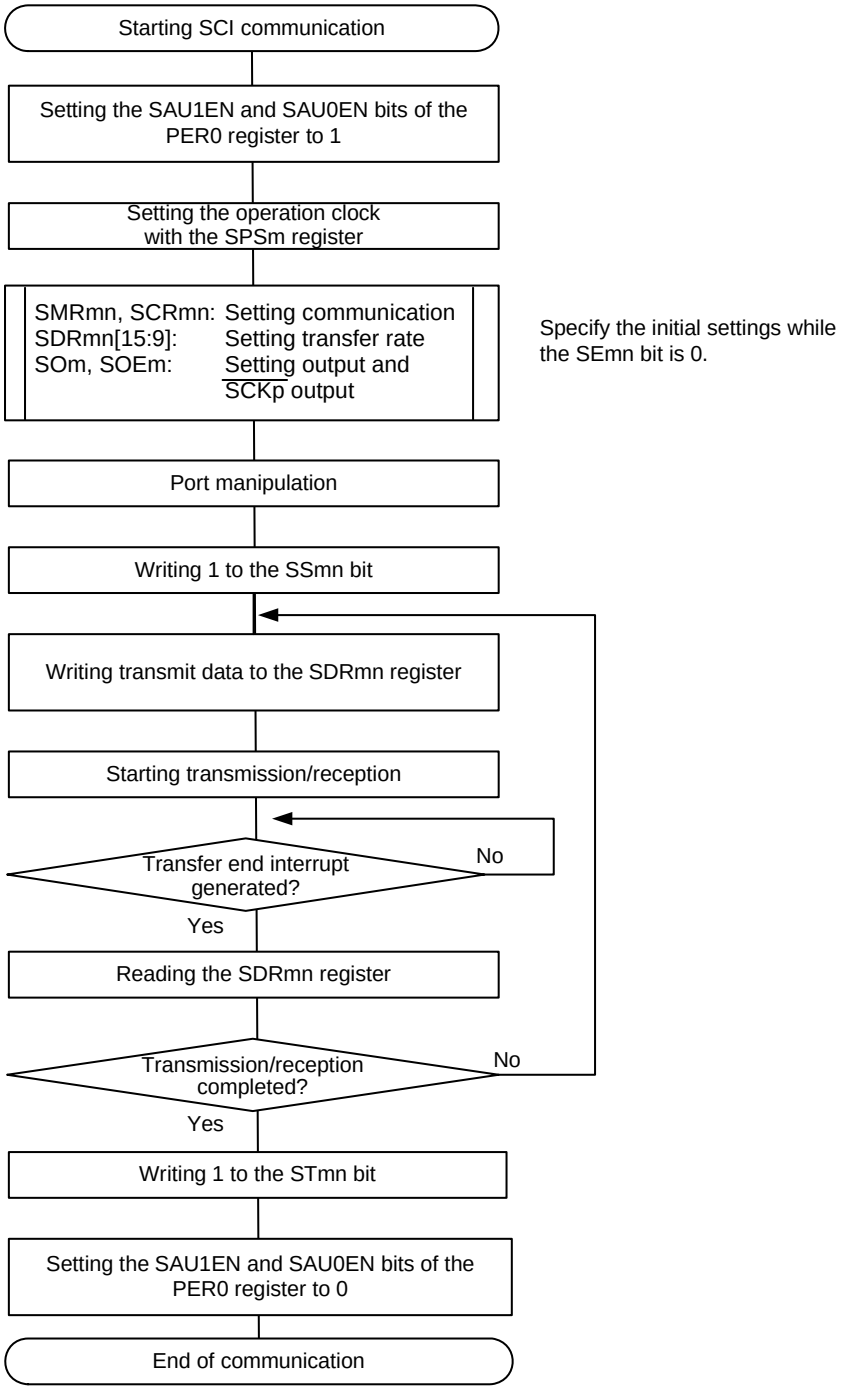
### (3) Processing flow (in single-transmission/reception mode)

**Figure 15-94. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

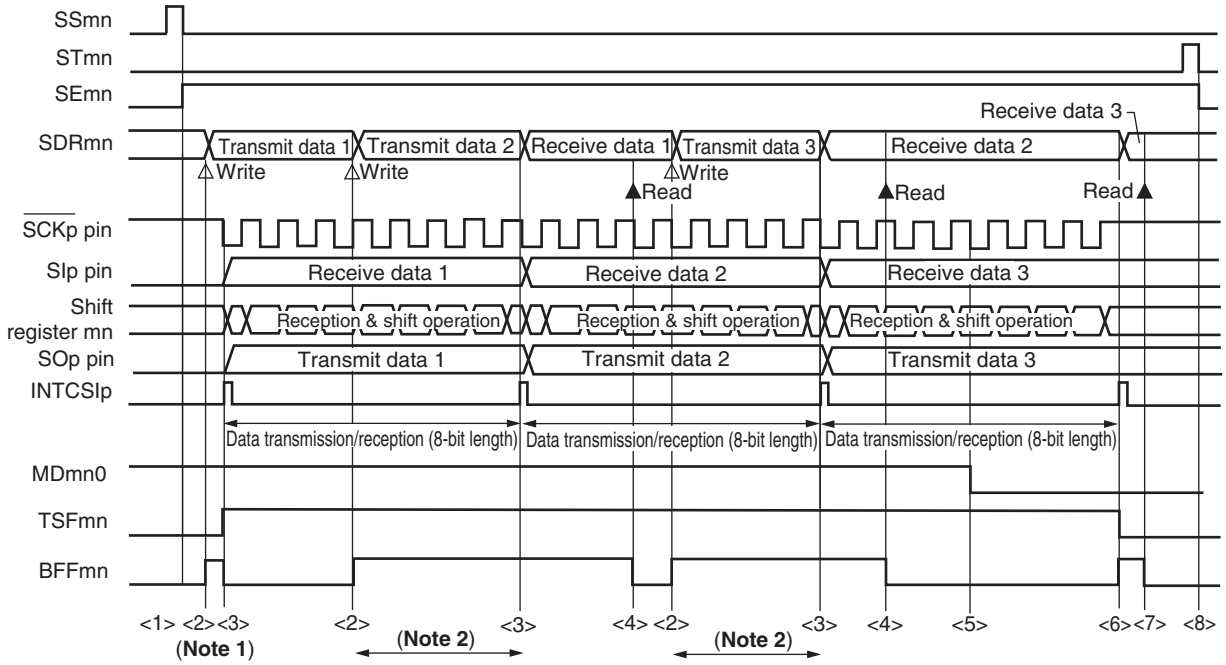
**Figure 15-95. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

### (4) Processing flow (in continuous transmission/reception mode)

**Figure 15-96. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**




**Notes 1.** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

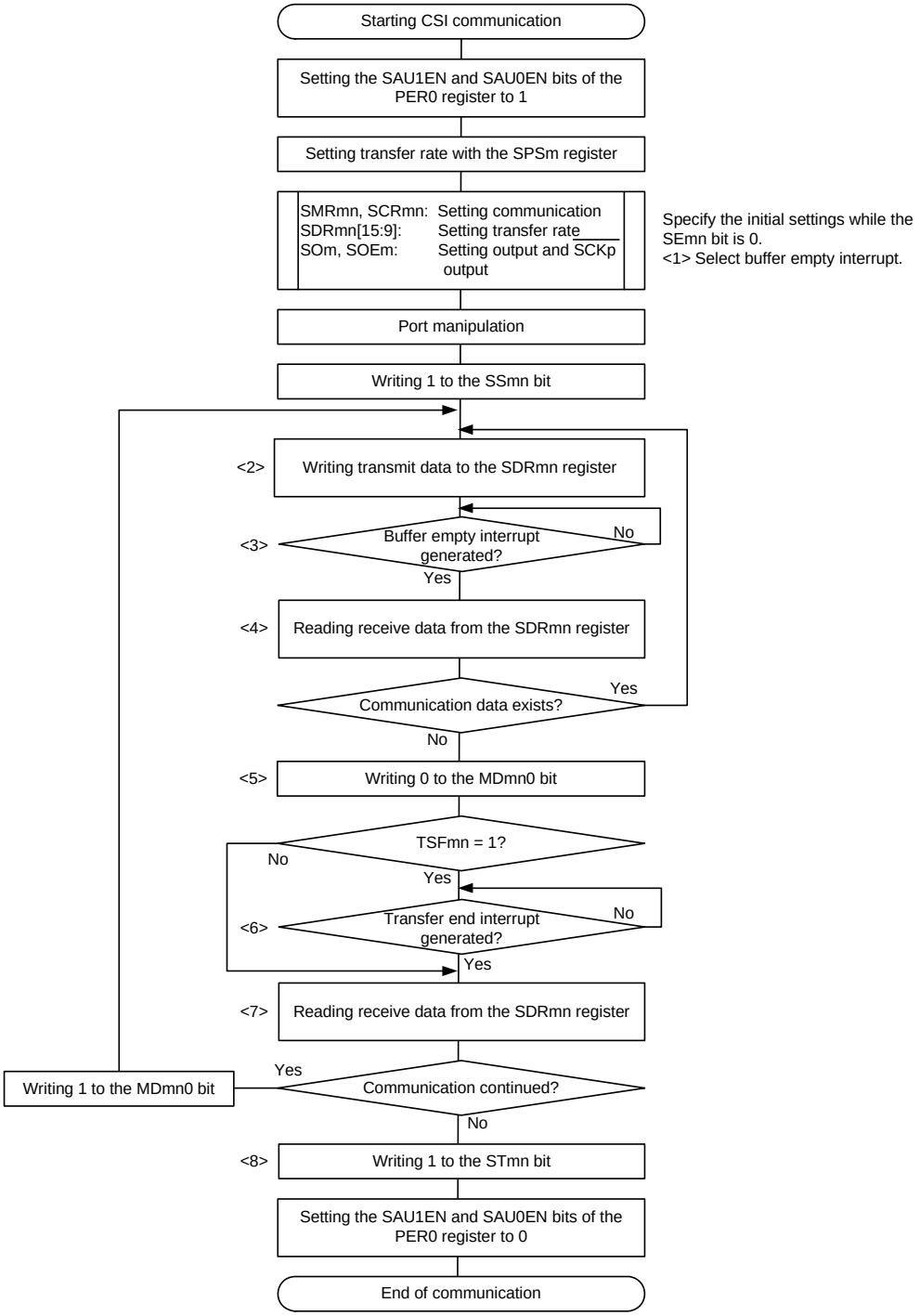
**2.** The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

**Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.**

**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-97 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-97. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**




**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-96 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

### 15.6.4 Slave transmission

Slave transmission is an operation wherein this MCU transmits data to another device in the state of a transfer clock being input from another device.

| SPI Function | CSI00 | CSI01 | CSI10 | CSI11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | $\overline{\text{SCK00}}$, SO00, $\overline{\text{SSI00}}$ | $\overline{\text{SCK01}}$, SO01, $\overline{\text{SSI01}}$ | $\overline{\text{SCK10}}$, SO10, $\overline{\text{SSI10}}$ | $\overline{\text{SCK11}}$, SO11, $\overline{\text{SSI11}}$ |
| Interrupt | INTCSI00 | INTCSI01 | INTCSI10 | INTCSI11 |
| | Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected. | | | |
| Error detection flag | Overrun error detection flag (OVFmn) only | | | |
| Transfer data length | 7 to 16 bits | | | |
| Transfer rate | Max. $f_{MCK}$/6 [Hz]Notes 1, 2. | | | |
| Data phase | Selectable by the DAPmn bit of the SCRmn register<br>• DAPmn = 0: Data output starts from the start of the serial clock operation.<br>• DAPmn = 1: Data output starts half a clock before the start of the serial clock operation. | | | |
| Clock phase | Selectable by the CKPmn bit of the SCRmn register<br>• CKPmn = 0: Forward<br>• CKPmn = 1: Reverse | | | |
| Data direction | MSB or LSB first | | | |
| SPI function | The operation of the slave select function can be selected. | | | |

**Notes 1.** Because the external serial clock input to the $\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$, and $\overline{\text{SCK11}}$ pins is sampled internally and used, the fastest transfer rate is $f_{MCK}$/6 [Hz].

**2.** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

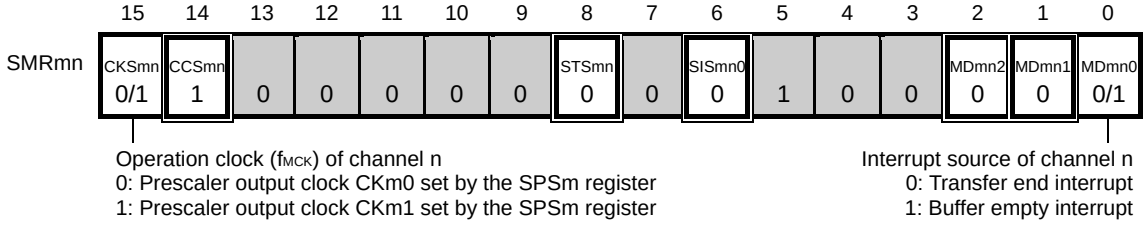
**Remarks 1.** $f_{MCK}$: Operation clock frequency of target channel

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

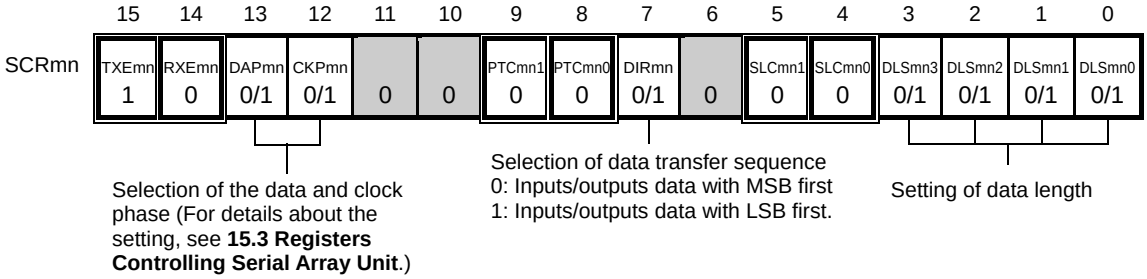
**(1) Register setting**

**Figure 15-98. Example of Contents of Registers for Slave Transmission of SPI Function (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

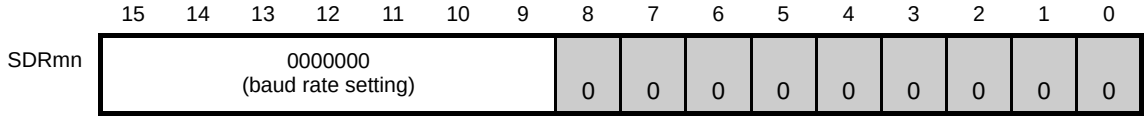



**(b) Serial communication operation setting register mn (SCRmn)**

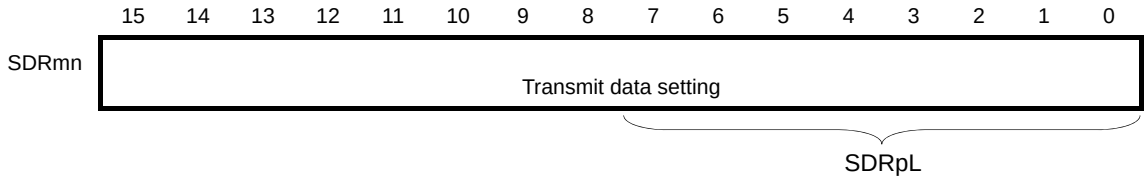



**(c) Serial data register mn (SDRmn)**

**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐ : Setting is fixed in the CSI slave transmission mode

▒ : Setting disabled (set to the initial value)

× : Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1 : Set to 0 or 1 depending on the usage of the user

**Figure 15-98. Example of Contents of Registers for Slave Transmission of SPI Function (CSI00, CSI01, CSI10, CSI11) (2/2)**

**(d) Serial slave select enable register m (SSEm) … Controls the $\overline{\text{SSI00}}$, $\overline{\text{SSI01}}$, $\overline{\text{SSI10}}$, and $\overline{\text{SSI11}}$ pin inputs of the target channel in slave mode.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSEm1<br>0/1 | SSEm0<br>0/1 |

**(e) Serial output register m (SOm) … Sets only the bits of the target channel.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOm | 0 | 0 | 0 | 0 | 0 | 0 | CKOm1<br>× | CKOm0<br>× | 0 | 0 | 0 | 0 | 0 | 0 | SOm1<br>0/1 | SOm0<br>0/1 |

**(f) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1<br>0/1 | SOEm0<br>0/1 |

**(g) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1<br>0/1 | SSm0<br>0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** □ : Setting is fixed in the CSI slave transmission mode

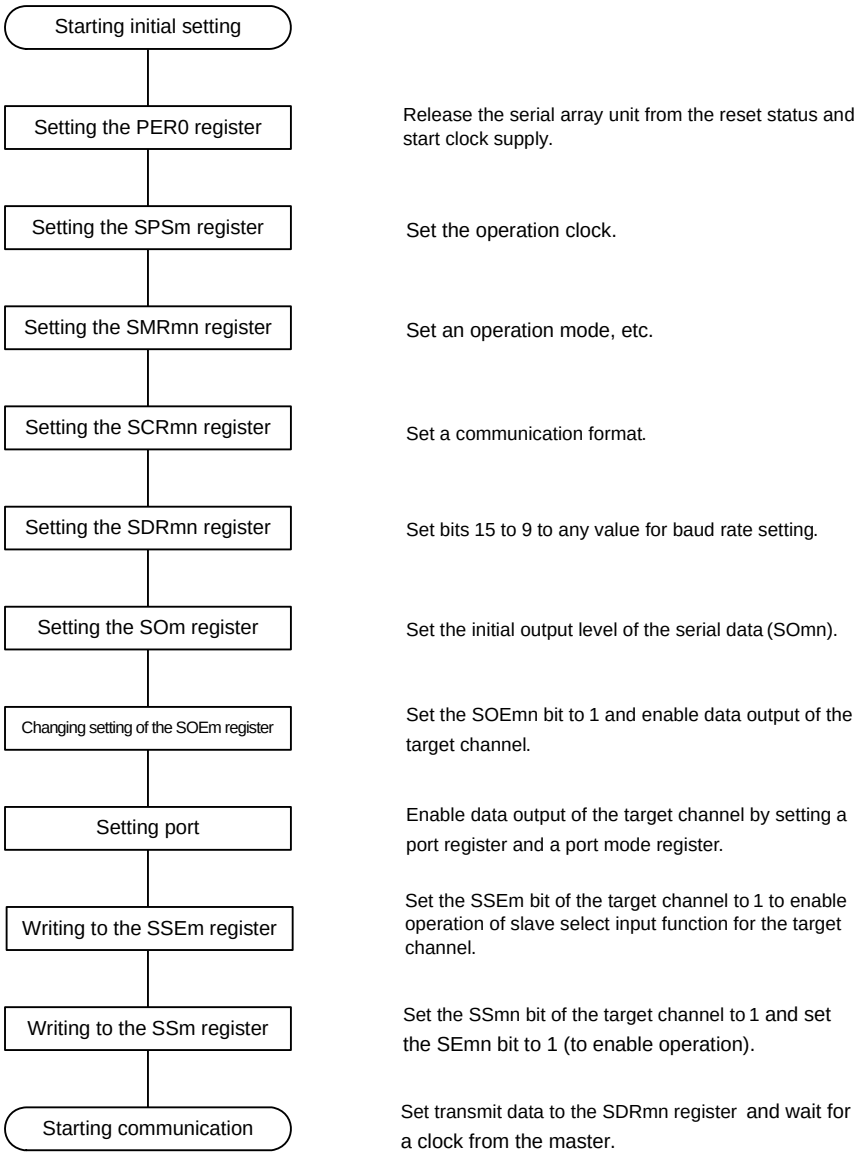
▒ : Setting disabled (set to the initial value)

× : Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1 : Set to 0 or 1 depending on the usage of the user

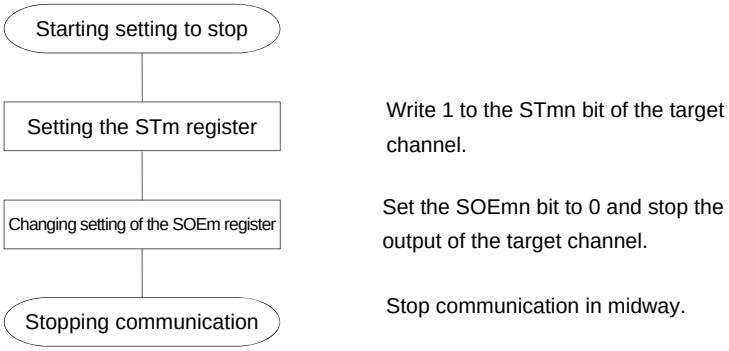
### (2) Operation procedure

**Figure 15-99. Initial Setting Procedure for Slave Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

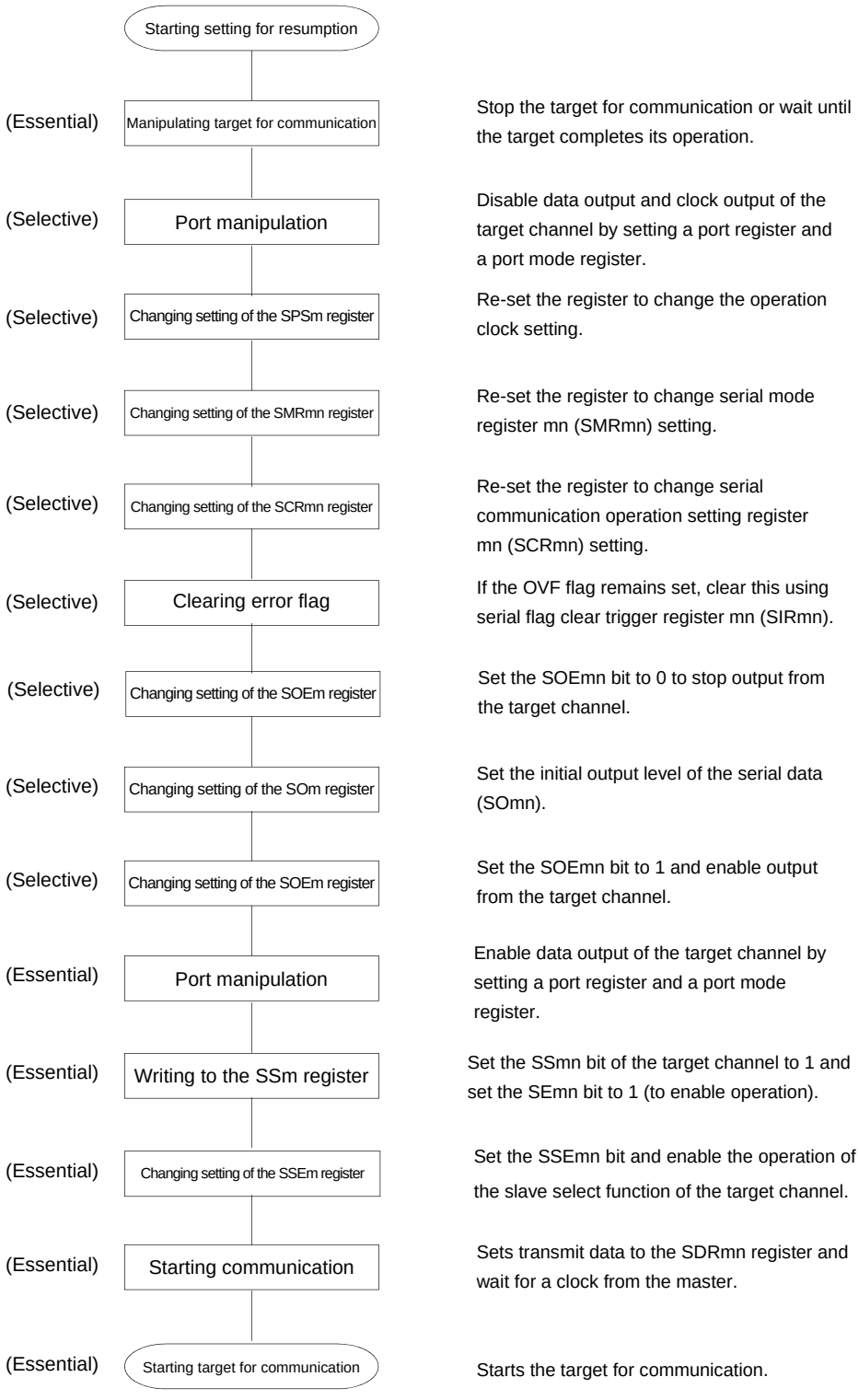
**Figure 15-100. Procedure for Stopping Slave Transmission**




**Remarks 1.** Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 15-101 Procedure for Resuming Slave Transmission**).

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

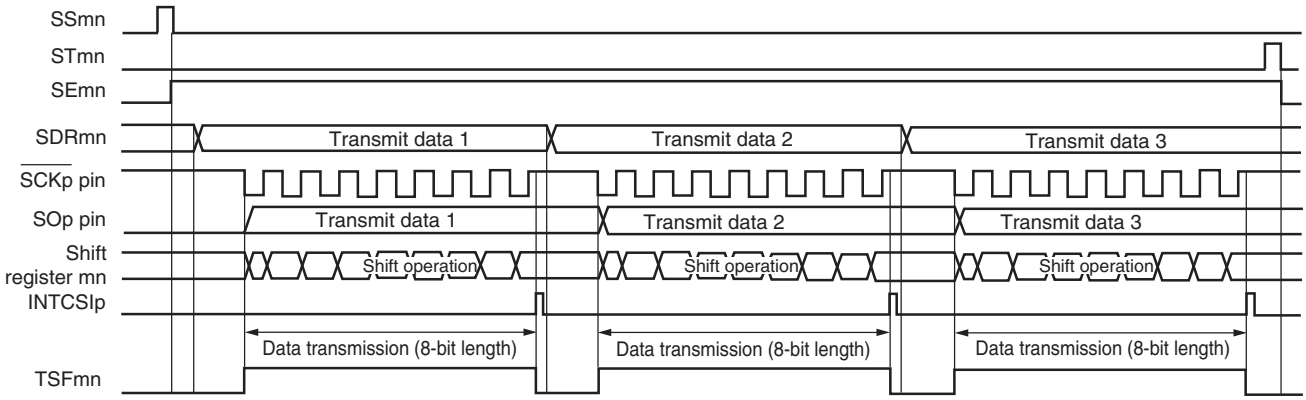
**Figure 15-101. Procedure for Resuming Slave Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

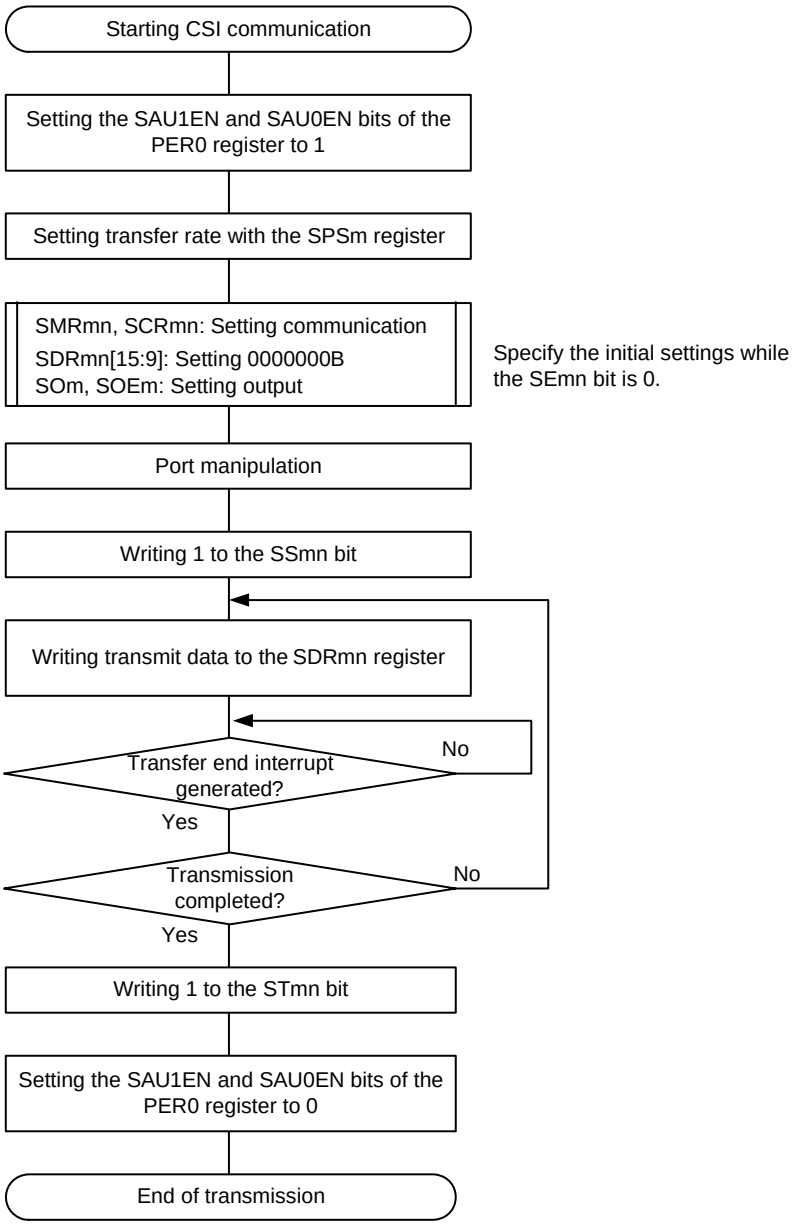
### (3) Processing flow (in single-transmission mode)

**Figure 15-102. Timing Chart of Slave Transmission (in Single-Transmission Mode)**
**(Type 1: DAPmn = 0, CKPmn = 0)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

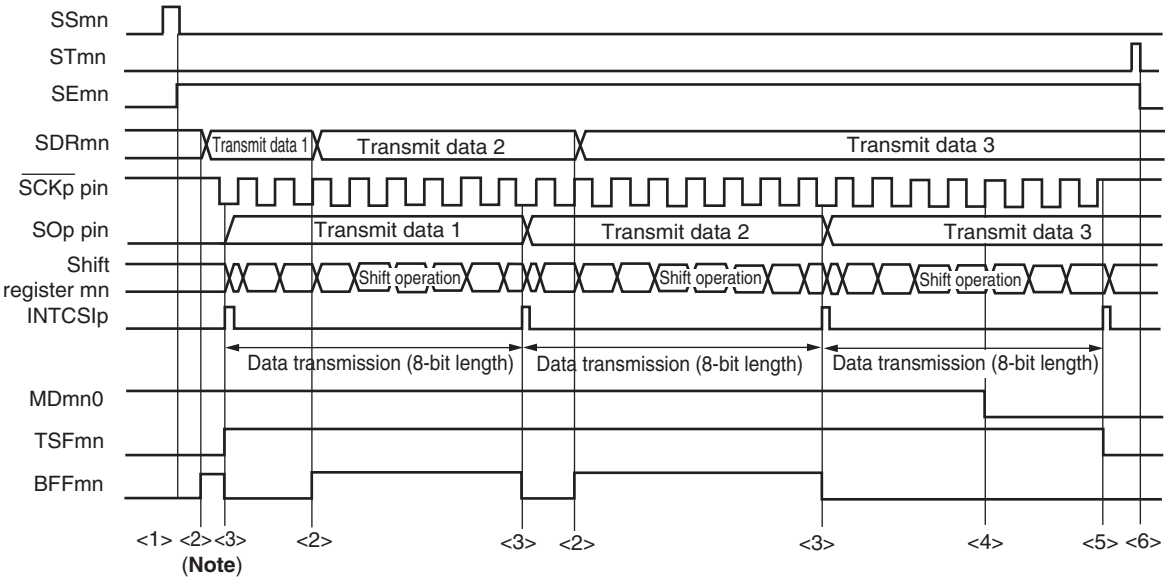
**Figure 15-103. Flowchart of Slave Transmission (in Single-Transmission Mode)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### (4) Processing flow (in continuous transmission mode)

**Figure 15-104. Timing Chart of Slave Transmission (in Continuous Transmission Mode)**
**(Type 1: DAPmn = 0, CKPmn = 0)**

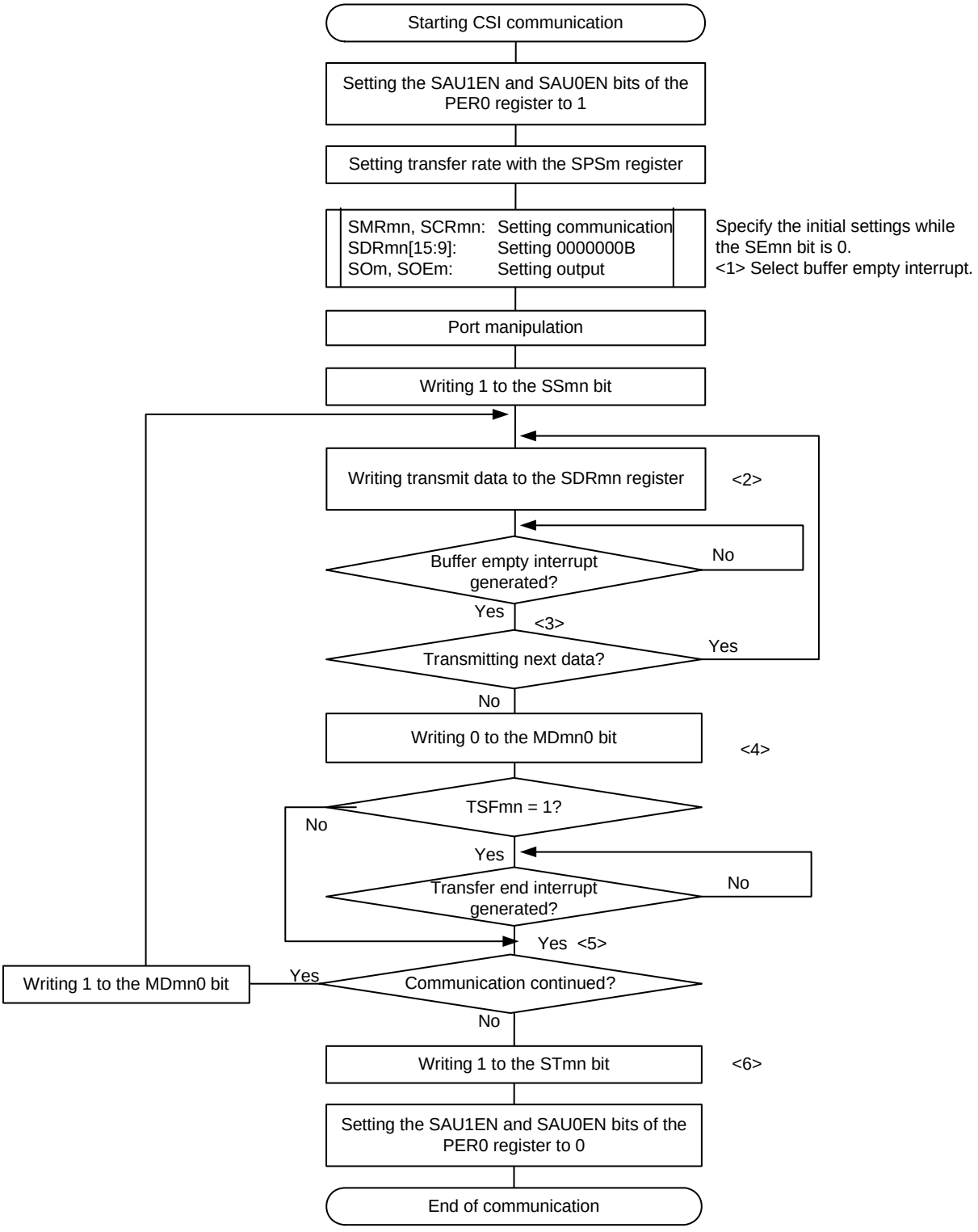



**Note** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

**Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-105. Flowchart of Slave Transmission (in Continuous Transmission Mode)**




**Remarks 1.** <1> to <6> in the figure correspond to <1> to <6> in **Figure 15-104 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

### 15.6.5 Slave reception

Slave reception is an operation wherein this MCU receives data from another device in the state of a transfer clock being input from another device.

| SPI Function | CSI00 | CSI01 | CSI10 | CSI11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 2 of SAU0 | Channel 3 of SAU0 |
| Pins used | $\overline{SCK00}$, SI00, $\overline{SII00}$ | $\overline{SCK01}$, SI01, $\overline{SII01}$ | $\overline{SCK10}$, SI10, $\overline{SII10}$ | $\overline{SCK11}$, SI11, $\overline{SII11}$ |
| Interrupt | INTCSI00 | INTCSI01 | INTCSI10 | INTCSI11 |
| | Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.) | | | |
| Error detection flag | Overrun error detection flag (OVFmn) only | | | |
| Transfer data length | 7 to 16 bits | | | |
| Transfer rate | Max. $f_{MCK}$/6 [Hz] Notes 1, 2 | | | |
| Data phase | Selectable by the DAPmn bit of the SCRmn register<br>• DAPmn = 0: Data input starts from the start of the serial clock operation.<br>• DAPmn = 1: Data input starts half a clock before the start of the serial clock operation. | | | |
| Clock phase | Selectable by the CKPmn bit of the SCRmn register<br>• CKPmn = 0: Forward<br>• CKPmn = 1: Reverse | | | |
| Data direction | MSB or LSB first | | | |
| SPI function | The operation of the slave select function can be selected. | | | |

**Notes 1.** Because the external serial clock input to the $\overline{SCK00}$, $\overline{SCK01}$, $\overline{SCK10}$, and $\overline{SCK11}$ pins is sampled internally and used, the fastest transfer rate is $f_{MCK}$/6 [Hz].

**2.** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

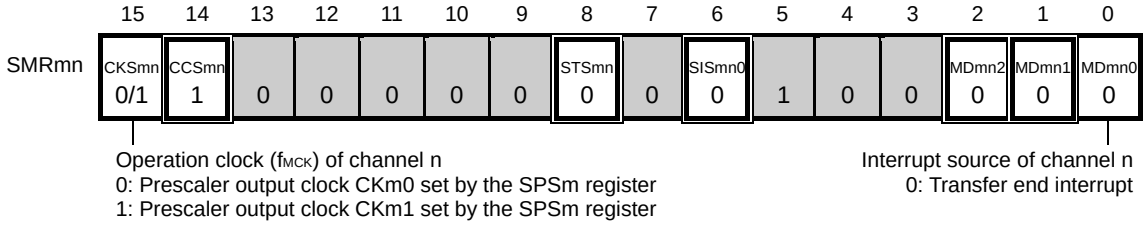
**Remarks 1.** $f_{MCK}$: Operation clock frequency of target channel

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

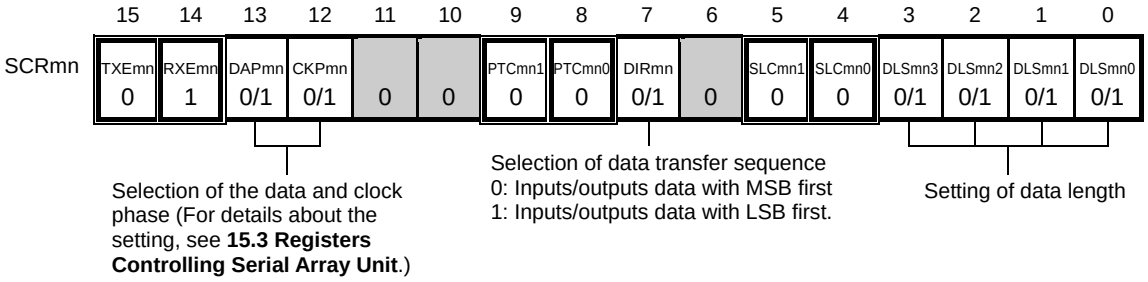
**(1) Register setting**

**Figure 15-106. Example of Contents of Registers for Slave Reception of SPI Function (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

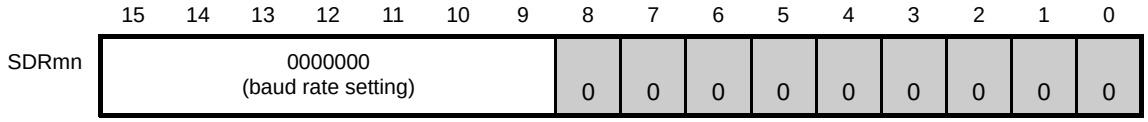



**(b) Serial communication operation setting register mn (SCRmn)**

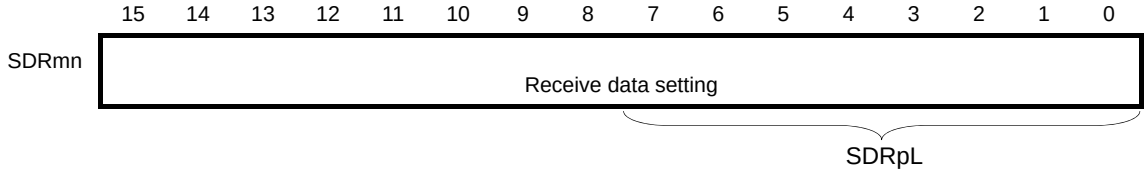



**(c) Serial data register mn (SDRmn)**

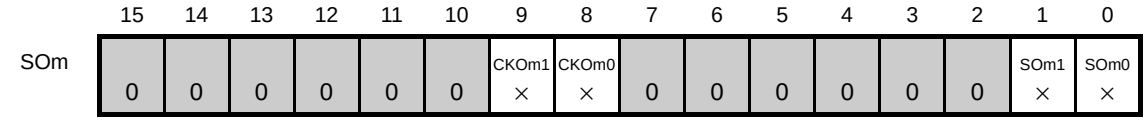
**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)**




**(d) Serial output register m (SOm) …The register that not used in this mode.**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** □ : Setting is fixed in the CSI slave reception mode, ■ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-106. Example of Contents of Registers for Slave Reception of SPI Function (CSI00, CSI01, CSI10, CSI11) (2/2)**

**(e) Serial output enable register m (SOEm) …The register that not used in this mode.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 × | SOEm0 × |

**(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1 0/1 | SSm0 0/1 |

**(g) Serial slave select enable register m (SSEm) … Controls the $\overline{\text{SSI00}}$, $\overline{\text{SSI01}}$, $\overline{\text{SSI10}}$, and $\overline{\text{SSI11}}$ pin inputs of the target channel in slave mode.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSEm1 0/1 | SSEm0 0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI slave reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

### (2) Operation procedure

**Figure 15-107. Initial Setting Procedure for Slave Reception**

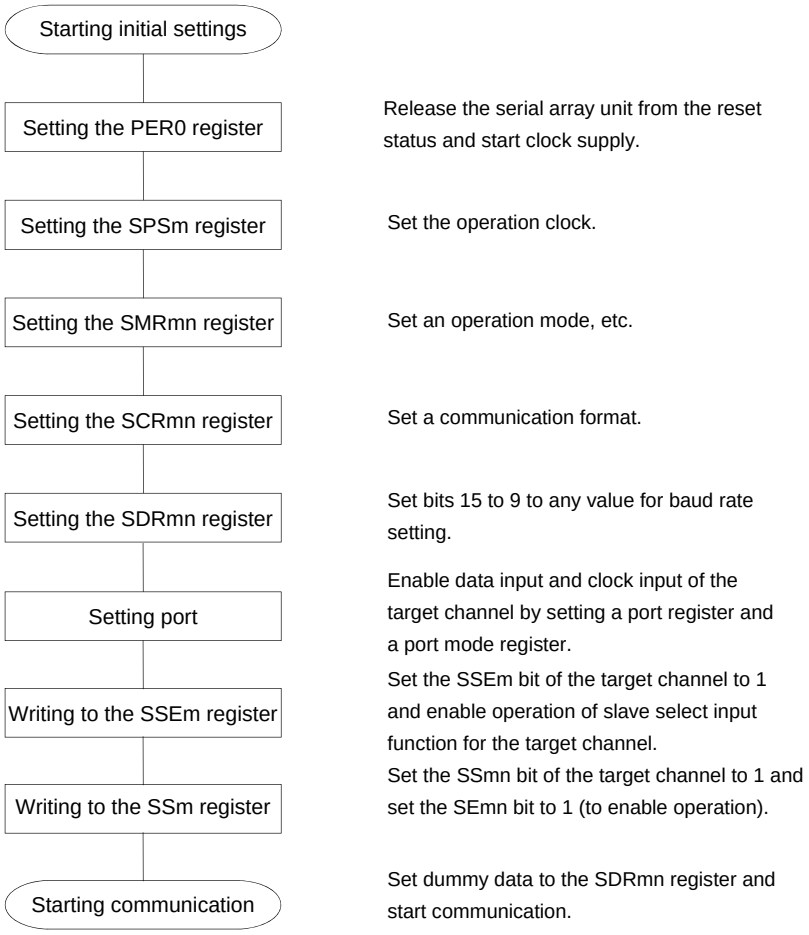
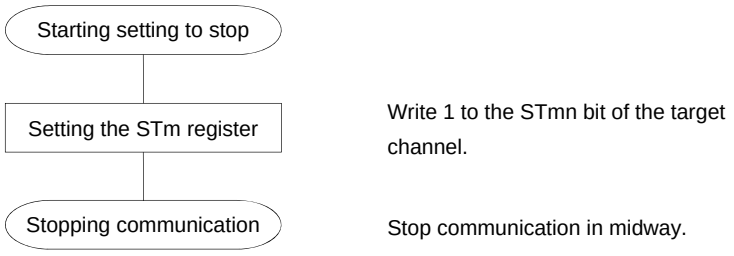
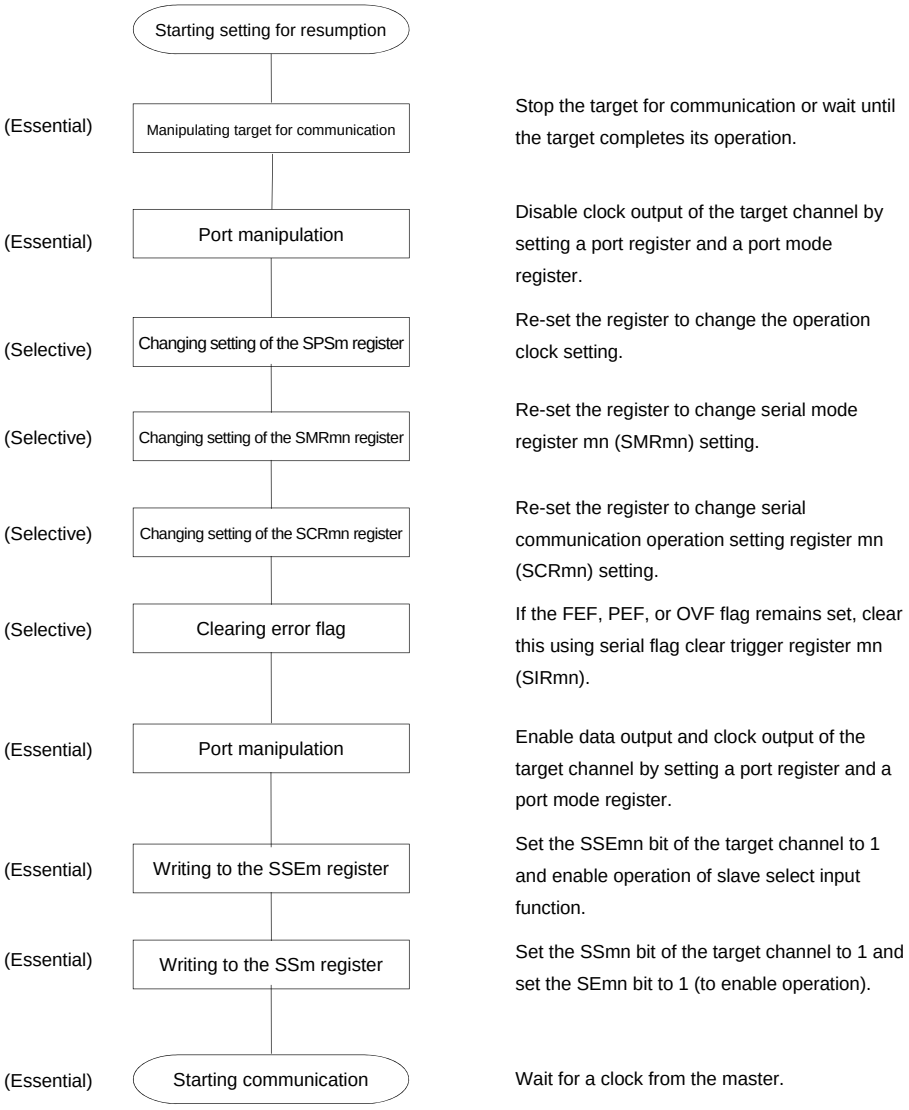



**Figure 15-108. Procedure for Stopping Slave Reception**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

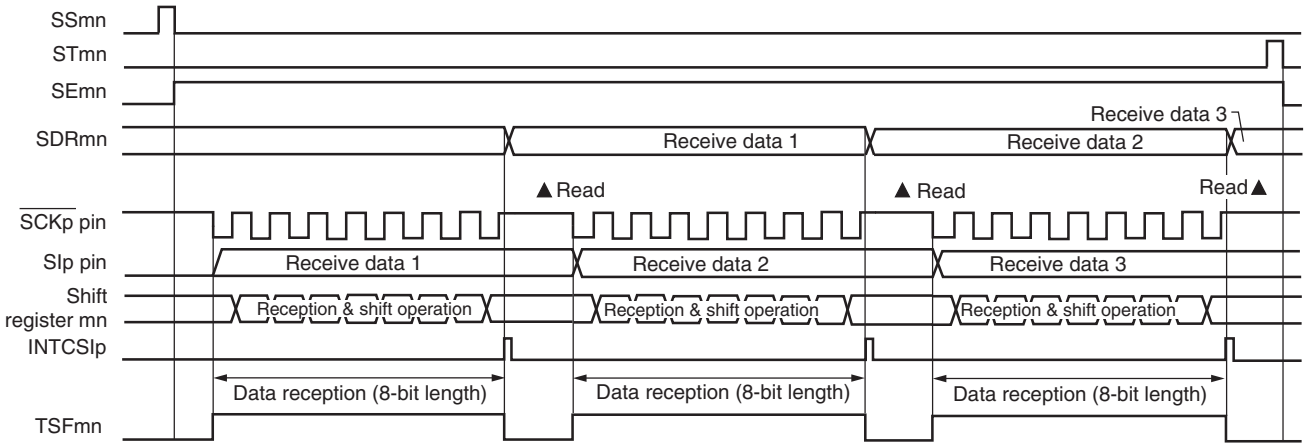
**Figure 15-109. Procedure for Resuming Slave Reception**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

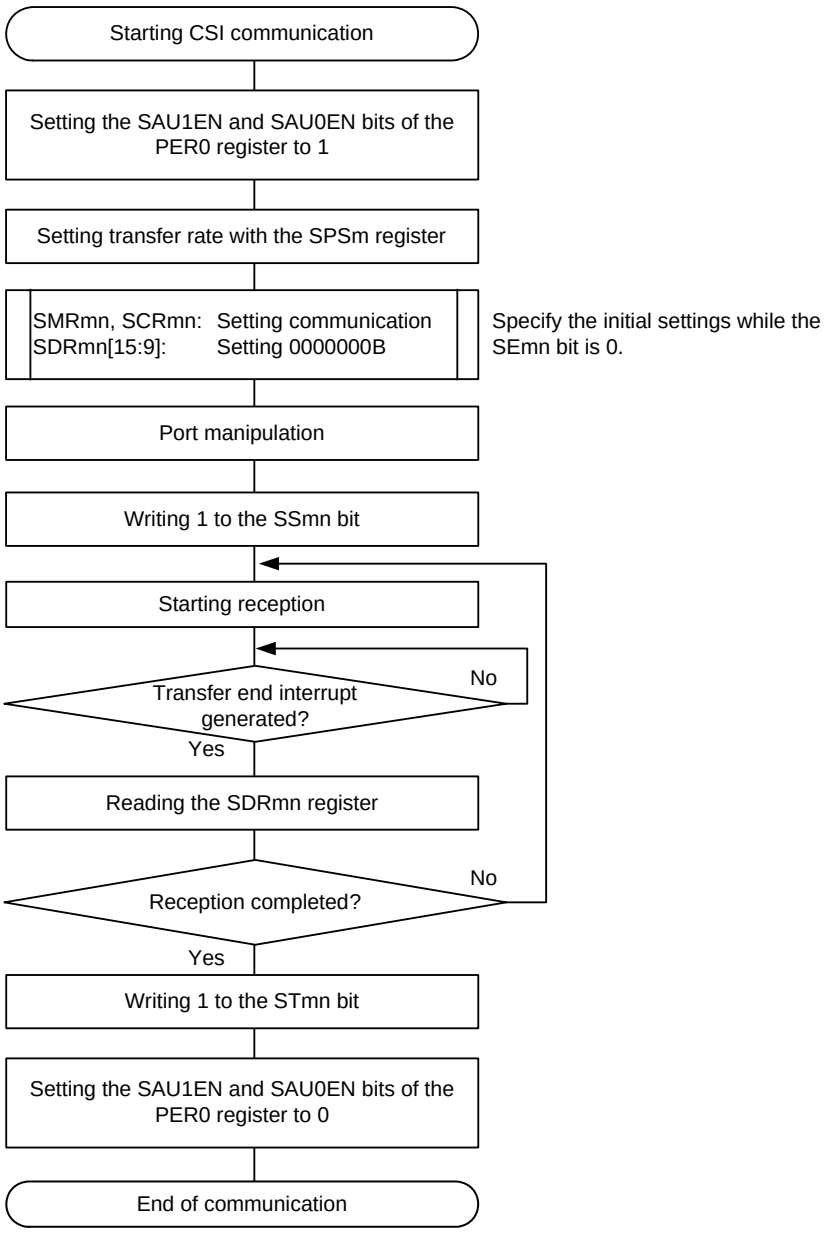
### (3) Processing flow (in single-reception mode)

**Figure 15-110. Timing Chart of Slave Reception (in Single-Reception Mode)**
**(Type 1: DAPmn = 0, CKPmn = 0)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-111. Flowchart of Slave Reception (in Single-Reception Mode)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### 15.6.6 Slave transmission/reception

Slave transmission/reception is an operation wherein this MCU transmits/receives data to/from another device in the state of a transfer clock being input from another device.

| SPI Function | CSI00 | CSI01 | CSI10 | CSI11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | $\overline{\text{SCK00}}$, SI00, $\overline{\text{SO00}}$, $\overline{\text{SSI00}}$ | $\overline{\text{SCK01}}$, SI01, $\overline{\text{SO01}}$, $\overline{\text{SSI01}}$ | $\overline{\text{SCK10}}$, SI10, $\overline{\text{SO10}}$, $\overline{\text{SSI10}}$ | $\overline{\text{SCK11}}$, SI11, $\overline{\text{SO11}}$, $\overline{\text{SSI11}}$ |
| Interrupt | INTCSI00 | INTCSI01 | INTCSI10 | INTCSI11 |
| | Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected. | | | |
| Error detection flag | Overrun error detection flag (OVFmn) only | | | |
| Transfer data length | 7 to 16 bits | | | |
| Transfer rate | Max. $f_{MCK}$/6 [Hz]Notes 1, 2. | | | |
| Data phase | Selectable by the DAPmn bit of the SCRmn register<br>• DAPmn = 0: Data I/O starts from the start of the serial clock operation.<br>• DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation. | | | |
| Clock phase | Selectable by the CKPmn bit of the SCRmn register<br>• CKPmn = 0: Forward<br>• CKPmn = 1: Reverse | | | |
| Data direction | MSB or LSB first | | | |
| SPI function | The operation of the slave select function can be selected. | | | |

**Notes 1.** Because the external serial clock input to the $\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$, and $\overline{\text{SCK11}}$ pins is sampled internally and used, the fastest transfer rate is $f_{MCK}$/6 [Hz].

**2.** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

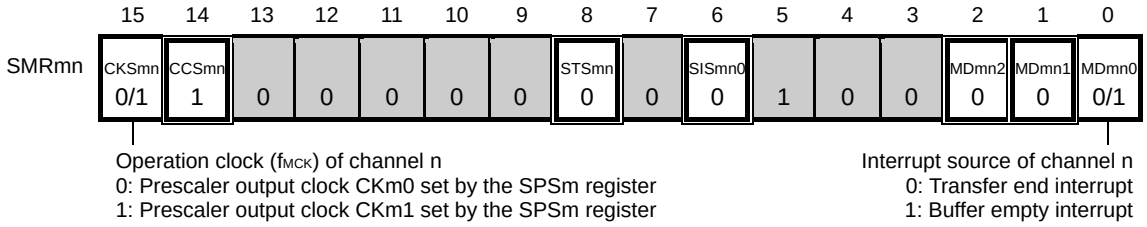
**Remarks 1.** $f_{MCK}$: Operation clock frequency of target channel

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

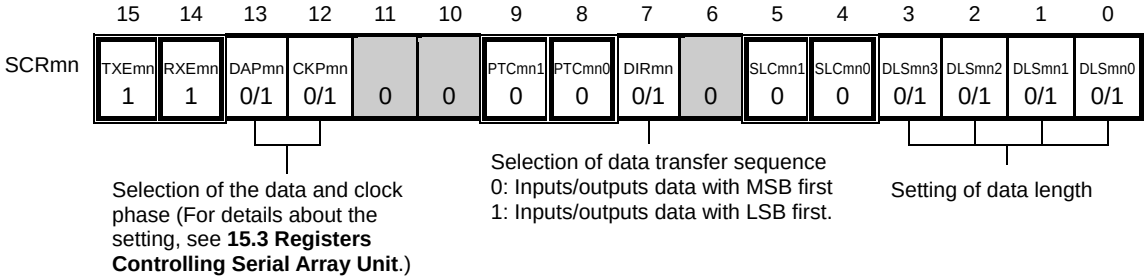
**(1) Register setting**

**Figure 15-112. Example of Contents of Registers for Slave Transmission/Reception of SPI Function (CSI00, CSI01, CSI10, CSI11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

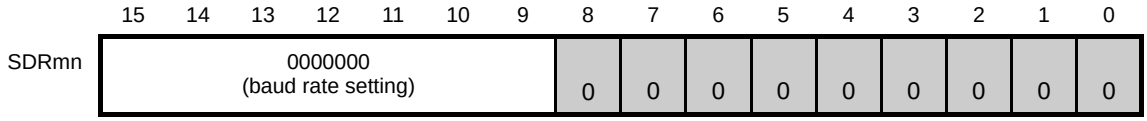



**(b) Serial communication operation setting register mn (SCRmn)**

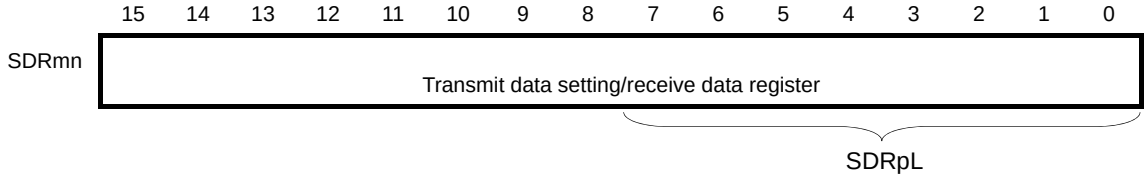



**(c) Serial data register mn (SDRmn)**

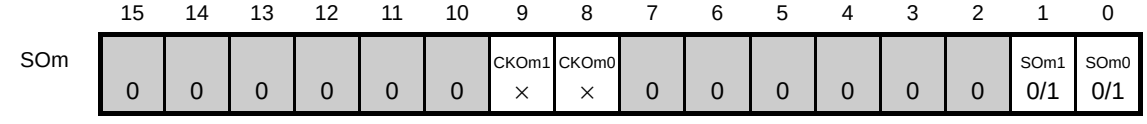
**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)**




**(d) Serial output register m (SOm) … Sets only the bits of the target channel.**




**Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI slave transmission/reception mode

▒: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-112. Example of Contents of Registers for Slave Transmission/Reception of SPI Function (CSI00, CSI01, CSI10, CSI11) (2/2)**

**(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 0/1 | SOEm0 0/1 |

**(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSEm1 0/1 | SSEm0 0/1 |

**(g) Serial slave select enable register m (SSEm) … Controls the $\overline{\text{SSI00}}$, $\overline{\text{SSI01}}$, $\overline{\text{SSI10}}$, and $\overline{\text{SSI11}}$ pin inputs of the target channel in slave mode.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSEm1 0/1 | SSEm0 0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the CSI slave transmission/reception mode

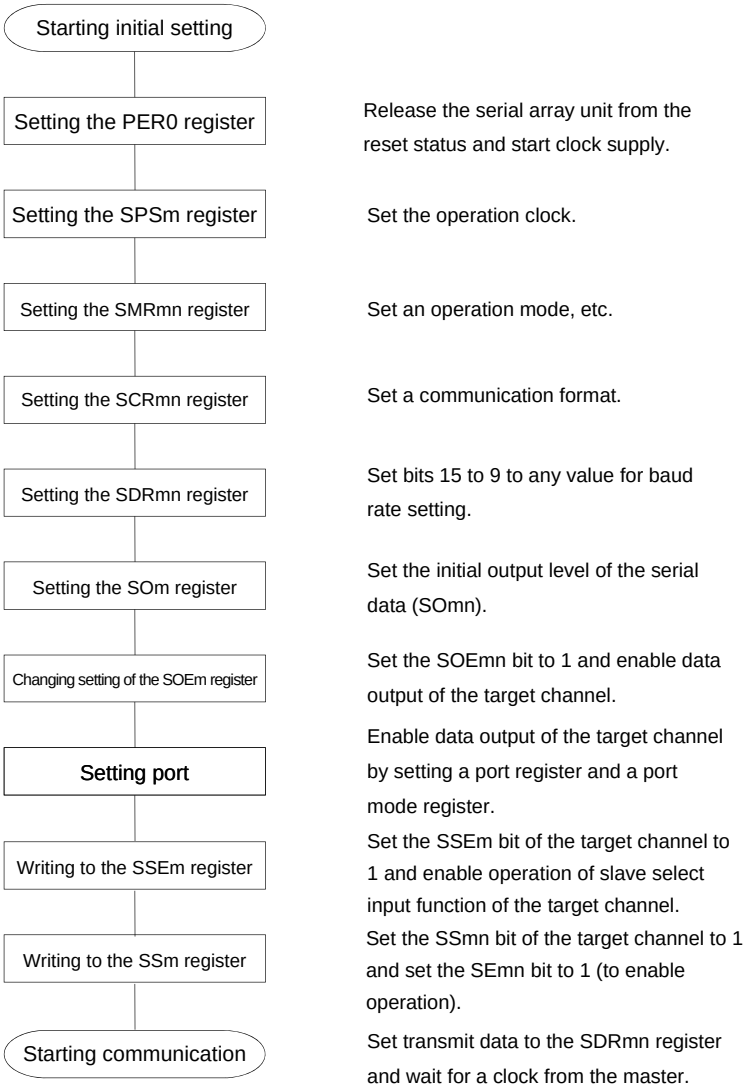
■: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

### (2) Operation procedure

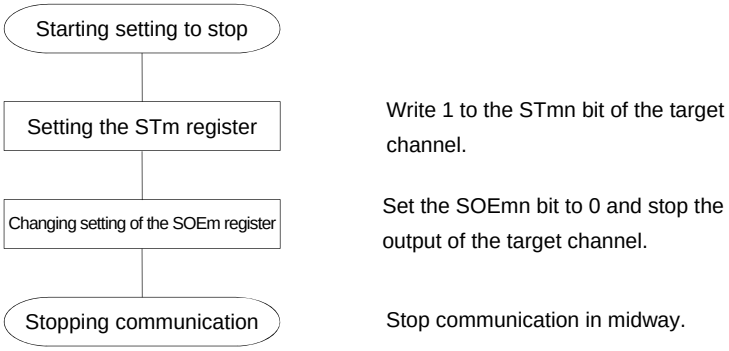
**Figure 15-113. Initial Setting Procedure for Slave Transmission/Reception**




**Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

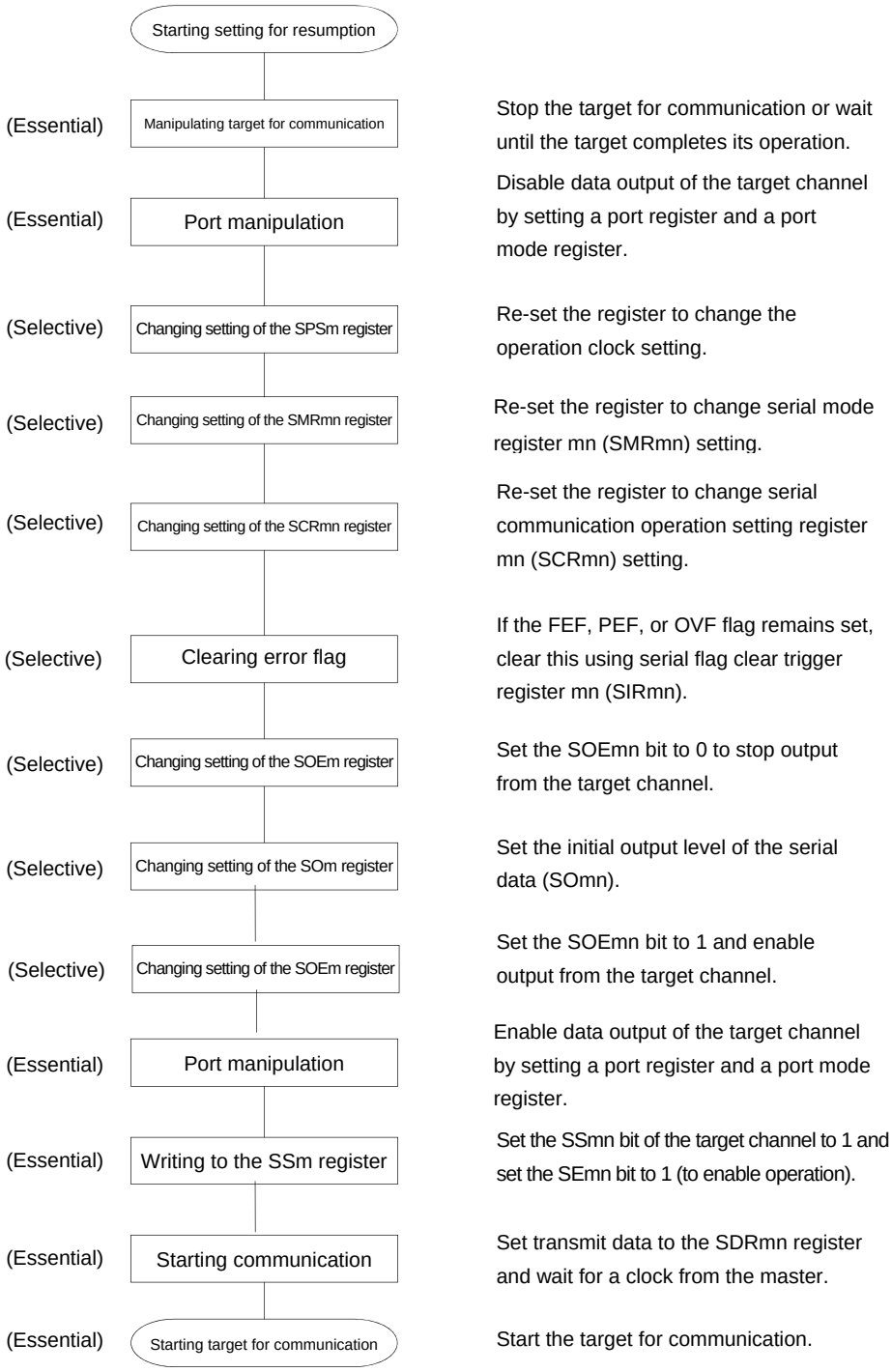
**Figure 15-114. Procedure for Stopping Slave Transmission/Reception**




**Remarks 1.** Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 15-115 Procedure for Resuming Slave Transmission/Reception**).

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-115. Procedure for Resuming Slave Transmission/Reception**

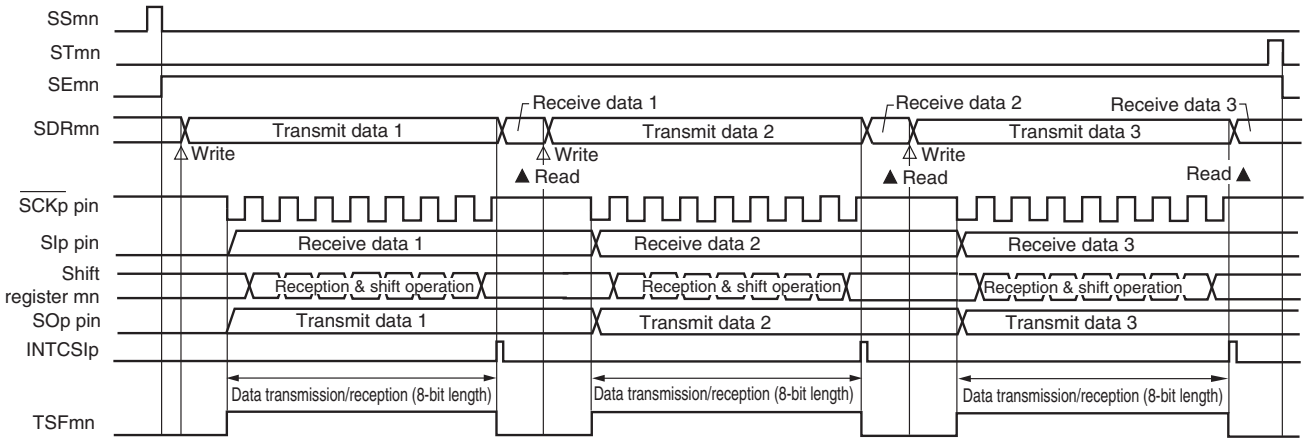



**Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

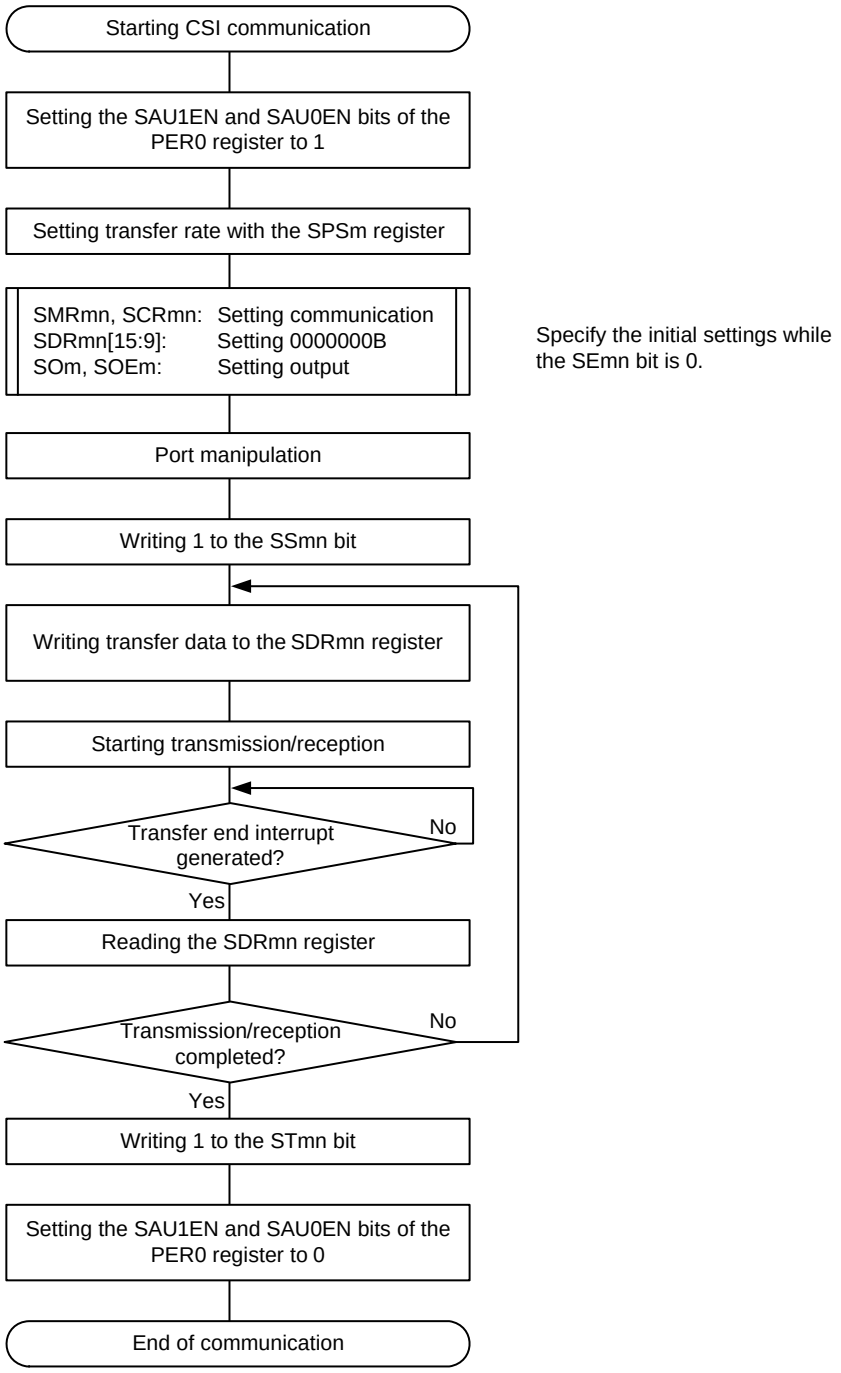
### (3) Processing flow (in single-transmission/reception mode)

**Figure 15-116. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-117. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)**

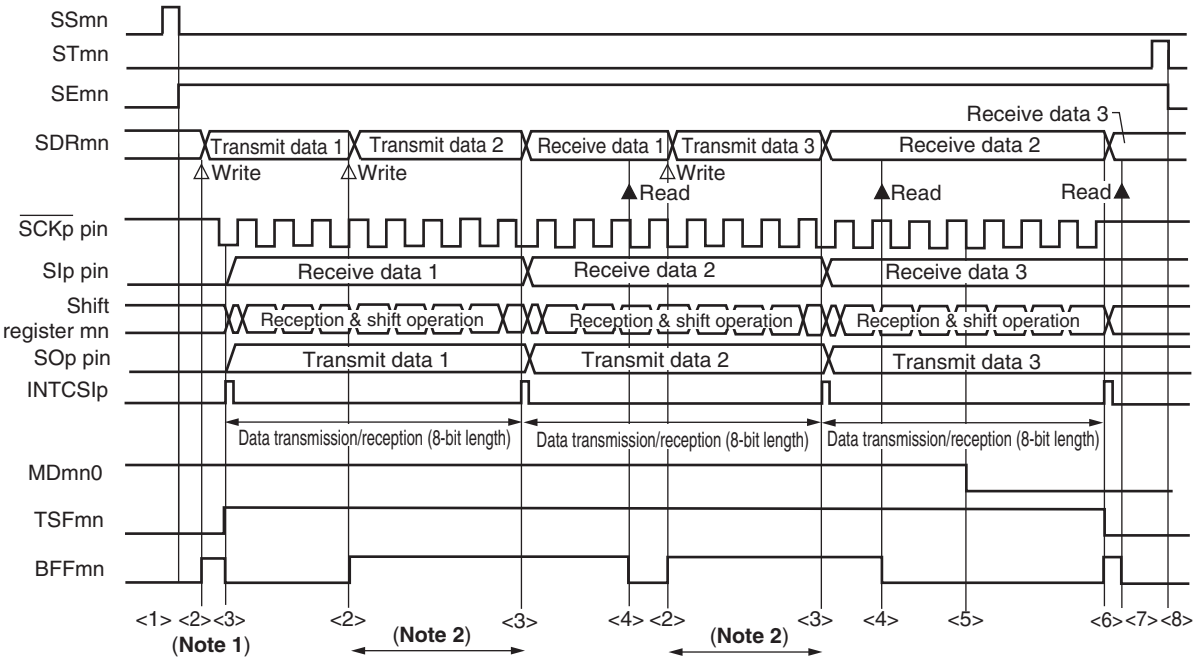



**Cautions 1. Be sure to set transmit data to the SDRpL register before the clock from the master is started.**

**2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11**

### (4) Processing flow (in continuous transmission/reception mode)

**Figure 15-118. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**




**Notes 1.** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

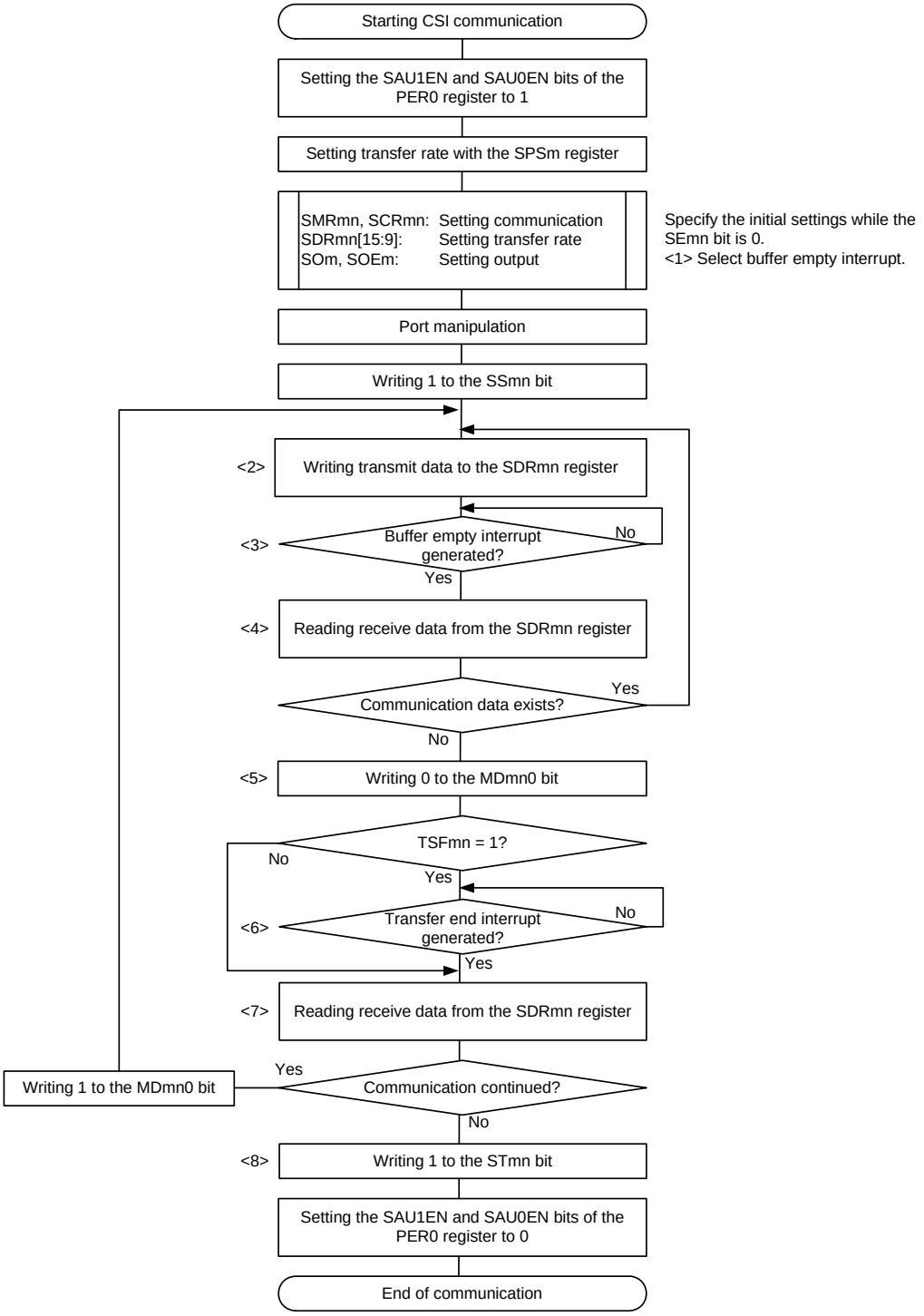
**2.** The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

**Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.**

**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-119 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-119. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**




**Caution Be sure to set transmit data to the SDRpL register before the clock from the master is started.**

**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 15-118 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

### 15.6.7 Calculating transfer clock frequency

The transfer clock frequency for SPI function (CSI00, CSI01, CSI10, CSI11) communication can be calculated by the following expressions.

**(1) Master**

(Transfer clock frequency) = {Operation clock ($f_{MCK}$) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

**(2) Slave**

(Transfer clock frequency) = {Frequency of serial clock ($f_{SCK}$) supplied by master}[Note] [Hz]

**Note** The permissible maximum transfer clock frequency is $f_{MCK}$/6.

**Remark** 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of serial data register mn (SDRmn) (0000000B to 1111111B) and therefore is 0 to 127.

**2.** The operation clock ($f_{MCK}$) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

**3.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Table 15-3. Selection of Operation Clock For SPI Function**

| SMRmn Register | SPSm Register | | | | | | | | Operation Clock ($f_{MCK}$) Note | |
|---|---|---|---|---|---|---|---|---|---|---|
| CKSmn | PRS m13 | PRS m12 | PRS m11 | PRS m10 | PRS m03 | PRS m02 | PRS m01 | PRS m00 | | $f_{CLK}$ = 32 MHz |
| 0 | X | X | X | X | 0 | 0 | 0 | 0 | $f_{CLK}$ | 32 MHz |
| | X | X | X | X | 0 | 0 | 0 | 1 | $f_{CLK}/2$ | 16 MHz |
| | X | X | X | X | 0 | 0 | 1 | 0 | $f_{CLK}/2^2$ | 8 MHz |
| | X | X | X | X | 0 | 0 | 1 | 1 | $f_{CLK}/2^3$ | 4 MHz |
| | X | X | X | X | 0 | 1 | 0 | 0 | $f_{CLK}/2^4$ | 2 MHz |
| | X | X | X | X | 0 | 1 | 0 | 1 | $f_{CLK}/2^5$ | 1 MHz |
| | X | X | X | X | 0 | 1 | 1 | 0 | $f_{CLK}/2^6$ | 500 kHz |
| | X | X | X | X | 0 | 1 | 1 | 1 | $f_{CLK}/2^7$ | 250 kHz |
| | X | X | X | X | 1 | 0 | 0 | 0 | $f_{CLK}/2^8$ | 125 kHz |
| | X | X | X | X | 1 | 0 | 0 | 1 | $f_{CLK}/2^9$ | 62.5 kHz |
| | X | X | X | X | 1 | 0 | 1 | 0 | $f_{CLK}/2^{10}$ | 31.25 kHz |
| | X | X | X | X | 1 | 0 | 1 | 1 | $f_{CLK}/2^{11}$ | 15.63 kHz |
| 1 | 0 | 0 | 0 | 0 | X | X | X | X | $f_{CLK}$ | 32 MHz |
| | 0 | 0 | 0 | 1 | X | X | X | X | $f_{CLK}/2$ | 16 MHz |
| | 0 | 0 | 1 | 0 | X | X | X | X | $f_{CLK}/2^2$ | 8 MHz |
| | 0 | 0 | 1 | 1 | X | X | X | X | $f_{CLK}/2^3$ | 4 MHz |
| | 0 | 1 | 0 | 0 | X | X | X | X | $f_{CLK}/2^4$ | 2 MHz |
| | 0 | 1 | 0 | 1 | X | X | X | X | $f_{CLK}/2^5$ | 1 MHz |
| | 0 | 1 | 1 | 0 | X | X | X | X | $f_{CLK}/2^6$ | 500 kHz |
| | 0 | 1 | 1 | 1 | X | X | X | X | $f_{CLK}/2^7$ | 250 kHz |
| | 1 | 0 | 0 | 0 | X | X | X | X | $f_{CLK}/2^8$ | 125 kHz |
| | 1 | 0 | 0 | 1 | X | X | X | X | $f_{CLK}/2^9$ | 62.5 kHz |
| | 1 | 0 | 1 | 0 | X | X | X | X | $f_{CLK}/2^{10}$ | 31.25 kHz |
| | 1 | 0 | 1 | 1 | X | X | X | X | $f_{CLK}/2^{11}$ | 15.63 kHz |
| Other than above | | | | | | | | | | Setting prohibited |

**Note** When changing the clock selected for $f_{CLK}$ (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 0003H) the operation of the serial array unit (SAU).

**Remarks 1.** X: Don't care

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### 15.6.8 Procedure for processing errors that occurred during clock synchronous serial communication with SPI function

The procedure for processing errors that occurred during clock synchronous serial communication with SPI function is described in Figure 15-120.

**Figure 15-120. Processing Procedure in Case of Overrun Error**

| Software Manipulation | Hardware Status | Remark |
|---|---|---|
| Reads serial data register mn (SDRmn). → | The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data. | This is to prevent an overrun error if the next reception is completed during error processing. |
| Reads serial status register mn (SSRmn). | | Error type is identified and the read value is used to clear error flag. |
| Writes 1 to serial flag clear trigger register mn (SIRmn). → | Error flag is cleared. | Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification. |

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

## 15.7 Operation of UART (UART0, UART1) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can be implemented by using timer array unit 0 with an external interrupt (INTP0).

[Data transmission/reception]

- Data length of 7, 8, 9, or 16 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Framing error, parity error, or overrun error

The LIN-bus is accepted in UART0 (channels 0 and 1 of unit 0).

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit 0

UART0 uses channels 0 and 1 of SAU0.
UART1 uses channels 0 and 1 of SAU1.

- Group A products

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note2 | | IIC01 |

- Products of Groups C-1 and D-1

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note2 | | IIC01 |
| 1 | 0 | CSI10 (supporting SPI function) Note1, 2 | UART1 | IIC10 |
| | 1 | - | | - |

- Products of Groups B, C-2, D-2, and E

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note2 | | IIC01 |
| 1 | 0 | CSI10 (supporting SPI function) Note1, 2 | UART1 | IIC10 |
| | 1 | CSI11 (supporting SPI function) Note2 | | IIC11 |

**Notes 1.** 48-pin, 32-pin and 30-pin products do not have $\overline{\text{SSI10}}$ pin.

<R> **2.** Set CKPmn bit of SCRmn register to 1, when SSEmn = 1 (Enables $\overline{\text{SSImn}}$ pin input).
(m = 0, 1, n = 0, 1)

**Caution When using serial array unit as UARTs, the channels of both the transmitting side (even-number channel) and the receiving side (odd-number channel) can be used only as UARTs.**

**Remark** Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory
Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory
Group C-1: RL78/F13 (CAN and LIN incorporated) products with 30 or 32 pins
Group C-2: RL78/F13 (CAN and LIN incorporated) products with 48, 64, or 80 pins
Group D-1: RL78/F14 products with 30 or 32 pins
Group D-2: RL78/F14 products with 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory
Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

UART performs the following four types of communication operations.

- UART transmission (See **15.7.1 UART transmission**.)
- UART reception (See **15.7.2 UART reception**.)
- LIN transmission (UART0 only) (See **15.8.1 LIN transmission**.)
- LIN reception (UART0 only) (See **15.8.2 LIN reception**.)

### 15.7.1 UART transmission

UART transmission is an operation to transmit data from this MCU to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

| UART | UART0 | UART1 |
|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 2 of SAU0 |
| Pins used | TxD0 | TxD1 |
| Interrupt | INTST0 | INTST1 |
| | Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected. | |
| Error detection flag | None | |
| Transfer data length | 7 to 9 or 16 bits | |
| Transfer rate | Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] **Note** | |
| Data phase | Forward output (default: high level)<br>Reverse output (default: low level) | |
| Parity bit | The following selectable<br>• No parity bit<br>• Appending 0 parity<br>• Appending even parity<br>• Appending odd parity | |
| Stop bit | The following selectable<br>• Appending 1 bit<br>• Appending 2 bits | |
| Data direction | MSB or LSB first | |

**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

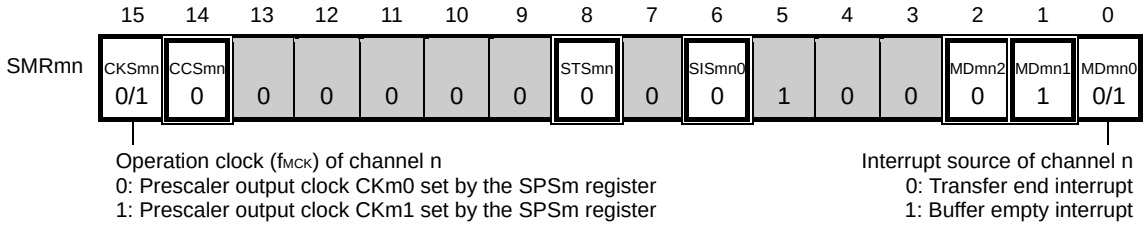
**Remarks** **1.** $f_{MCK}$: Operation clock frequency of target channel
$f_{CLK}$: System clock frequency

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 01), mn = 00, 10

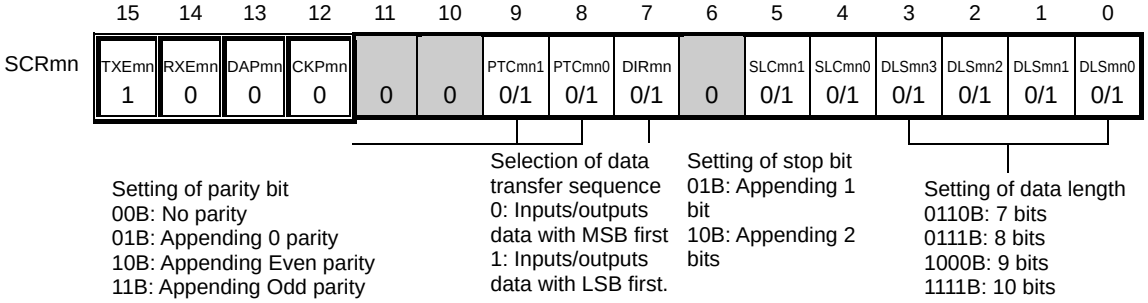
### (1) Register setting

**Figure 15-121. Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (1/2)**

**(a) Serial mode register mn (SMRmn)**

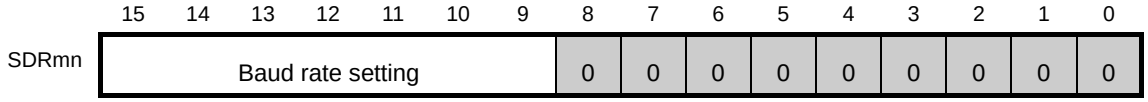



**(b) Serial communication operation setting register mn (SCRmn)**

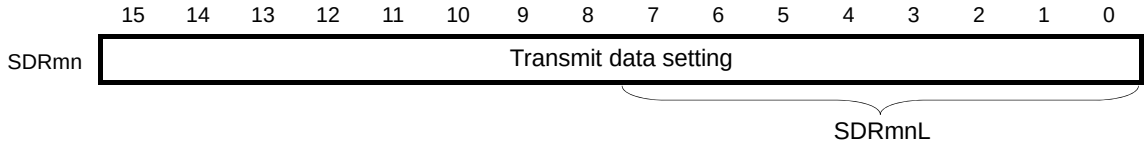



**(c) Serial data register mn (SDRmn)**

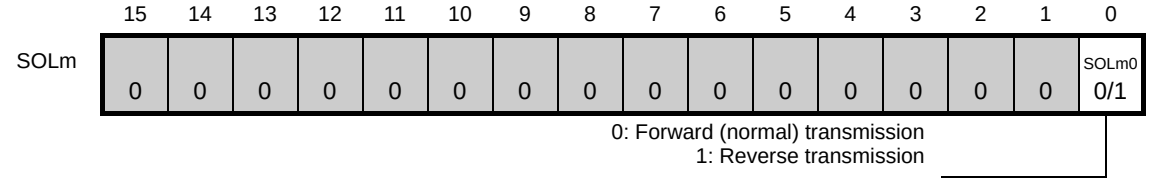
**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRmnL)**




**(d) Serial output level register m (SOLm) … Sets only the bits of the target channel.**




**Note** Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

**2.** ☐: Setting is fixed in the UART transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-121. Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (2/2)**

**(e) Serial output register m (SOm) … Sets only the bits of the target channel.**

SOm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | CKOm1 × | CKOm0 × | 0 | 0 | 0 | 0 | 0 | 0 | SOm1 × | SOm0 0/1 Note |

0: Serial data output value is 0
1: Serial data output value is 1

**(f) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.**

SOEm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 × | SOEm0 0/1 |

**(g) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

SSm

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1 × | SSm0 0/1 |

**Note** Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

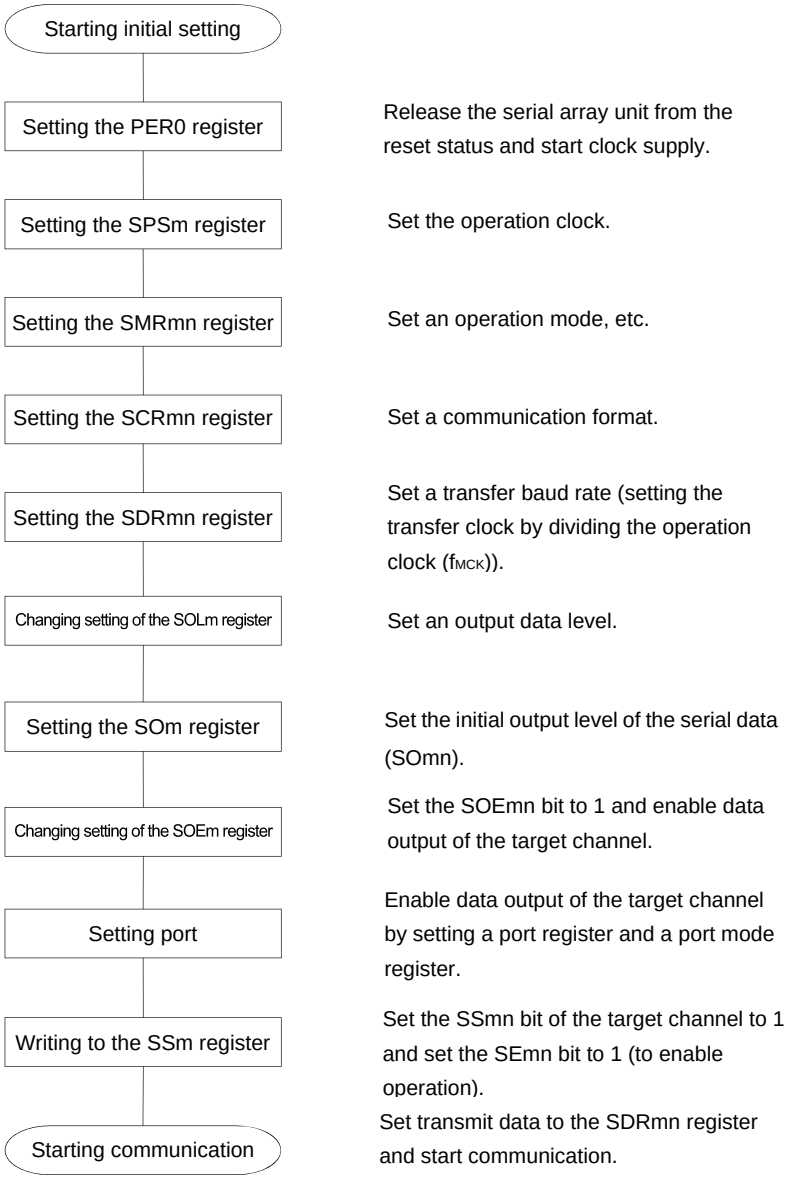
**2.** ☐ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

### (2) Operation procedure

**Figure 15-122. Initial Setting Procedure for UART Transmission**



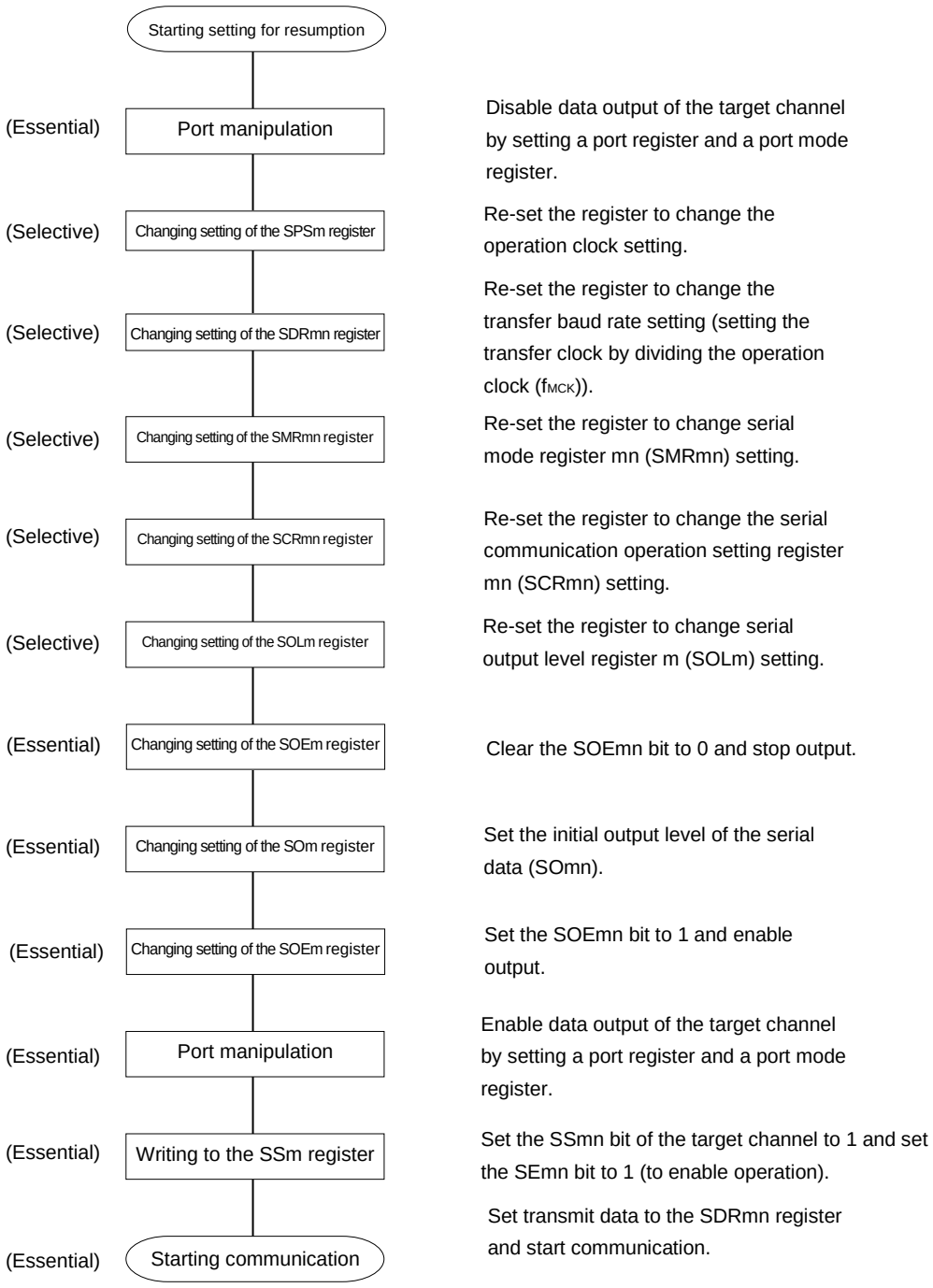

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

**Figure 15-123. Procedure for Stopping UART Transmission**

**Remarks 1.** Even after communication is stopped, the pin level is retained. To resume the operation, re-set serial output register m (SOm) (see **Figure 15-124 Procedure for Resuming UART Transmission**).

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

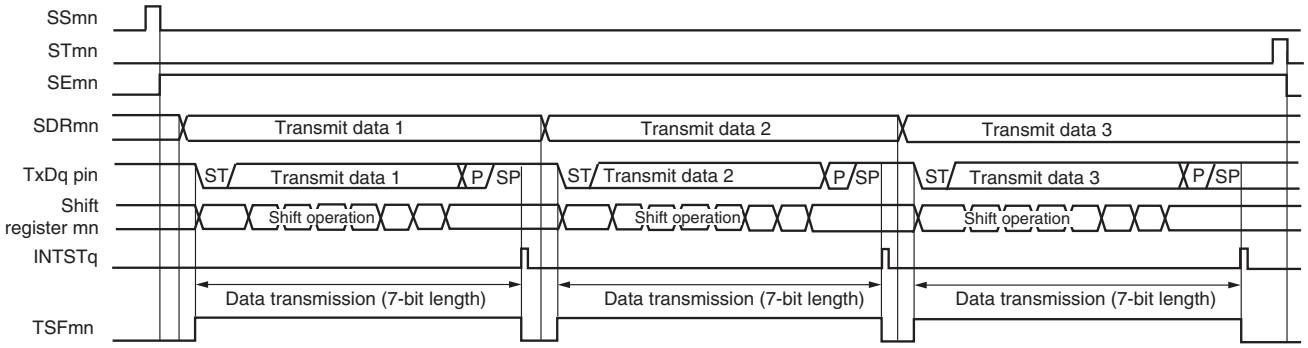
**Figure 15-124. Procedure for Resuming UART Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

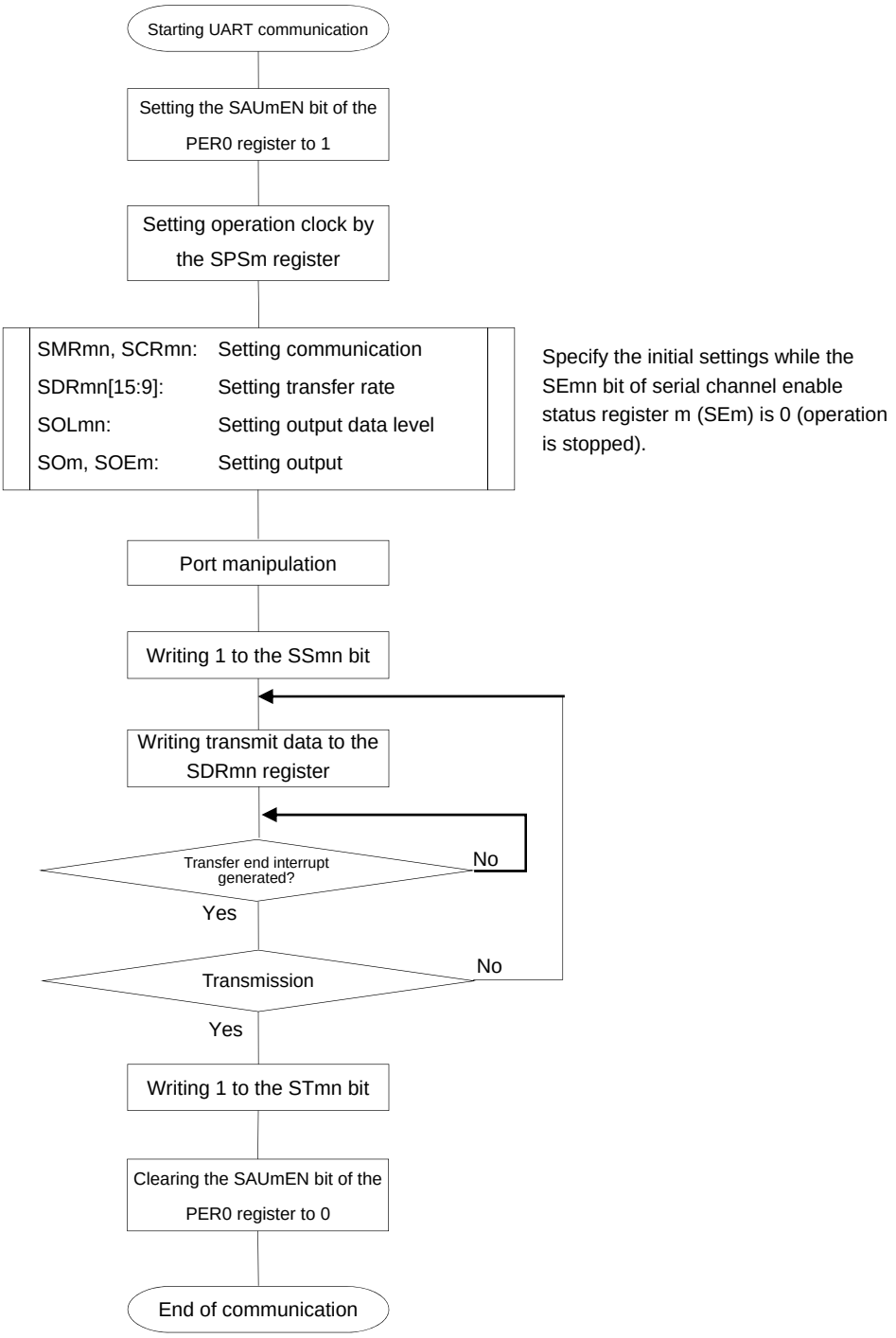
### (3) Processing flow (in single-transmission mode)

**Figure 15-125. Timing Chart of UART Transmission (in Single-Transmission Mode)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0), q: UART number (q = 0, 1)
mn = 00, 10

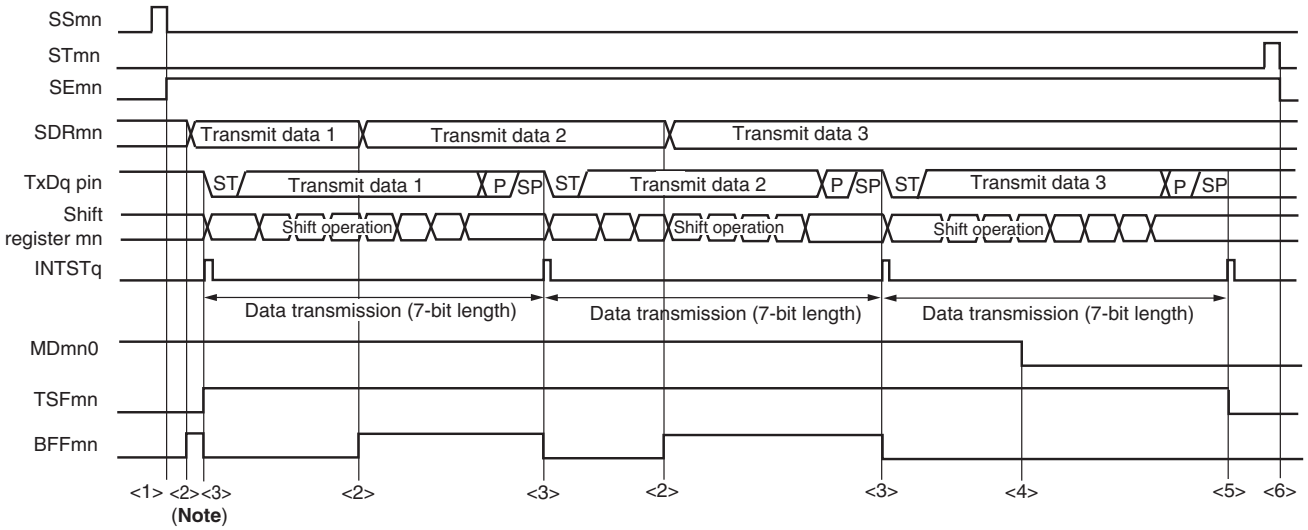
**Figure 15-126. Flowchart of UART Transmission (in Single-Transmission Mode)**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

### (4) Processing flow (in continuous transmission mode)

**Figure 15-127. Timing Chart of UART Transmission (in Continuous Transmission Mode)**

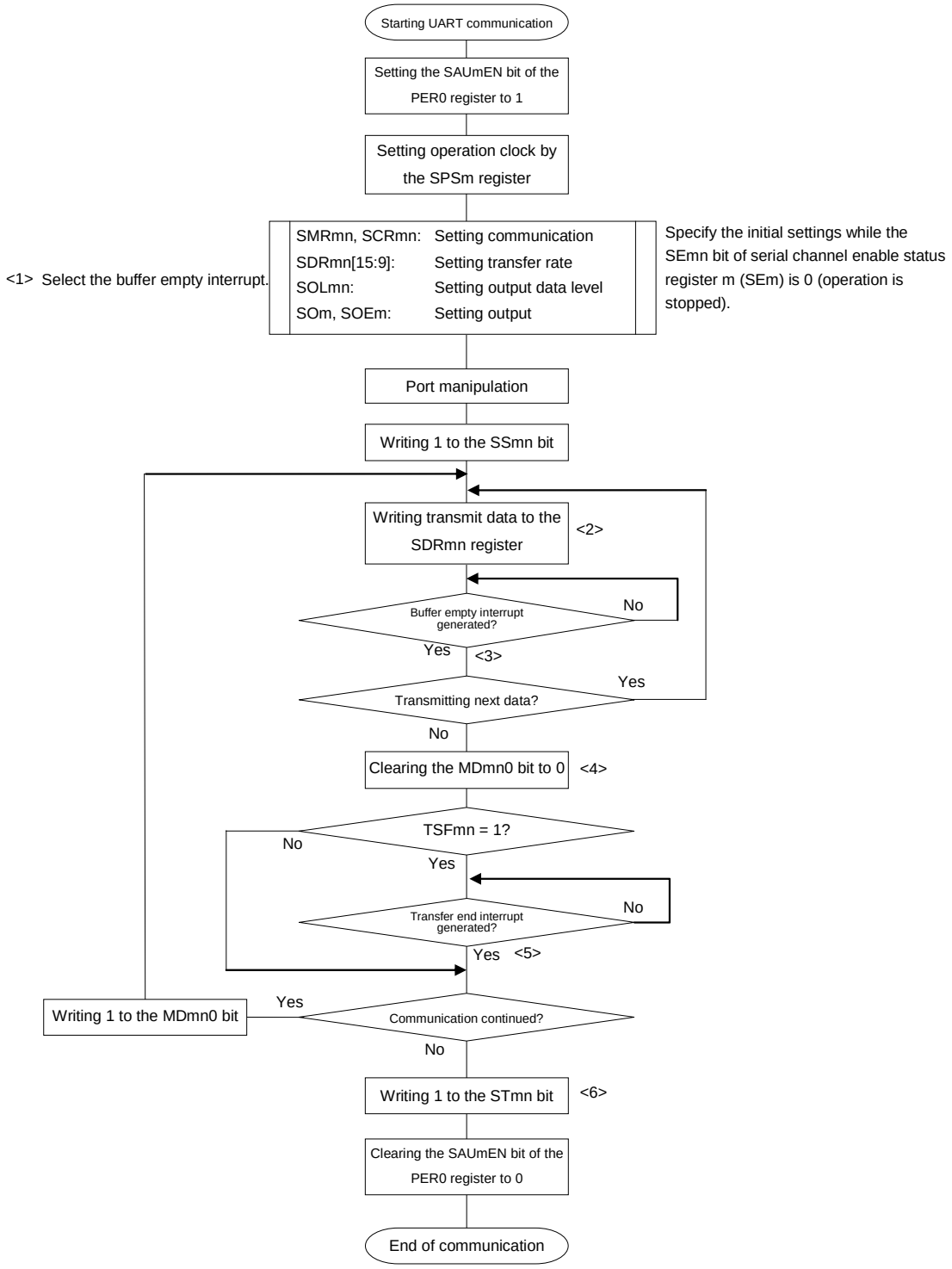



**Note** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

**Caution The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0), q: UART number (q = 0, 1)
mn = 00, 10

**Figure 15-128. Flowchart of UART Transmission (in Continuous Transmission Mode)**




**Remarks 1.** <1> to <6> in the figure correspond to <1> to <6> in **Figure 15-127 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 00, 10

### 15.7.2 UART reception

UART reception is an operation wherein this MCU asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMR register of both the odd- and even-numbered channels must be set.

| UART | UART0 | UART1 |
|---|---|---|
| Target channel | Channel 1 of SAU0 | Channel 1 of SAU1 |
| Pins used | RxD0 | RxD1 |
| Interrupt | INTSR0 | INTSR1 |
| | Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.) | |
| Error detection flag | • Framing error detection flag (FEFmn)<br>• Parity error detection flag (PEFmn)<br>• Overrun error detection flag (OVFmn) | |
| Transfer data length | 7 to 9 or 16 bits | |
| Transfer rate | Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 3 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] Note | |
| Data phase | Forward output (default: high level)<br>Reverse output (default: low level) | |
| Parity bit | The following selectable<br>• No parity bit (no parity check)<br>• Appending 0 parity (no parity check)<br>• Appending even parity<br>• Appending odd parity | |
| Stop bit | Appending 1 bit | |
| Data direction | MSB or LSB first | |

**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

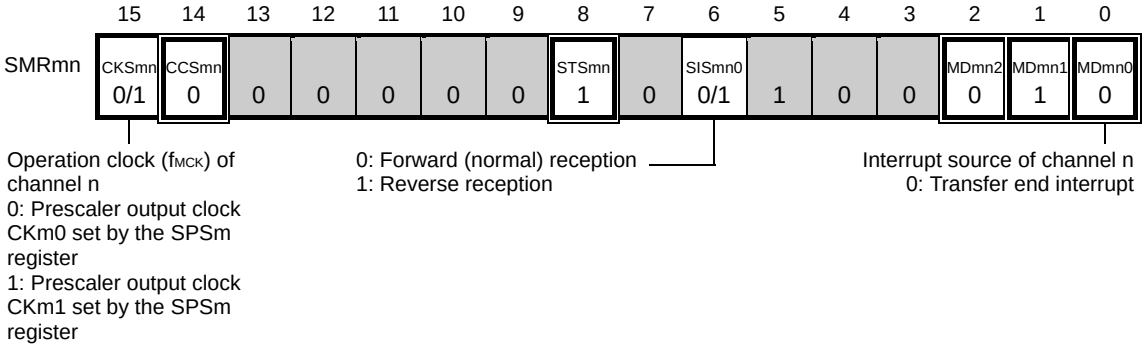
**Remarks 1.** $f_{MCK}$: Operation clock frequency of target channel
$f_{CLK}$: System clock frequency

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 01, 11

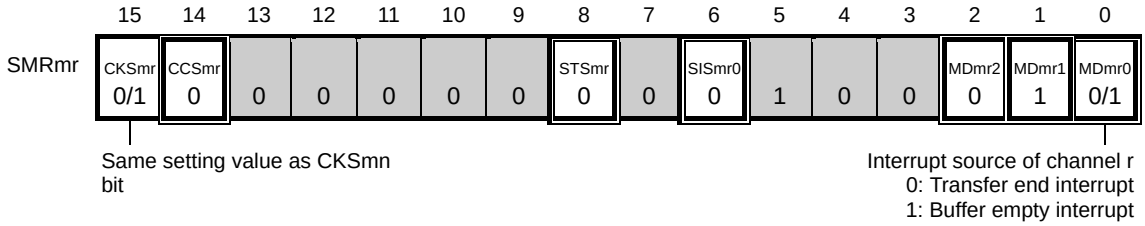
**(1) Register setting**

**Figure 15-129. Example of Contents of Registers for UART Reception of UART (UART0, UART1) (1/2)**

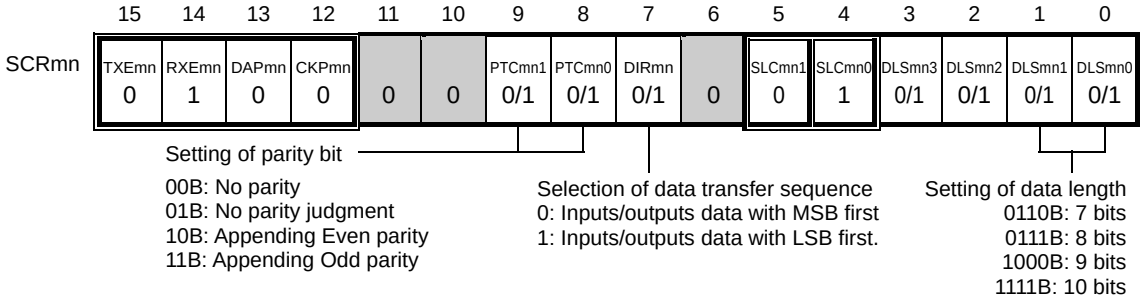
**(a) Serial mode register mn (SMRmn)**




**(b) Serial mode register mr (SMRmr)**

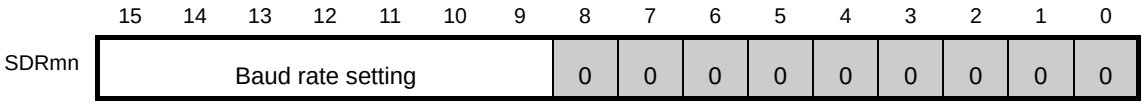



**(c) Serial communication operation setting register mn (SCRmn)**

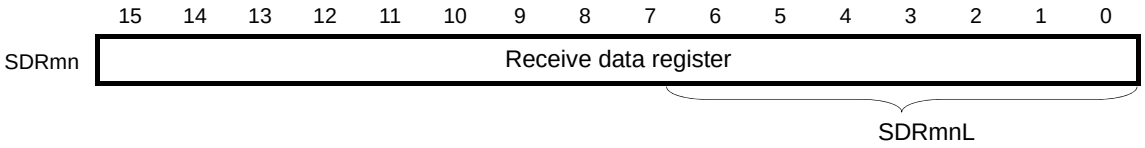



**(d) Serial data register mn (SDRmn)**

**(1) When operation is stopped (SEmn = 0)**




**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRmnL)**




**Caution For the UART reception, be sure to set the SMRmr register of channel r that is to be paired with channel n.**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 01, 11
r: Channel number (r = n – 1), q: UART number (q = 0, 1)

**2.** ☐ : Setting is fixed in the UART reception mode, ▒ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-129. Example of Contents of Registers for UART Reception of UART (UART0, UART1) (2/2)**

**(e) Serial output register m (SOm) … The register that not used in this mode.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOm | 0 | 0 | 0 | 0 | 0 | 0 | CKOm1 × | CKOm0 × | 0 | 0 | 0 | 0 | 0 | 0 | SOm1 × | SOm0 × |

**(f) Serial output enable register m (SOEm) …The register that not used in this mode.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 × | SOEm0 × |

**(g) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1 0/1 | SSm0 × |

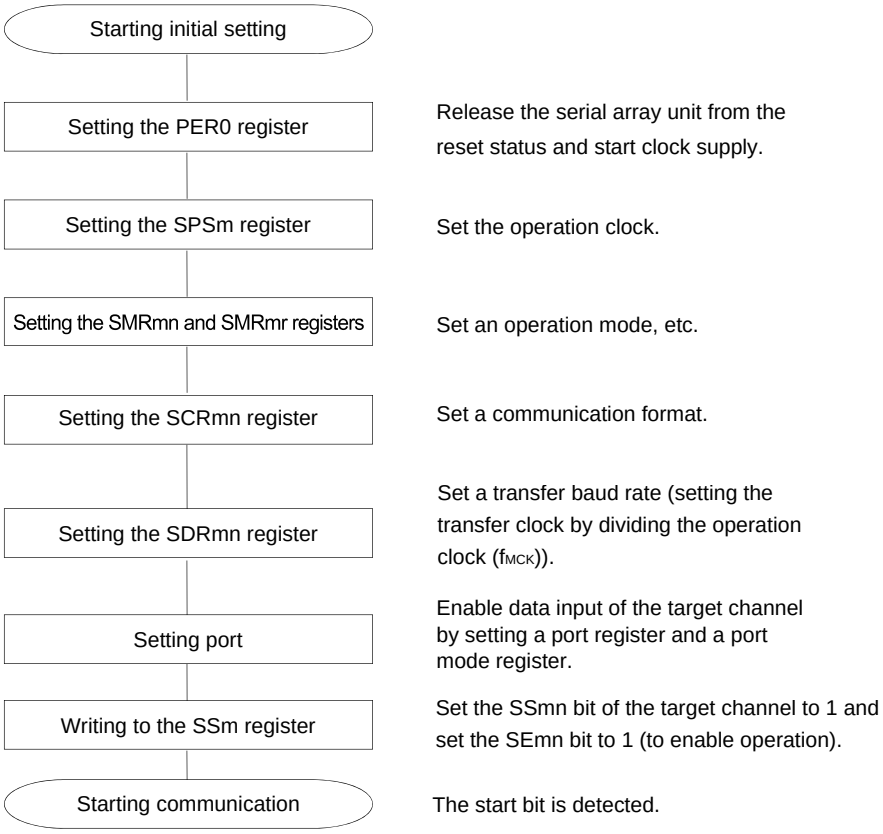
**Caution For the UART reception, be sure to set the SMRmr register of channel r that is to be paired with channel n.**

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 01, 11
r: Channel number (r = n – 1), q: UART number (q = 0, 1)

**2.** ▢ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

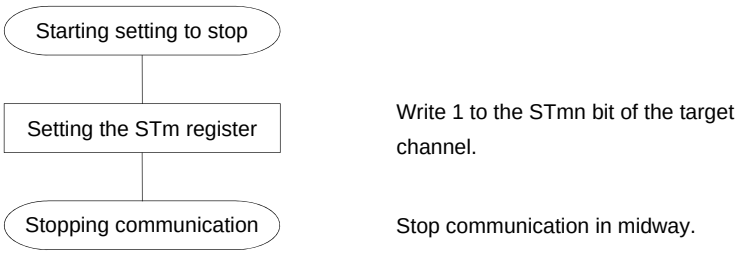
### (2) Operation procedure

**Figure 15-130. Initial Setting Procedure for UART Reception**



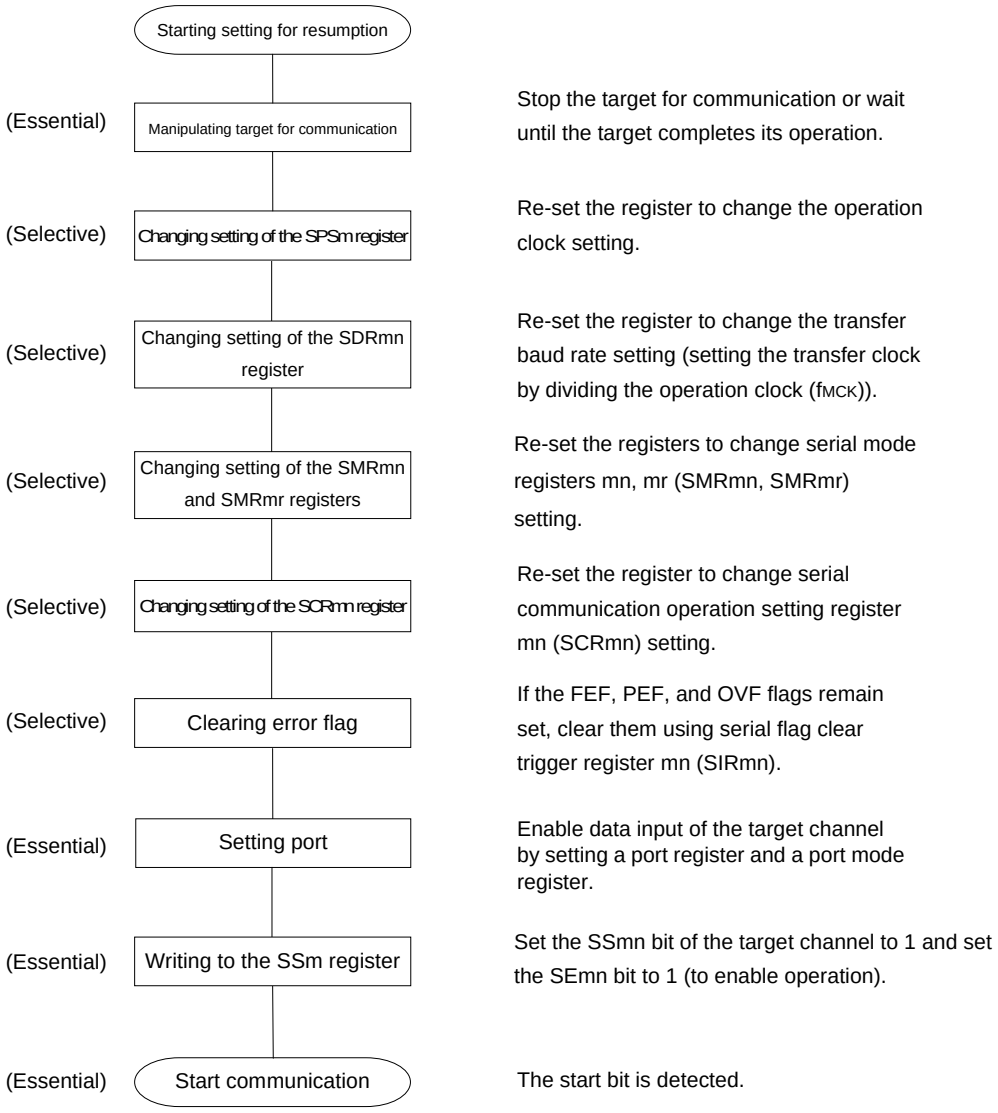

**Caution For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more $f_{MCK}$ clocks have elapsed.**

**Figure 15-131. Procedure for Stopping UART Reception**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 00, 10, r: Channel number (r = n – 1)

**Figure 15-132. Procedure for Resuming UART Reception**

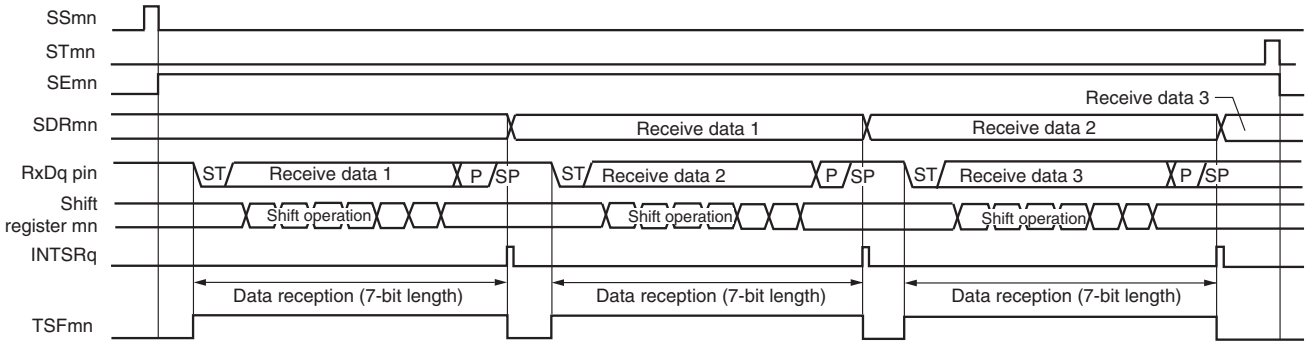



**Caution For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more $f_{MCK}$ clocks have elapsed.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 00, 10, r: Channel number (r = n – 1)

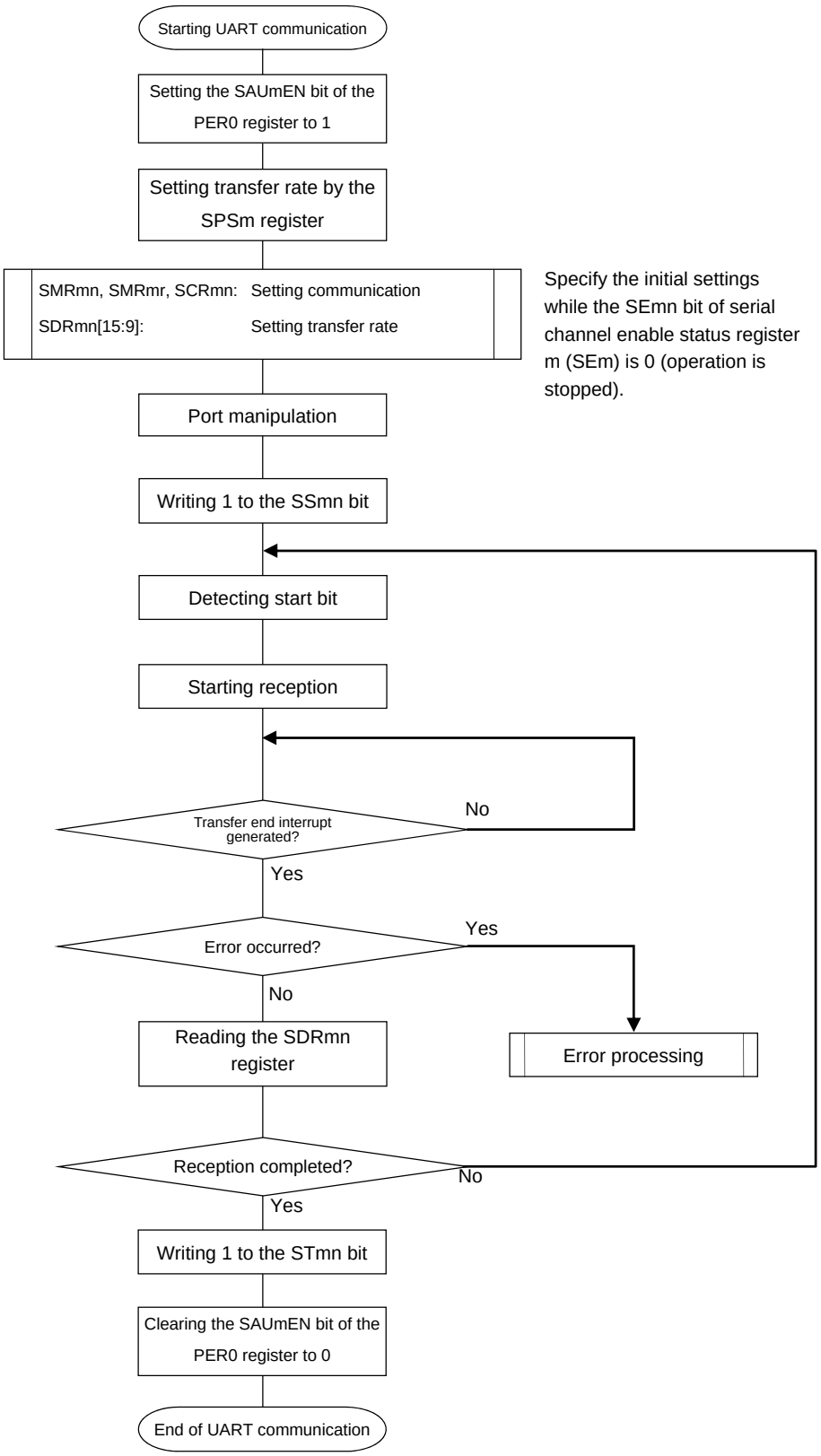
### (3) Processing flow

**Figure 15-133. Timing Chart of UART Reception**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 01, 11, r: Channel number (r = n – 1), q: UART number (q = 0, 1)

**Figure 15-134. Flowchart of UART Reception**




**Caution For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more $f_{MCK}$ clocks have elapsed.**

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 01, 11, r: Channel number (r = n – 1)

### 15.7.3 Calculating baud rate

#### (1) Baud rate calculation expression

The baud rate for UART (UART0, UART1) communication can be calculated by the following expressions.

| (Baud rate) = {Operation clock ($f_{MCK}$) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps] |
|---|

**Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.**

**Remarks 1.** When UART is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B) and therefore is 2 to 127.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

The operation clock ($f_{MCK}$) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

**Table 15-4. Selection of Operation Clock For UART**

| SMRmn Register | SPSm Register | | | | | | | | Operation Clock ($f_{MCK}$) Note | |
|---|---|---|---|---|---|---|---|---|---|---|
| CKSmn | PRS m13 | PRS m12 | PRS m11 | PRS m10 | PRS m03 | PRS m02 | PRS m01 | PRS m00 | | $f_{CLK}$ = 32 MHz |
| 0 | X | X | X | X | 0 | 0 | 0 | 0 | $f_{CLK}$ | 32 MHz |
| | X | X | X | X | 0 | 0 | 0 | 1 | $f_{CLK}/2$ | 16 MHz |
| | X | X | X | X | 0 | 0 | 1 | 0 | $f_{CLK}/2^2$ | 8 MHz |
| | X | X | X | X | 0 | 0 | 1 | 1 | $f_{CLK}/2^3$ | 4 MHz |
| | X | X | X | X | 0 | 1 | 0 | 0 | $f_{CLK}/2^4$ | 2 MHz |
| | X | X | X | X | 0 | 1 | 0 | 1 | $f_{CLK}/2^5$ | 1 MHz |
| | X | X | X | X | 0 | 1 | 1 | 0 | $f_{CLK}/2^6$ | 500 kHz |
| | X | X | X | X | 0 | 1 | 1 | 1 | $f_{CLK}/2^7$ | 250 kHz |
| | X | X | X | X | 1 | 0 | 0 | 0 | $f_{CLK}/2^8$ | 125 kHz |
| | X | X | X | X | 1 | 0 | 0 | 1 | $f_{CLK}/2^9$ | 62.5 kHz |
| | X | X | X | X | 1 | 0 | 1 | 0 | $f_{CLK}/2^{10}$ | 31.25 kHz |
| | X | X | X | X | 1 | 0 | 1 | 1 | $f_{CLK}/2^{11}$ | 15.63 kHz |
| 1 | 0 | 0 | 0 | 0 | X | X | X | X | $f_{CLK}$ | 32 MHz |
| | 0 | 0 | 0 | 1 | X | X | X | X | $f_{CLK}/2$ | 16 MHz |
| | 0 | 0 | 1 | 0 | X | X | X | X | $f_{CLK}/2^2$ | 8 MHz |
| | 0 | 0 | 1 | 1 | X | X | X | X | $f_{CLK}/2^3$ | 4 MHz |
| | 0 | 1 | 0 | 0 | X | X | X | X | $f_{CLK}/2^4$ | 2 MHz |
| | 0 | 1 | 0 | 1 | X | X | X | X | $f_{CLK}/2^5$ | 1 MHz |
| | 0 | 1 | 1 | 0 | X | X | X | X | $f_{CLK}/2^6$ | 500 kHz |
| | 0 | 1 | 1 | 1 | X | X | X | X | $f_{CLK}/2^7$ | 250 kHz |
| | 1 | 0 | 0 | 0 | X | X | X | X | $f_{CLK}/2^8$ | 125 kHz |
| | 1 | 0 | 0 | 1 | X | X | X | X | $f_{CLK}/2^9$ | 62.5 kHz |
| | 1 | 0 | 1 | 0 | X | X | X | X | $f_{CLK}/2^{10}$ | 31.25 kHz |
| | 1 | 0 | 1 | 1 | X | X | X | X | $f_{CLK}/2^{11}$ | 15.63 kHz |
| Other than above | | | | | | | | | | Setting prohibited |

**Note** When changing the clock selected for $f_{CLK}$ (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 0003H) the operation of the serial array unit (SAU).

**Remarks 1.** X: Don't care

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

**(2) Baud rate error during transmission**

The baud rate error of UART (UART0, UART1) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 – 100 [%]

Here is an example of setting a UART baud rate at $f_{CLK}$ = 32 MHz.

| UART Baud Rate (Target Baud Rate) | $f_{CLK}$ = 32 MHz | | | |
|---|---|---|---|---|
| | Operation Clock ($f_{MCK}$) | SDRmn[15:9] | Calculated Baud Rate | Error from Target Baud Rate |
| 300 bps | $f_{CLK}/2^9$ | 103 | 300.48 bps | +0.16 % |
| 600 bps | $f_{CLK}/2^8$ | 103 | 600.96 bps | +0.16 % |
| 1200 bps | $f_{CLK}/2^7$ | 103 | 1201.92 bps | +0.16 % |
| 2400 bps | $f_{CLK}/2^6$ | 103 | 2403.85 bps | +0.16 % |
| 4800 bps | $f_{CLK}/2^5$ | 103 | 4807.69 bps | +0.16 % |
| 9600 bps | $f_{CLK}/2^4$ | 103 | 9615.38 bps | +0.16 % |
| 19200 bps | $f_{CLK}/2^3$ | 103 | 19230.8 bps | +0.16 % |
| 31250 bps | $f_{CLK}/2^3$ | 63 | 31250.0 bps | ±0.0 % |
| 38400 bps | $f_{CLK}/2^2$ | 103 | 38461.5 bps | +0.16 % |
| 76800 bps | $f_{CLK}/2$ | 103 | 76923.1 bps | +0.16 % |
| 153600 bps | $f_{CLK}$ | 103 | 153846 bps | +0.16 % |
| 312500 bps | $f_{CLK}$ | 50 | 313725.5 bps | +0.39 % |

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

**(3) Permissible baud rate range for reception**

The permissible baud rate range for reception during UART (UART0, UART1) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

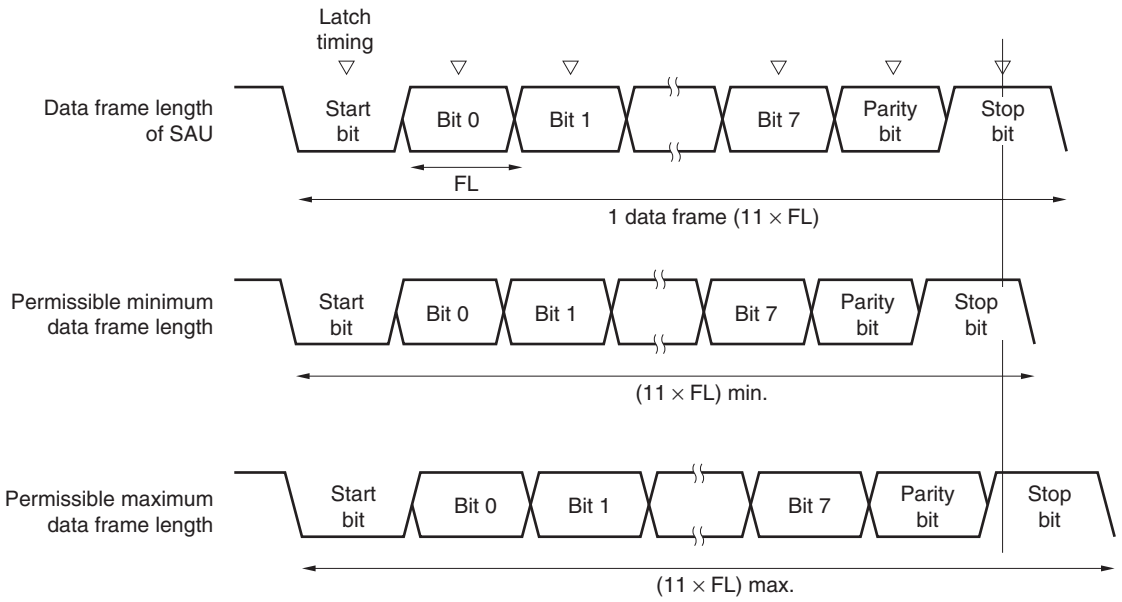
Brate: Calculated baud rate value at the reception side (See **15.7.3 (1) Baud rate calculation expression**.)

k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 01, 11

**Figure 15-135. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)**




As shown in Figure 15-135, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of serial data register mn (SDRmn) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

### 15.7.4 Procedure for processing errors that occurred during UART (UART0, UART1) communication

The procedure for processing errors that occurred during UART (UART0, UART1) communication is described in Figures 15-136 and 15-137.

**Figure 15-136. Processing Procedure in Case of Parity Error or Overrun Error**

| Software Manipulation | Hardware Status | Remark |
|---|---|---|
| Reads serial data register mn (SDRmn). → | The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data. | This is to prevent an overrun error if the next reception is completed during error processing. |
| Reads serial status register mn (SSRmn). | | Error type is identified and the read value is used to clear error flag. |
| Writes 1 to serial flag clear trigger register mn (SIRmn). → | Error flag is cleared. | Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification. |

**Figure 15-137. Processing Procedure in Case of Framing Error**

| Software Manipulation | Hardware Status | Remark |
|---|---|---|
| Reads serial data register mn (SDRmn). → | The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data. | This is to prevent an overrun error if the next reception is completed during error processing. |
| Reads serial status register mn (SSRmn). | | Error type is identified and the read value is used to clear error flag. |
| Writes serial flag clear trigger register mn (SIRmn). → | Error flag is cleared. | Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification. |
| Sets the STmn bit of serial channel stop register m (STm) to 1. → | The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operating. | |
| Synchronization with other party of communication | | Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted. |
| Sets the SSmn bit of serial channel start register m (SSm) to 1. → | The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate. | |

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

## 15.8 LIN Communication Operation

### 15.8.1 LIN transmission

Of UART transmission, UART0 support LIN communication.

For LIN transmission, channel 0 of unit 0 is used.

| UART | UART0 | UART1 |
|---|---|---|
| Support of LIN communication | Supported | Not supported |
| Target channel | Channel 0 of SAU0 | – |
| Pins used | TxD0 | – |
| Interrupt | INTST0 | – |
| | Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected. | |
| Error detection flag | None | |
| Transfer data length | 8 bits | |
| Transfer rate | Max. $f_{MCK}/6$ [bps] (SDR00 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note | |
| Data phase | Forward output (default: high level)<br>Reverse output (default: low level) | |
| Parity bit | No parity bit | |
| Stop bit | Appending 1 bit | |
| Data direction | MSB or LSB first | |

**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications. The transfer rate of LIN communication is usually set to 2.4, 9.6, or 9.2 kbps.

**Remark** $f_{MCK}$: Operation clock frequency of target channel

$f_{CLK}$: System clock frequency

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network.

Communication of LIN is single-master communication and up to 15 slaves can be connected to one master.

The slaves are used to control switches, actuators, and sensors, which are connected to the master via LIN.

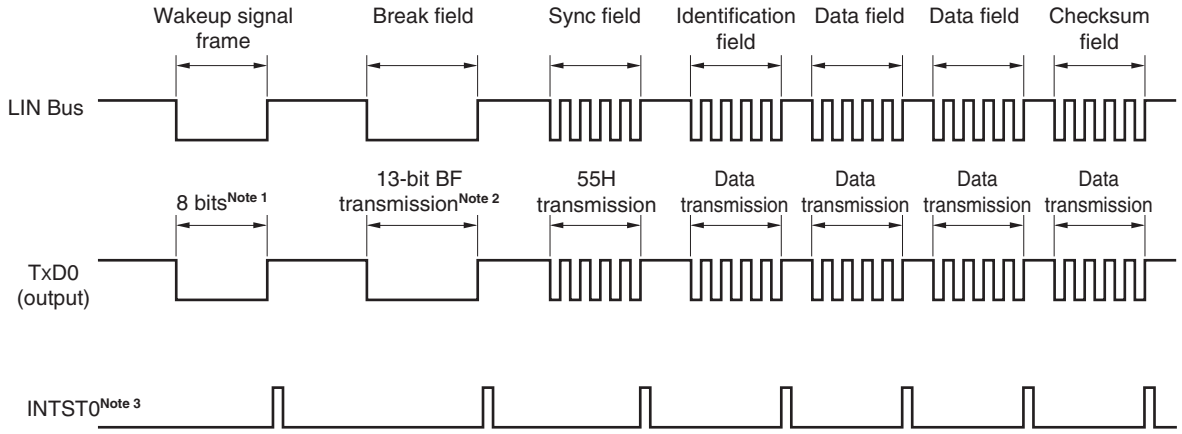
Usually, the master is connected to a network such as CAN (Controller Area Network).

A LIN bus is a single-wire bus to which nodes are connected via transceiver conforming to ISO9141.

According to the protocol of LIN, the master transmits a frame by attaching baud rate information to it. A slave receives this frame and corrects a baud rate error from the master. If the baud rate error of a slave is within ±15%, communication can be established.

Figure 15-138 outlines a transmission operation of LIN.

**Figure 15-138. Transmission Operation of LIN**




**Notes 1.** Data of 80H is transmitted.

**2.** A break field is defined to have a width of 13 bits and output a low level. Where the baud rate for main transfer is N [bps], therefore, the baud rate of the break field is calculated as follows.

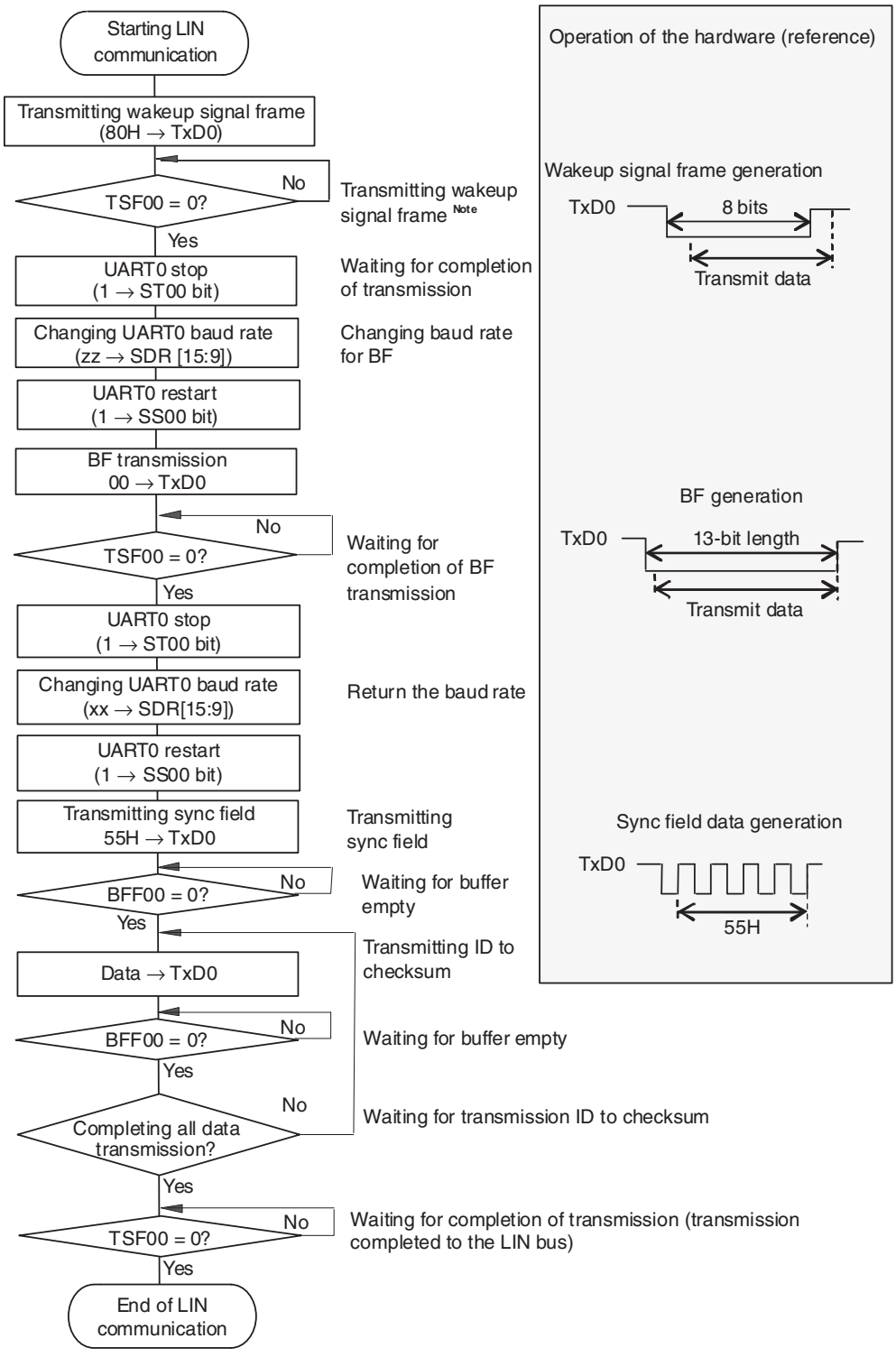
(Baud rate of break field) = 9/13 × N

By transmitting data of 00H at this baud rate, a break field is generated.

**3.** INTST0 is output upon completion of transmission.

**Remark** The interval between fields is controlled by software.

**Figure 15-139. Flowchart for LIN Transmission**




**Note** When LIN-bus start from sleep status only

**Remark** Default setting of the UART is complete, and the flow from the transmission enable status.

### 15.8.2 LIN reception

Of UART reception, UART0 support LIN communication.

For LIN reception, channel 1 of unit 0 is used.

| UART | UART0 | UART1 |
|---|---|---|
| Support of LIN communication | Supported | Not supported |
| Target channel | Channel 1 of SAU0 | – |
| Pins used | RxD0 | – |
| Interrupt | INTSR0 | – |
| | Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.) | |
| Error detection flag | • Framing error detection flag (FEF01)<br>• Overrun error detection flag (OVF01) | |
| Transfer data length | 8 bits | |
| Transfer rate | Max. $f_{MCK}$/6 [bps] (SDR01 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note | |
| Data phase | Forward output (default: high level)<br>Reverse output (default: low level) | |
| Parity bit | No parity bit (The parity bit is not checked.) | |
| Stop bit | The first bit is checked. | |
| Data direction | MSB or LSB first | |

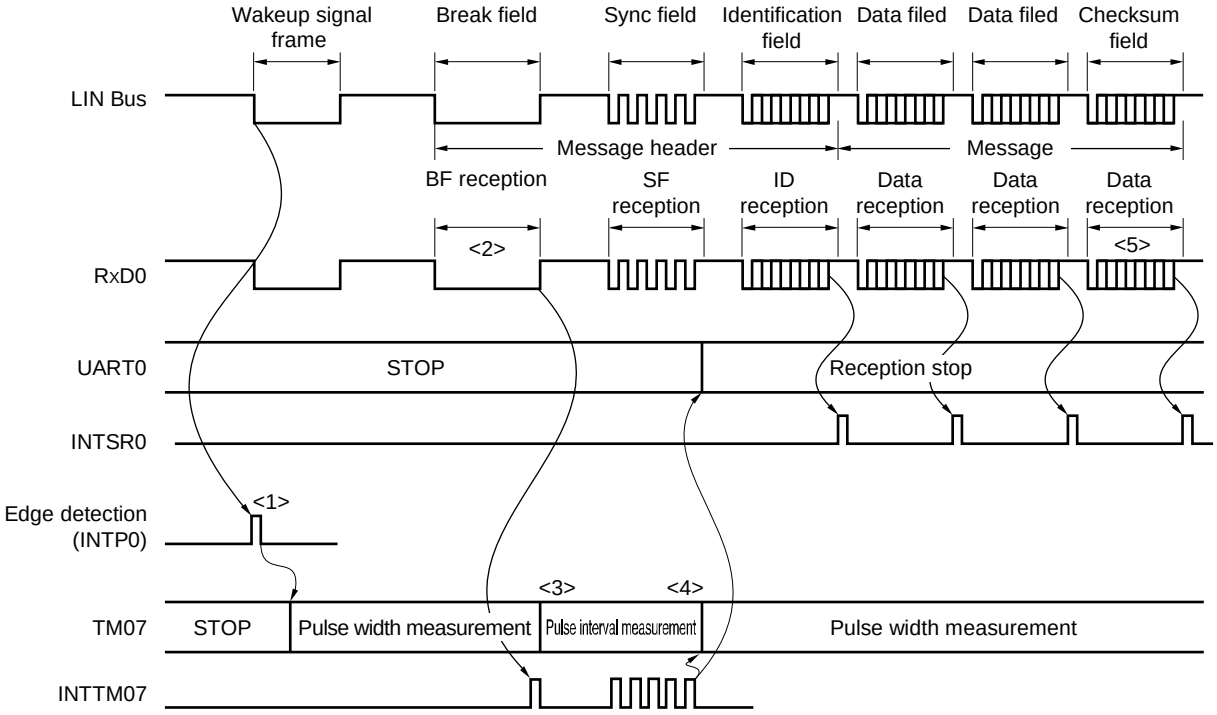
**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications.

**Remark** $f_{MCK}$: Operation clock frequency of target channel

$f_{CLK}$: System clock frequency

Figure 15-140 outlines a reception operation of LIN.

**Figure 15-140. Reception Operation of LIN**




Here is the flow of signal processing.

<1> The wakeup signal is detected by detecting an interrupt edge (INTP0) on a pin. When the wakeup signal is detected, change TM07 to pulse width measurement upon detection of the wakeup signal to measure the low-level width of the BF signal. Then wait for BF signal reception.

<2> TM07 starts measuring the low-level width upon detection of the falling edge of the BF signal, and then captures the data upon detection of the rising edge of the BF signal. The captured data is used to judge whether it is the BF signal.

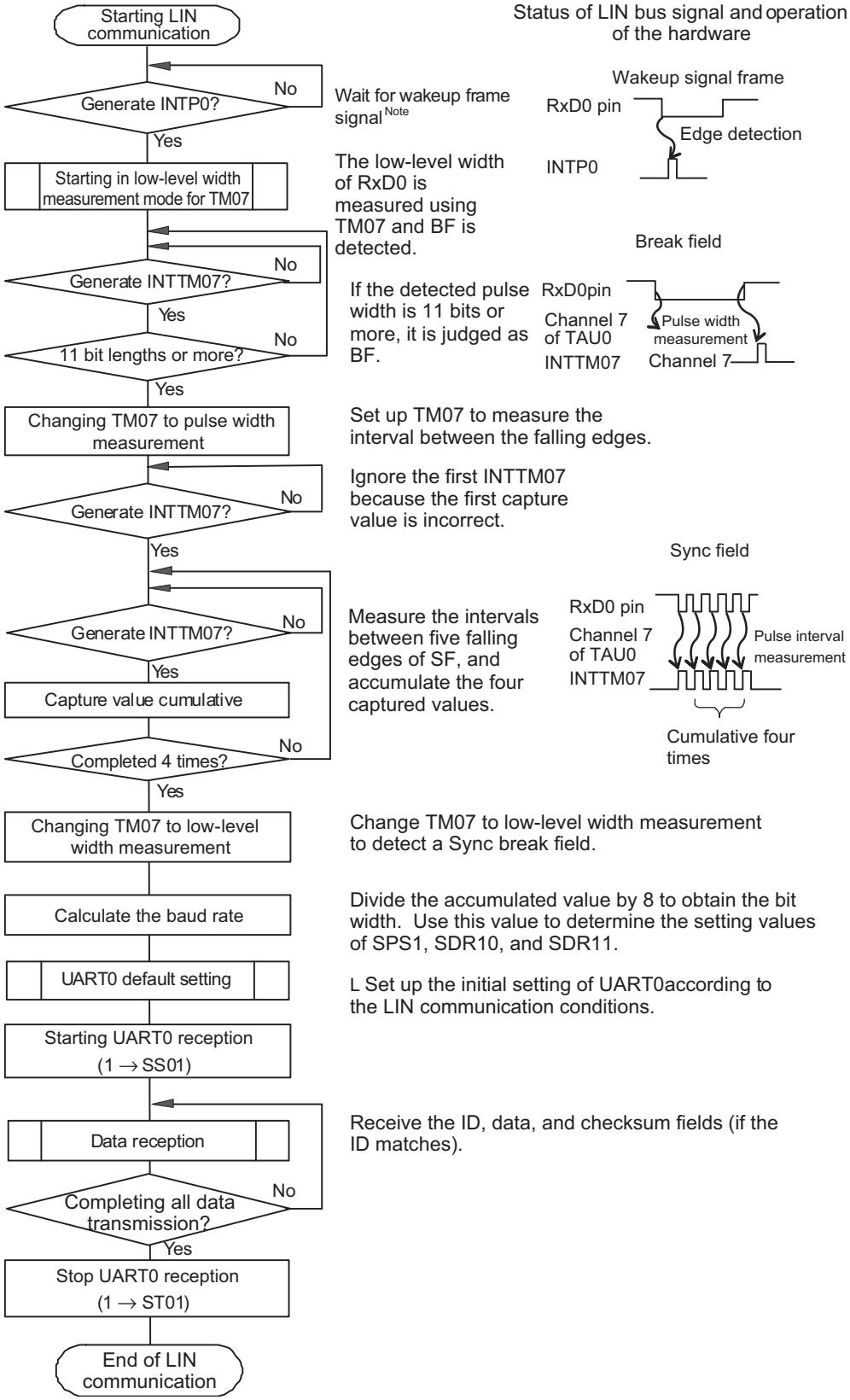
<3> When the BF signal has been received normally, change TM07 to pulse interval measurement and measure the interval between the falling edges of the RxD0 signal in the Sync field four times.

<4> When BF reception has been correctly completed, start channel 7 of the timer array unit and measure the bit interval (pulse width) of the sync field (see **6.7.4 Operation as input pulse interval measurement**).

<5> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART0 once and adjust (re-set) the baud rate.

<6> The checksum field should be distinguished by software. In addition, processing to initialize UART0 after the checksum field is received and to wait for reception of BF should also be performed by software.

**Figure 15-141. Flowchart for LIN Reception**




**Note** Required in the sleep status only.

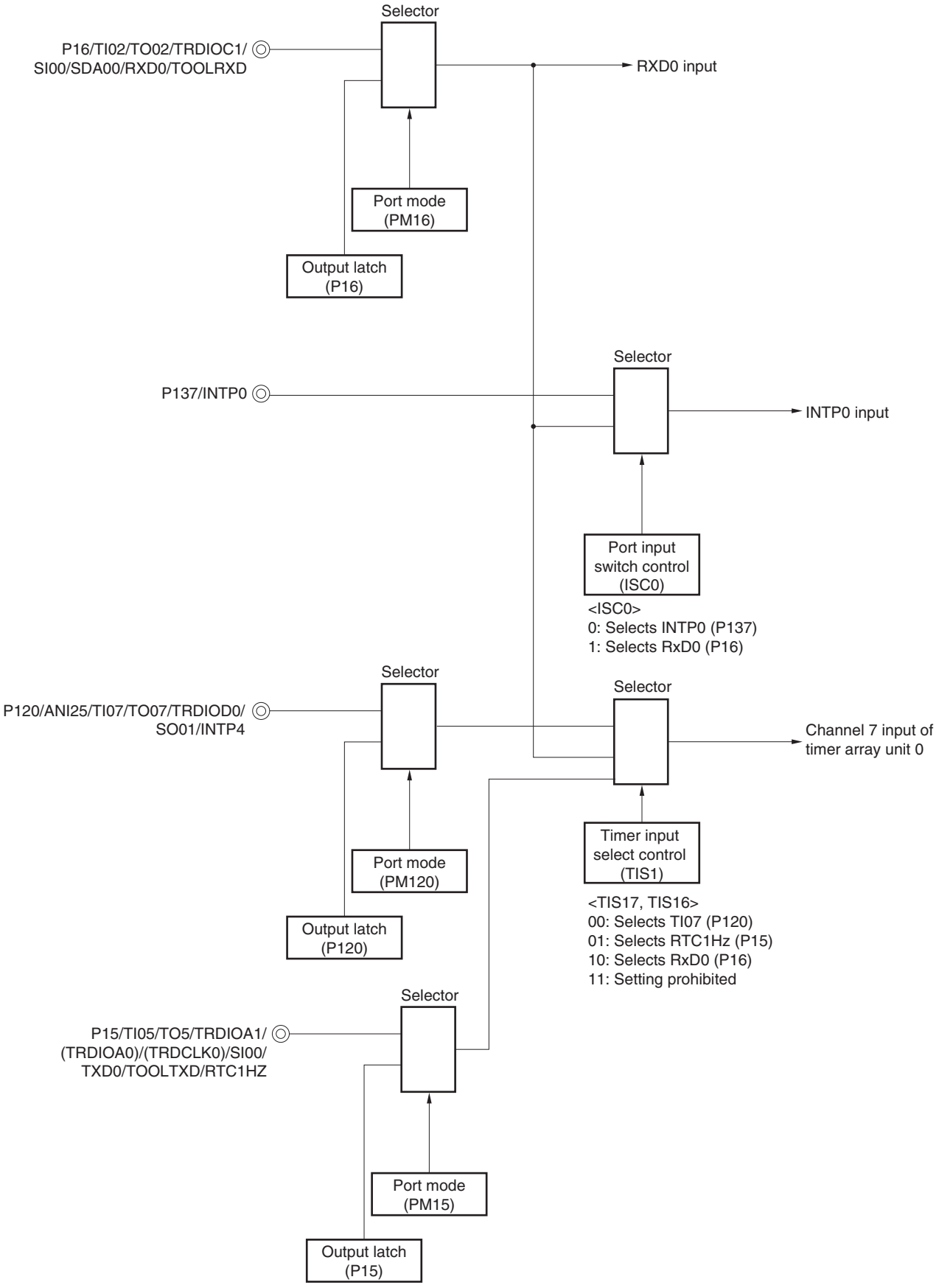
**Caution For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more $f_{MCK}$ clocks have elapsed.**

Figure 15-142 shows the configuration of a port that manipulates reception of LIN.

The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (INTP0). The length of the sync field transmitted from the master can be measured by using the external event capture operation of the timer array unit 0 to calculate a baud-rate error.

By controlling switch of port input (ISC0/TIS1), the input source of port input (RxD0) for reception can be input to the external interrupt pin (INTP0) and timer array unit 0.

**Figure 15-142. Port Configuration for Manipulating Reception of LIN**




**Remark** ISC0: Bit 0 of the input switch control register (ISC) (See **Figure 15-18**.)

The peripheral functions used for the LIN communication operation are as follows.

<Peripheral functions used>

- External interrupt (INTP0); Wakeup signal detection
  Usage: To detect an edge of the wakeup signal and the start of communication
- Channel 7 of timer array unit; Baud rate error detection and break field (BF) detection
  Usage: To detect the length of the sync field (SF) and divide it by the number of bits in order to detect an error (The interval of the edge input to RxD0 is measured in the capture mode.)
  To measure a low-level width and determine whether the field is a break field (BF)
- Channels 0 and 1 (UART0) of serial array unit 0 (SAU0)

## 15.9 Operation of Simplified I²C (IIC00, IIC01, IIC10, IIC11) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function**Note** and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Overrun error
- Parity error (ACK error)

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection function

**Note** When receiving the last data, ACK will not be output if 0 is written to the SOEmn (SOEm register) bit and serial communication data output is stopped. See **15.9.3 (2) Processing flow** for details.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

The channel supporting simplified $I^2C$ (IIC00, IIC01, IIC10, IIC11) is channels 0 and 1 of SAU0 and channels 0 and 1 of SAU1.

- Group A products

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note2 | | IIC01 |

- Products of Groups C-1 and D-1

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note2 | | IIC01 |
| 1 | 0 | CSI10 (supporting SPI function) Note1, 2 | UART1 | IIC10 |
| | 1 | - | | - |

- Products of Groups B, C-2, D-2, and E

| Unit | Channel | Used as CSI | Used as UART | Used as Simplified $I^2C$ |
|---|---|---|---|---|
| 0 | 0 | CSI00 (supporting SPI function) Note2 | UART0 (supporting LIN-bus) | IIC00 |
| | 1 | CSI01 (supporting SPI function) Note2 | | IIC01 |
| 1 | 0 | CSI10 (supporting SPI function) Note1, 2 | UART1 | IIC10 |
| | 1 | CSI11 (supporting SPI function) Note2 | | IIC11 |

**Notes 1.** 48-pin, 32-pin and 30-pin products do not have $\overline{SSI10}$ pin.

<R> **2.** Set CKPmn bit of SCRmn register to 1, when SSEmn = 1 (Enables $\overline{SSImn}$ pin input).
(m = 0, 1, n = 0, 1)

**Remark** Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory
Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory
Group C-1: RL78/F13 (CAN and LIN incorporated) products with 30 or 32 pins
Group C-2: RL78/F13 (CAN and LIN incorporated) products with 48, 64, or 80 pins
Group D-1: RL78/F14 products with 30 or 32 pins
Group D-2: RL78/F14 products with 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory
Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

Simplified $I^2C$ (IIC00, IIC01, IIC10, IIC11) performs the following four types of communication operations.

- Address field transmission (See **15.9.1 Address field transmission.**)
- Data transmission (See **15.9.2 Data transmission**.)
- Data reception (See **15.9.3 Data reception**.)
- Stop condition generation (See **15.9.4 Stop condition generation**.)

### 15.9.1 Address field transmission

Address field transmission is a transmission operation that first executes in $I^2C$ communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

| Simplified $I^2C$ | IIC00 | IIC01 | IIC10 | IIC11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | SCL00, SDA00 Note | SCL01, SDA01 Note | SCL10, SDA10 Note | SCL11, SDA11 Note |
| Interrupt | INTIIC00 | INTIIC01 | INTIIC10 | INTIIC11 |
| | Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.) | | | |
| Error detection flag | Parity error detection flag (PEFmn) | | | |
| Transfer data length | 8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control) | | | |
| Transfer rate | Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) $f_{MCK}$: Operation clock frequency of target channel<br>However, the following condition must be satisfied in each mode of $I^2C$.<br>• Max. 400 kHz (first mode)<br>• Max. 100 kHz (standard mode) | | | |
| Data level | Forward output (default: high level) | | | |
| Parity bit | No parity bit | | | |
| Stop bit | Appending 1 bit (for ACK reception timing) | | | |
| Data direction | MSB first | | | |

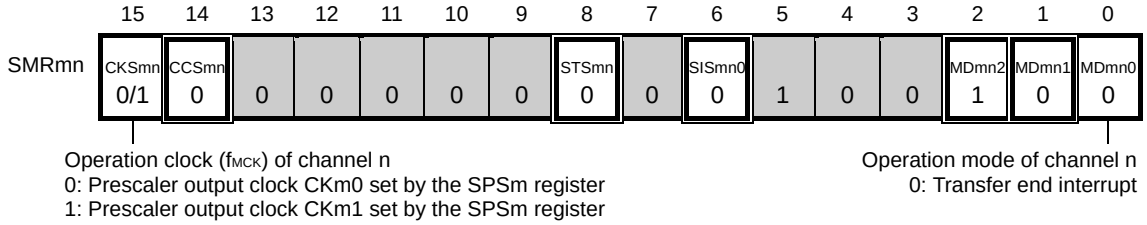
**Note** To perform communication via simplified $I^2C$, set the N-ch open-drain output ($EV_{DD0}$ tolerance) mode (POMxx = 1) for the port output mode registers (POMxx) (see **4.3 Registers Controlling Port Function** for details). When IIC00, IIC01, IIC10, IIC11 communicating with an external device with a different potential, set the N-ch open-drain output ($EV_{DD0}$ tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL01, SCL10, SCL11) (see **4.4.4 Connecting to external device with different potential (3 V)** for details).

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

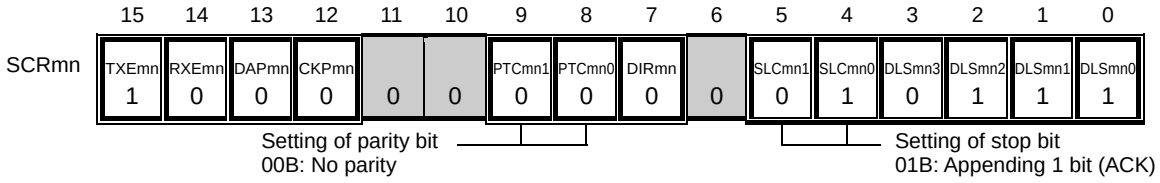
**(1) Register setting**

**Figure 15-143. Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00, IIC01, IIC10, IIC11) (1/2)**

**(a) Serial mode register mn (SMRmn)**

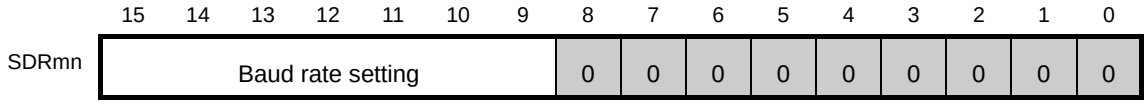



**(b) Serial communication operation setting register mn (SCRmn)**

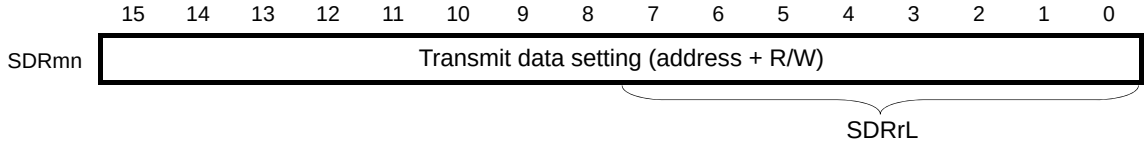



**(c) Serial data register mn (SDRmn)**

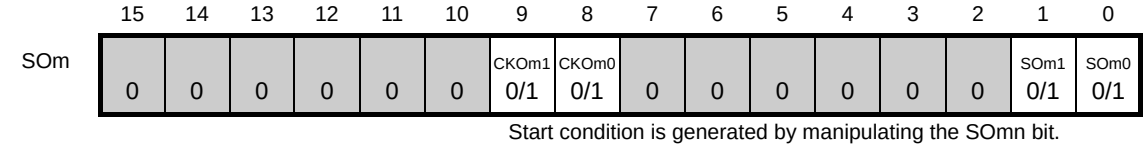
**(1) When operation is stopped (SEmn = 0)**



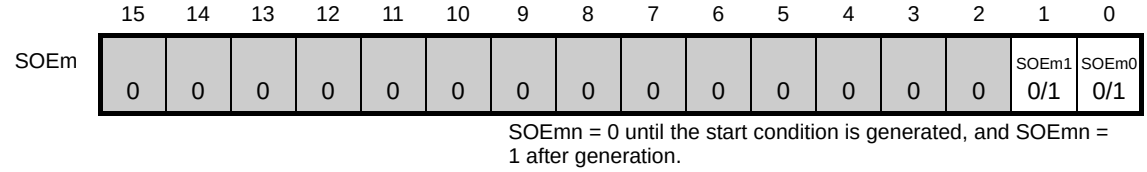

**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRrL)**




**(d) Serial output register m (SOm) ⋯ Sets only the bits of the target channel.**




**(e) Serial output enable register m (SOEm) ⋯ Sets only the bits of the target channel.**




**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-143. Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00, IIC01, IIC10, IIC11) (2/2)**

**(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1<br>0/1 | SSm0<br>0/1 |

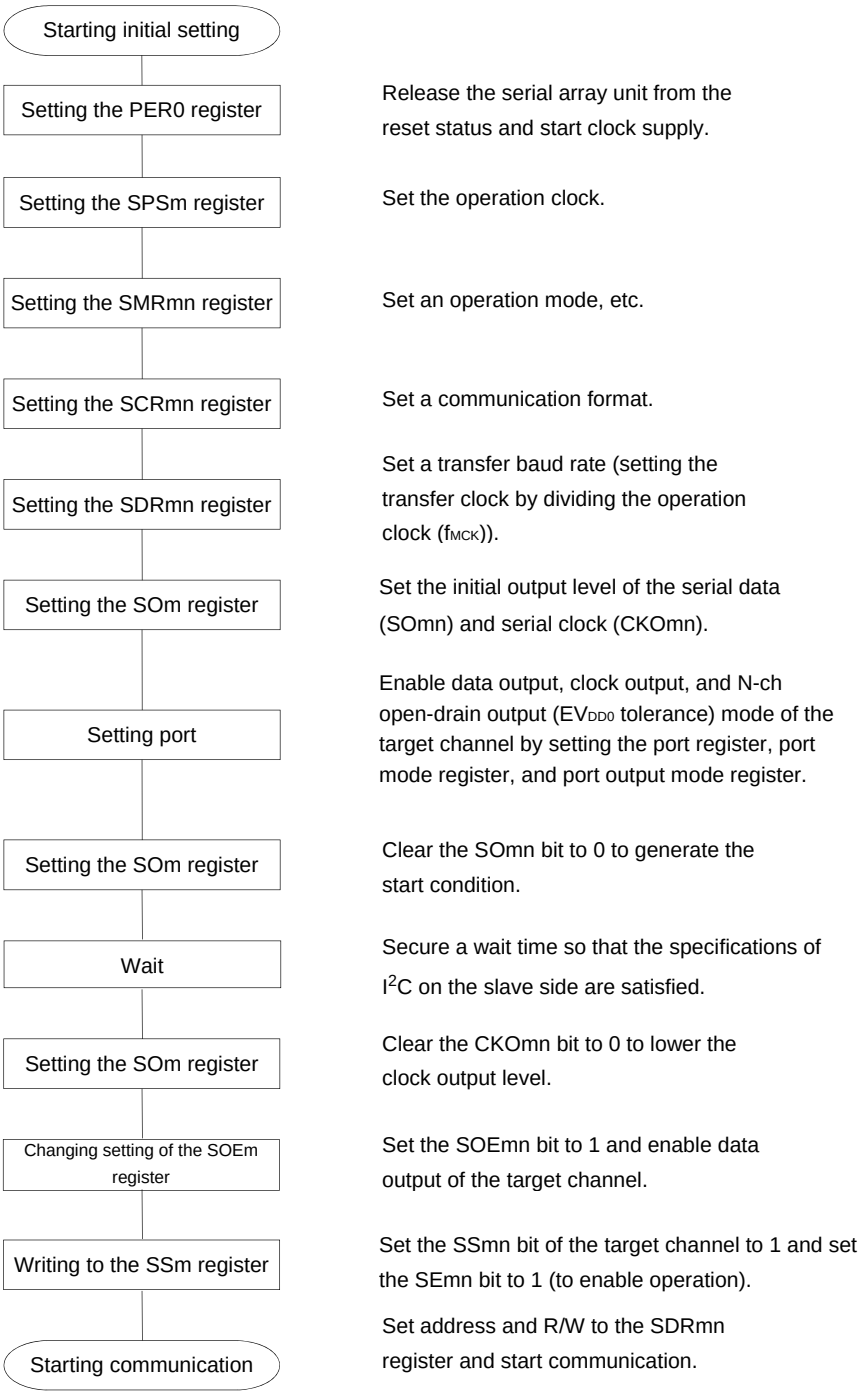
**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

### (2) Operation procedure

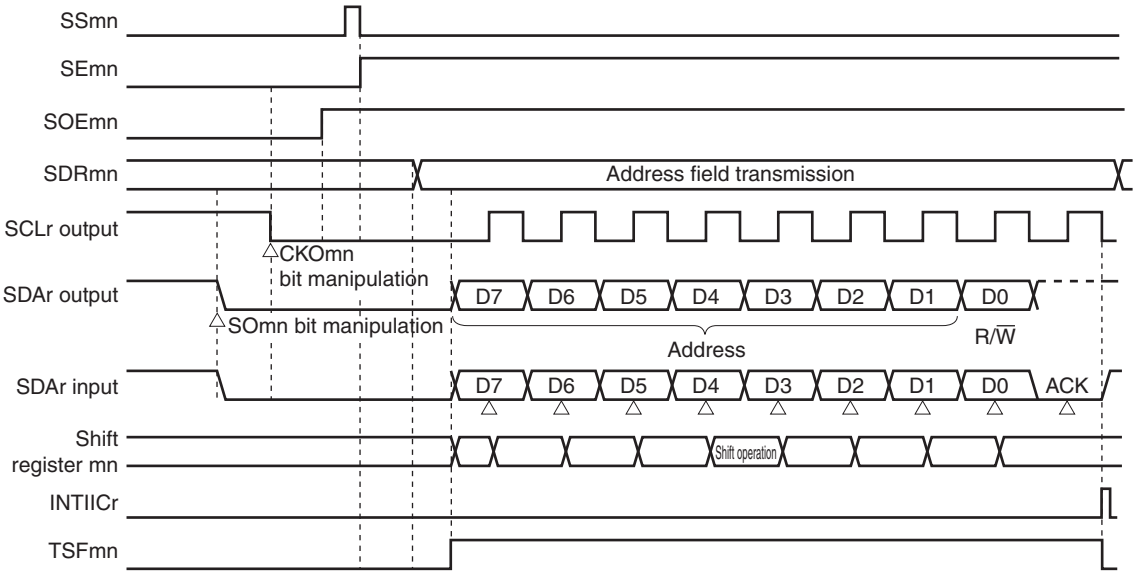
**Figure 15-144. Initial Setting Procedure for Address Field Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

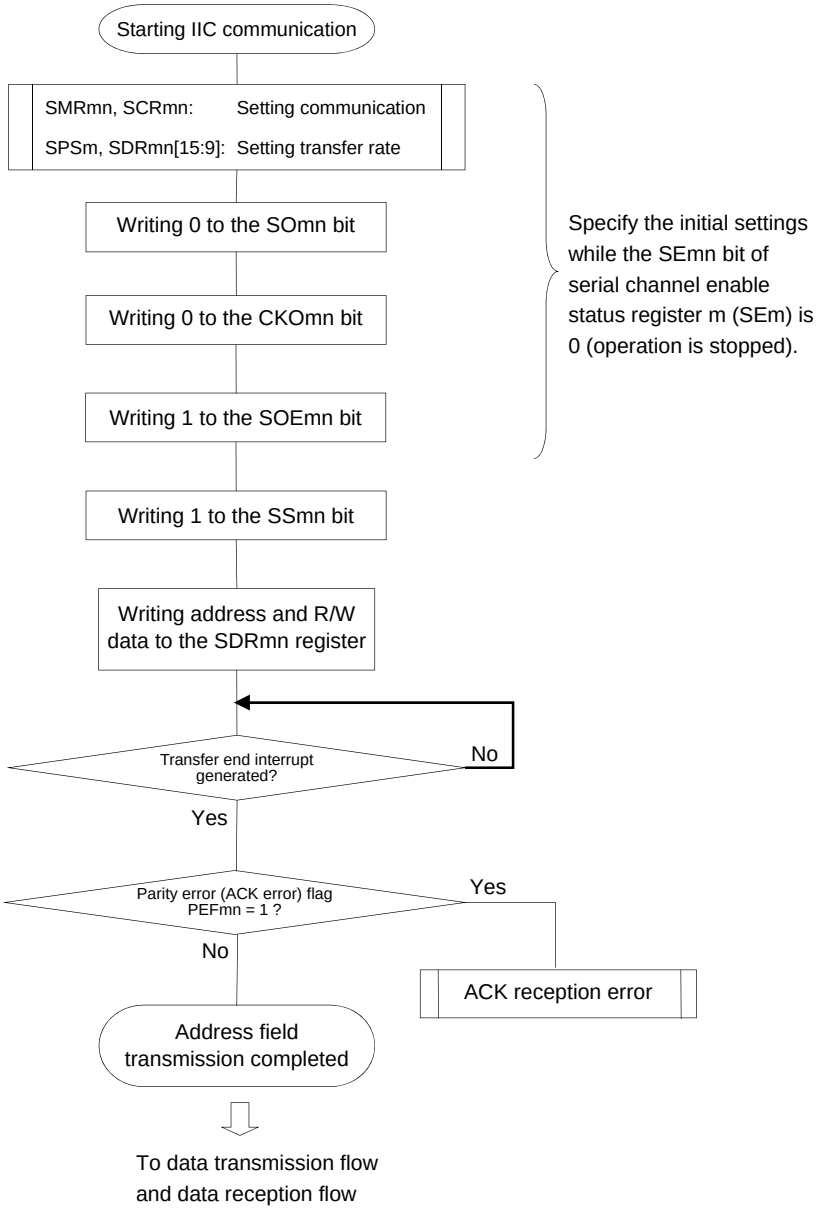
### (3) Processing flow

**Figure 15-145. Timing Chart of Address Field Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-146. Flowchart of Address Field Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### 15.9.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

| Simplified $I^2C$ | IIC00 | IIC01 | IIC10 | IIC11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | SCL00, SDA00 Note | SCL01, SDA01 Note | SCL10, SDA10 Note | SCL11, SDA11 Note |
| Interrupt | INTIIC00 | INTIIC01 | INTIIC10 | INTIIC11 |
| | Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.) | | | |
| Error detection flag | Parity error detection flag (PEFmn) | | | |
| Transfer data length | 8 bits | | | |
| Transfer rate | Max. $f_{MCK}$/4 [Hz] (SDRmn[15:9] = 1 or more) $f_{MCK}$: Operation clock frequency of target channel<br>However, the following condition must be satisfied in each mode of $I^2C$.<br>• Max. 400 kHz (first mode)<br>• Max. 100 kHz (standard mode) | | | |
| Data level | Forward output (default: high level) | | | |
| Parity bit | No parity bit | | | |
| Stop bit | Appending 1 bit (for ACK reception timing) | | | |
| Data direction | MSB first | | | |

**Note** To perform communication via simplified $I^2C$, set the N-ch open-drain output ($EV_{DD0}$ tolerance) mode (POMxx = 1) for the port output mode registers (POMxx) (see **4.3 Registers Controlling Port Function** for details). When IIC00, IIC01, IIC10, IIC11 communicating with an external device with a different potential, set the N-ch open-drain output ($EV_{DD0}$ tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL01, SCL10, SCL11) (see **4.4.4 Connecting to external device with different potential (3 V)** for details).

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

**(1) Register setting**

**Figure 15-147. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00, IIC01, IIC10, IIC11) (1/2)**

**(a) Serial mode register mn (SMRmn) … Do not manipulate this register during data transmission/reception.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SMRmn | CKSmn | CCSmn | 0 | 0 | 0 | 0 | 0 | STSmn | 0 | SISmn0 | 1 | 0 | 0 | MDmn2 | MDmn1 | MDmn0 |
| | 0/1 | 0 | | | | | | 0 | | 0 | | | | 1 | 0 | 0 |

**(b) Serial communication operation setting register mn (SCRmn) … Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SCRmn | TXEmn | RXEmn | DAPmn | CKPmn | 0 | 0 | PTCmn1 | PTCmn0 | DIRmn | 0 | SLCmn1 | SLCmn0 | DLSmn3 | DLSmn2 | DLSmn1 | DLSmn0 |
| | 1 | 0 | 0 | 0 | | | 0 | 0 | 0 | | 0 | 1 | 0 | 1 | 1 | 1 |

**(c) Serial data register mn (SDRmn)**

**(1) When operation is stopped (SEmn = 0)**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SDRmn | Baud rate setting Note 1 | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRrL)**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SDRmn | Transmit data setting | | | | | | | | | | | | | | | |

SDRrL

**(d) Serial output register m (SOm) … Do not manipulate this register during data transmission/reception.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOm | 0 | 0 | 0 | 0 | 0 | 0 | CKOm1 | CKOm0 | 0 | 0 | 0 | 0 | 0 | 0 | SOm1 | SOm0 |
| | | | | | | | 0/1 Note 2 | 0/1 Note 2 | | | | | | | 0/1 Note 2 | 0/1 Note 2 |

**(e) Serial output enable register m (SOEm) … Do not manipulate this register during data transmission/reception.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 | SOEm0 |
| | | | | | | | | | | | | | | | 0/1 | 0/1 |

**Notes 1.** Setting these bits is unnecessary because they are set for transmission of an address field.

**2.** The value varies depending on the communication data during communication operation.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-147. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00, IIC01, IIC10, IIC11) (2/2)**

**(f) Serial channel start register m (SSm) … Do not manipulate this register during data transmission/reception.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1<br>0/1 | SSm0<br>0/1 |

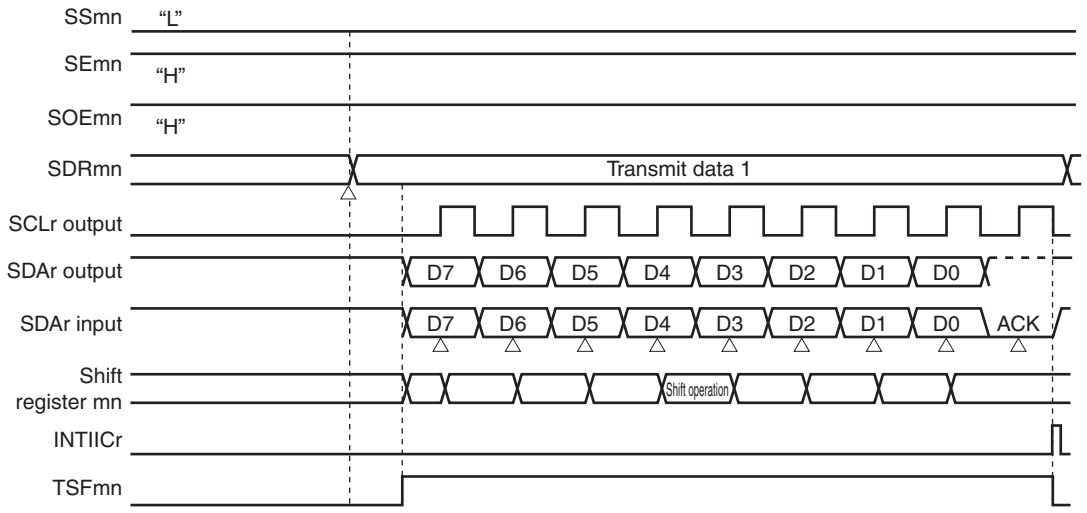
**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐ : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

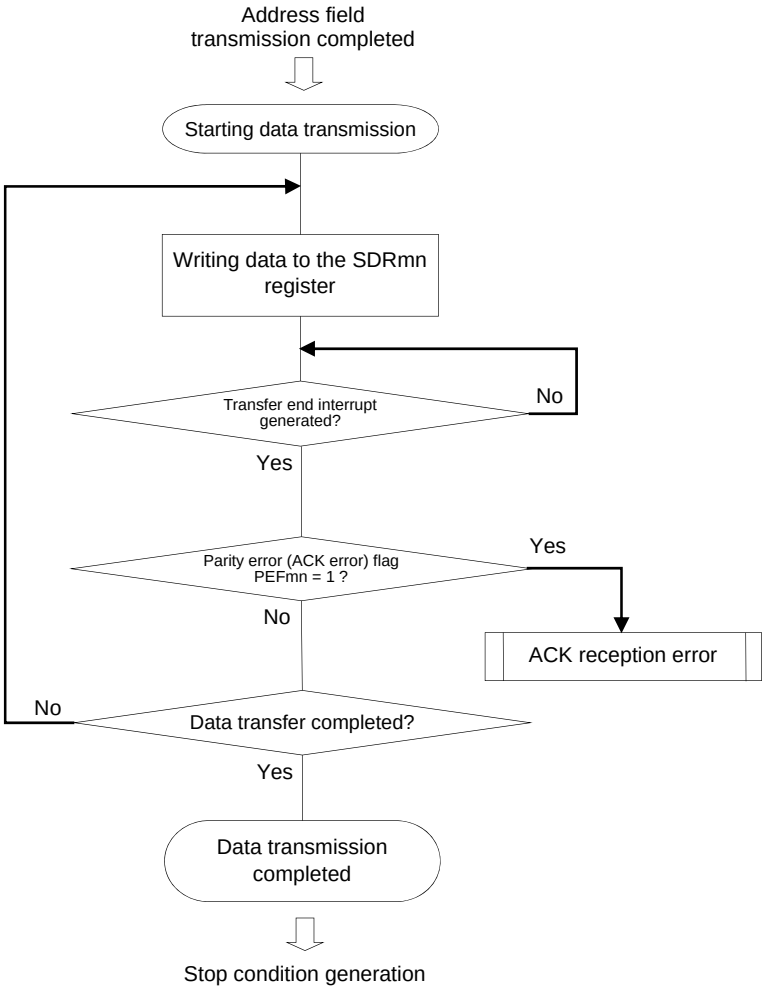
### (2) Processing flow

**Figure 15-148. Timing Chart of Data Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-149. Flowchart of Data Transmission**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11), mn = 00, 01, 10, 11

### 15.9.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

| Simplified $I^2C$ | IIC00 | IIC01 | IIC10 | IIC11 |
|---|---|---|---|---|
| Target channel | Channel 0 of SAU0 | Channel 1 of SAU0 | Channel 0 of SAU1 | Channel 1 of SAU1 |
| Pins used | SCL00, SDA00 Note | SCL01, SDA01 Note | SCL10, SDA10 Note | SCL11, SDA11 Note |
| Interrupt | INTIIC00 | INTIIC01 | INTIIC10 | INTIIC11 |
| | Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.) | | | |
| Error detection flag | Overrun error detection flag (OVFmn) only | | | |
| Transfer data length | 8 bits | | | |
| Transfer rate | Max. $f_{MCK}$/4 [Hz] (SDRmn[15:9] = 1 or more) $f_{MCK}$: Operation clock frequency of target channel<br>However, the following condition must be satisfied in each mode of $I^2C$.<br>• Max. 400 kHz (first mode)<br>• Max. 100 kHz (standard mode) | | | |
| Data level | Forward output (default: high level) | | | |
| Parity bit | No parity bit | | | |
| Stop bit | Appending 1 bit (ACK transmission) | | | |
| Data direction | MSB first | | | |

**Note** To perform communication via simplified $I^2C$, set the N-ch open-drain output ($EV_{DD0}$ tolerance) mode (POMxx = 1) for the port output mode registers (POMxx) (see **4.3 Registers Controlling Port Function** for details). When IIC00, IIC01, IIC10, IIC11 communicating with an external device with a different potential, set the N-ch open-drain output ($EV_{DD0}$ tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL01, SCL10, SCL11) (see **4.4.4 Connecting to external device with different potential (3 V)** for details).

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

**(1) Register setting**

**Figure 15-150. Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC01, IIC10, IIC11) (1/2)**

**(a) Serial mode register mn (SMRmn) … Do not manipulate this register during data transmission/reception.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SMRmn | CKSmn | CCSmn | 0 | 0 | 0 | 0 | 0 | STSmn | 0 | SISmn0 | 1 | 0 | 0 | MDmn2 | MDmn1 | MDmn0 |
| | 0/1 | 0 | | | | | | 0 | | 0 | | | | 1 | 0 | 0 |

**(b) Serial communication operation setting register mn (SCRmn) … Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SCRmn | TXEmn | RXEmn | DAPmn | CKPmn | 0 | 0 | PTCmn1 | PTCmn0 | DIRmn | 0 | SLCmn1 | SLCmn0 | DLSmn3 | DLSmn2 | DLSmn1 | DLSmn0 |
| | 0 | 1 | 0 | 0 | | | 0 | 0 | 0 | | 0 | 1 | 0 | 1 | 1 | 1 |

**(c) Serial data register mn (SDRmn)**

**(1) When operation is stopped (SEmn = 0)**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SDRmn | Baud rate setting Note 1 | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**(2) When operation is in progress (SEmn = 1) (Lower 8 bits: SDRrL)**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SDRmn | Dummy transmit data setting (FFH) | | | | | | | | | | | | | | | |

**(d) Serial output register m (SOm) … Do not manipulate this register during data transmission/reception.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOm | 0 | 0 | 0 | 0 | 0 | 0 | CKOm1 | CKOm0 | 0 | 0 | 0 | 0 | 0 | 0 | SOm1 | SOm0 |
| | | | | | | | 0/1 Note 2 | 0/1 Note 2 | | | | | | | 0/1 Note 2 | 0/1 Note 2 |

**(e) Serial output enable register m (SOEm) … Do not manipulate this register during data transmission/reception.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOEm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SOEm1 | SOEm0 |
| | | | | | | | | | | | | | | | 0/1 | 0/1 |

**Notes 1.** Setting these bits is unnecessary because they are set for transmission of an address field.

**2.** The value varies depending on the communication data during communication operation.

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11), mn = 00, 01, 10, 11

**2.** ☐: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

**Figure 15-150. Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC01, IIC10, IIC11) (2/2)**

**(f) Serial channel start register m (SSm) … Do not manipulate this register during data transmission/reception.**

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSm | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSm1<br>0/1 | SSm0<br>0/1 |

**Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11), mn = 00, 01, 10, 11

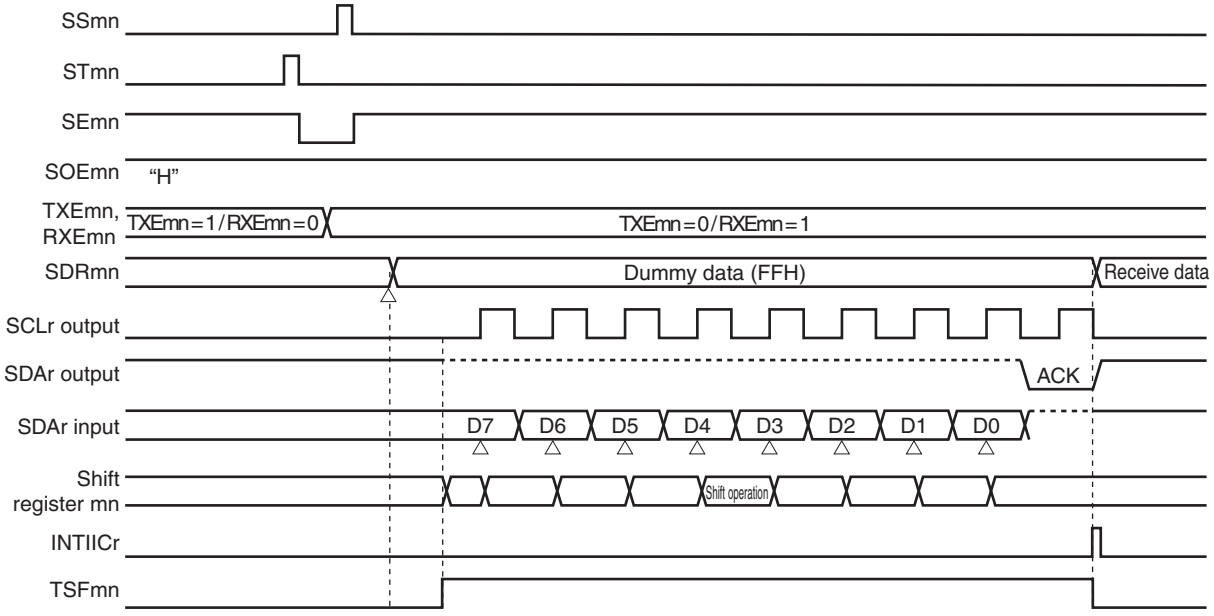
**2.** ☐: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

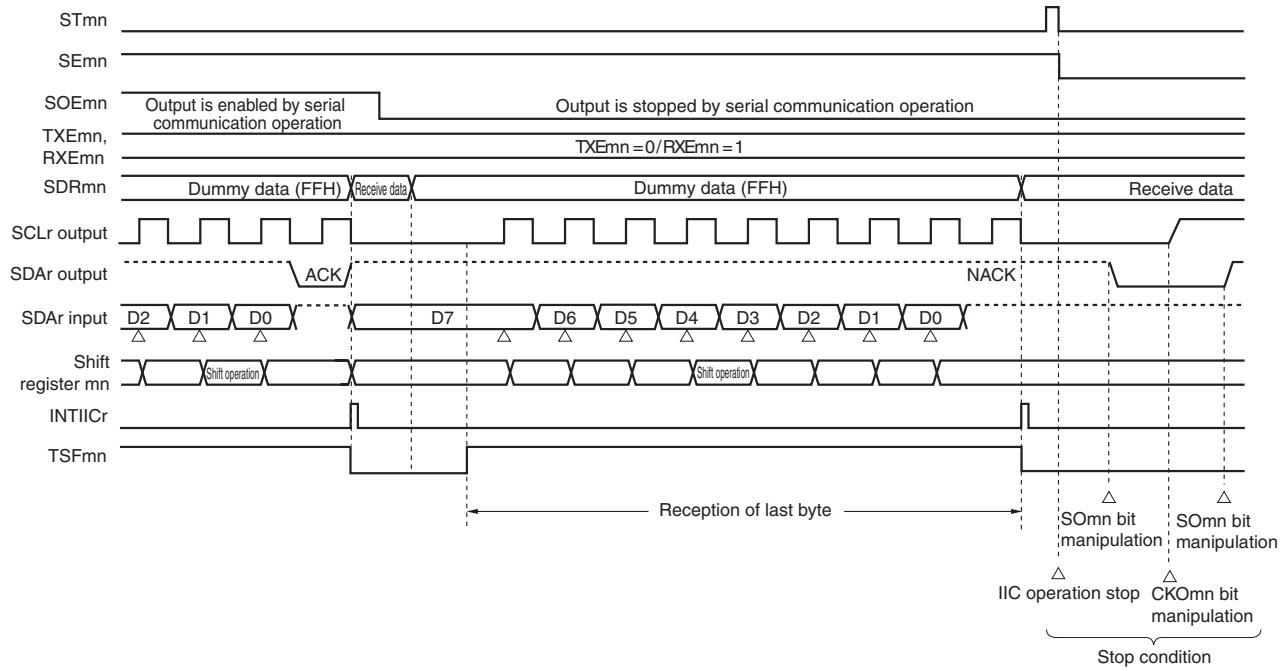
### (2) Processing flow

**Figure 15-151. Timing Chart of Data Reception**

#### (a) When starting data reception

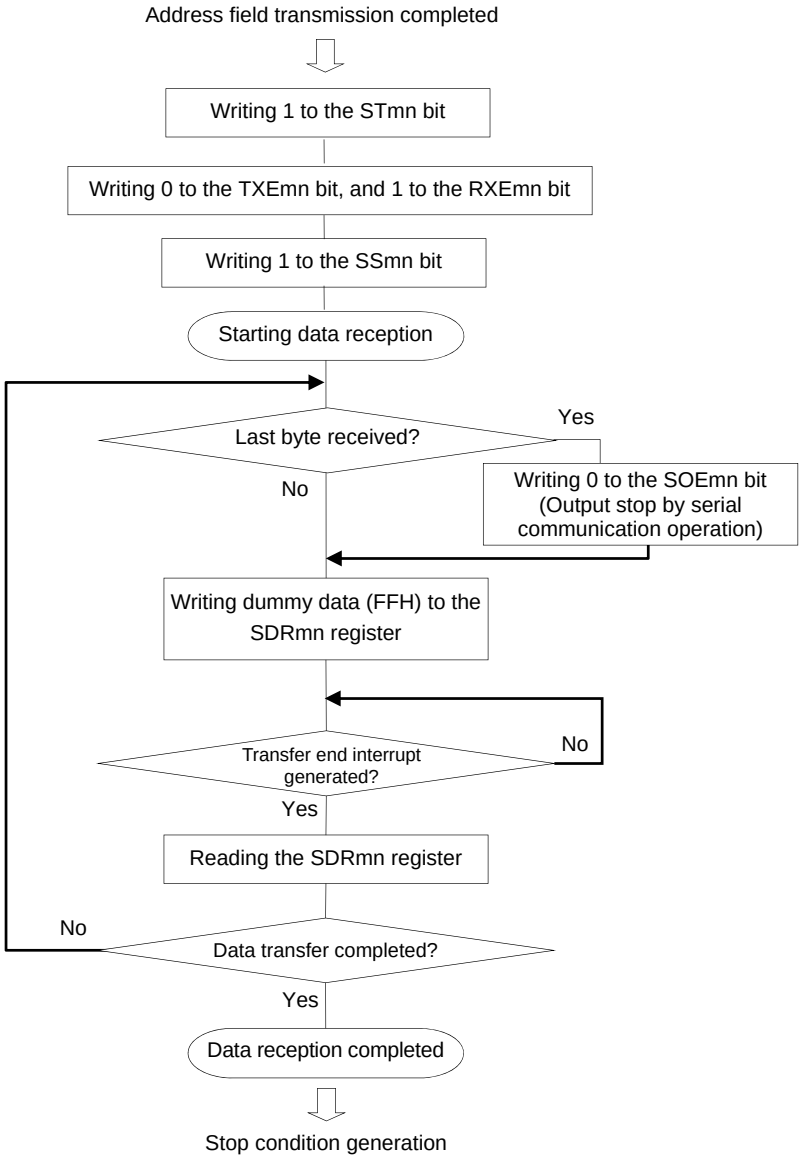



#### (b) When receiving last data




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11)
mn = 00, 01, 10, 11

**Figure 15-152. Flowchart of Data Reception**




**Caution** **ACK is not output when the last data is received (NACK). Communication is then completed by setting 1 to the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.**

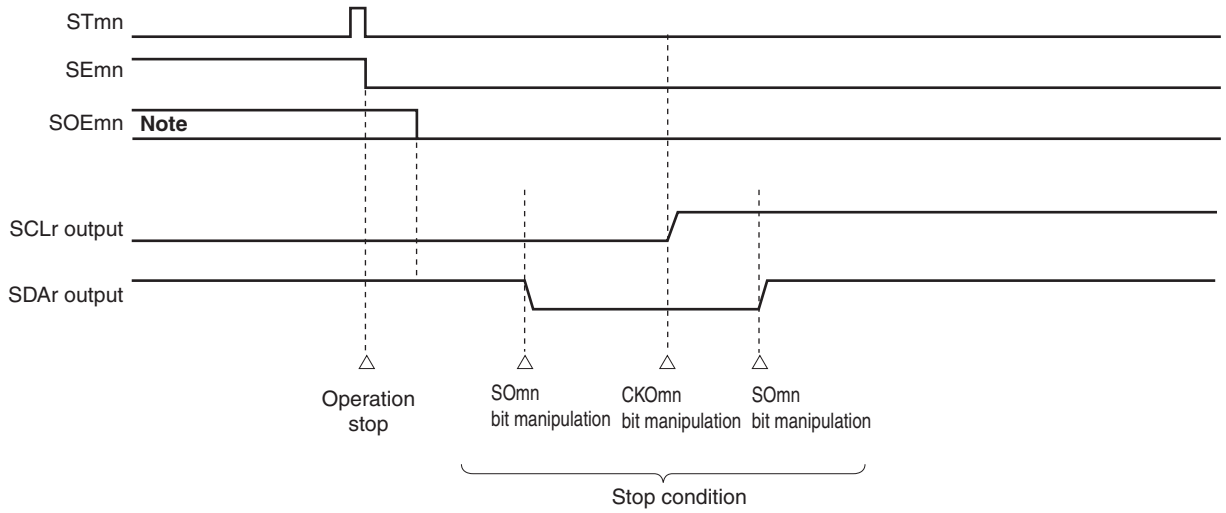
**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### 15.9.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

#### (1) Processing flow

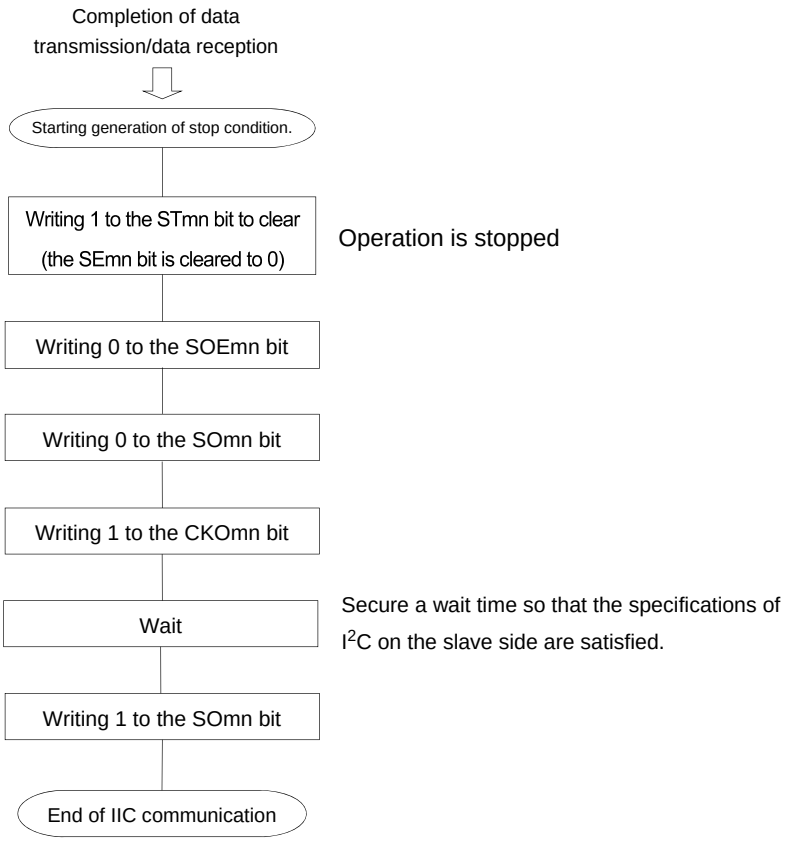
**Figure 15-153. Timing Chart of Stop Condition Generation**




**Note** During a receive operation, the SOEmn bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11), mn = 00, 01, 10, 11

**Figure 15-154. Flowchart of Stop Condition Generation**




**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### 15.9.5 Calculating transfer rate

The transfer rate for simplified $I^2C$ (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock ($f_{MCK}$) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

**Caution Setting SDRmn[15:9] = 0000000B is prohibited. Setting SDRmn[15:9] = 0000001B or more.**

**Remarks 1.** The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000001B to 1111111B) and therefore is 1 to 127.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

The operation clock ($f_{MCK}$) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

**Table 15-5. Selection of Operation Clock For Simplified I²C**

| SMRmn Register | SPSm Register | | | | | | | | Operation Clock ($f_{MCK}$) Note | |
|---|---|---|---|---|---|---|---|---|---|---|
| CKSmn | PRS m13 | PRS m12 | PRS m11 | PRS m10 | PRS m03 | PRS m02 | PRS m01 | PRS m00 | | $f_{CLK}$ = 32 MHz |
| 0 | X | X | X | X | 0 | 0 | 0 | 0 | $f_{CLK}$ | 32 MHz |
| | X | X | X | X | 0 | 0 | 0 | 1 | $f_{CLK}/2$ | 16 MHz |
| | X | X | X | X | 0 | 0 | 1 | 0 | $f_{CLK}/2^2$ | 8 MHz |
| | X | X | X | X | 0 | 0 | 1 | 1 | $f_{CLK}/2^3$ | 4 MHz |
| | X | X | X | X | 0 | 1 | 0 | 0 | $f_{CLK}/2^4$ | 2 MHz |
| | X | X | X | X | 0 | 1 | 0 | 1 | $f_{CLK}/2^5$ | 1 MHz |
| | X | X | X | X | 0 | 1 | 1 | 0 | $f_{CLK}/2^6$ | 500 kHz |
| | X | X | X | X | 0 | 1 | 1 | 1 | $f_{CLK}/2^7$ | 250 kHz |
| | X | X | X | X | 1 | 0 | 0 | 0 | $f_{CLK}/2^8$ | 125 kHz |
| | X | X | X | X | 1 | 0 | 0 | 1 | $f_{CLK}/2^9$ | 62.5 kHz |
| | X | X | X | X | 1 | 0 | 1 | 0 | $f_{CLK}/2^{10}$ | 31.25 kHz |
| | X | X | X | X | 1 | 0 | 1 | 1 | $f_{CLK}/2^{11}$ | 15.63 kHz |
| 1 | 0 | 0 | 0 | 0 | X | X | X | X | $f_{CLK}$ | 32 MHz |
| | 0 | 0 | 0 | 1 | X | X | X | X | $f_{CLK}/2$ | 16 MHz |
| | 0 | 0 | 1 | 0 | X | X | X | X | $f_{CLK}/2^2$ | 8 MHz |
| | 0 | 0 | 1 | 1 | X | X | X | X | $f_{CLK}/2^3$ | 4 MHz |
| | 0 | 1 | 0 | 0 | X | X | X | X | $f_{CLK}/2^4$ | 2 MHz |
| | 0 | 1 | 0 | 1 | X | X | X | X | $f_{CLK}/2^5$ | 1 MHz |
| | 0 | 1 | 1 | 0 | X | X | X | X | $f_{CLK}/2^6$ | 500 kHz |
| | 0 | 1 | 1 | 1 | X | X | X | X | $f_{CLK}/2^7$ | 250 kHz |
| | 1 | 0 | 0 | 0 | X | X | X | X | $f_{CLK}/2^8$ | 125 kHz |
| | 1 | 0 | 0 | 1 | X | X | X | X | $f_{CLK}/2^9$ | 62.5 kHz |
| | 1 | 0 | 1 | 0 | X | X | X | X | $f_{CLK}/2^{10}$ | 31.25 kHz |
| | 1 | 0 | 1 | 1 | X | X | X | X | $f_{CLK}/2^{11}$ | 15.63 kHz |
| Other than above | | | | | | | | | | Setting prohibited |

**Note** When changing the clock selected for $f_{CLK}$ (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 0003H) the operation of the serial array unit (SAU).

**Remarks 1.** X: Don't care

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

Here is an example of setting an $I^2C$ transfer rate where $f_{MCK} = f_{CLK} = 32$ MHz.

| $I^2C$ Transfer Mode (Desired Transfer Rate) | $f_{CLK}$ = 32 MHz | | | |
|---|---|---|---|---|
| | Operation Clock ($f_{MCK}$) | SDRmn[15:9] | Calculated Transfer Rate | Error from Desired Transfer Rate |
| 100 kHz | $f_{CLK}/2$ | 79 | 100 kHz | 0.0% |
| 400 kHz | $f_{CLK}$ | 39 | 400 kHz | 0.0% |

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), mn = 00, 01, 10, 11

### 15.9.6 Procedure for processing errors that occurred during simplified I²C (IIC00, IIC01, IIC10, IIC11) communication

The procedure for processing errors that occurred during simplified I²C (IIC00, IIC01, IIC10, IIC11) communication is described in Figure 15-155 and 15-156.

**Figure 15-155. Processing Procedure in Case of Overrun Error**

| Software Manipulation | Hardware Status | Remark |
|---|---|---|
| Reads serial data register mn (SDRmn). → | The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data. | This is to prevent an overrun error if the next reception is completed during error processing. |
| Reads serial status register mn (SSRmn). | | Error type is identified and the read value is used to clear error flag. |
| Writes 1 to serial flag clear trigger register mn (SIRmn). → | Error flag is cleared. | Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification. |

**Figure 15-156. Processing Procedure in Case of Parity Error (ACK error) in Simplified I²C Mode**

| Software Manipulation | Hardware Status | Remark |
|---|---|---|
| Reads serial data register mn (SDRmn). → | The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data. | This is to prevent an overrun error if the next reception is completed during error processing. |
| Reads serial status register mn (SSRmn). | | Error type is identified and the read value is used to clear error flag. |
| Writes serial flag clear trigger register mn (SIRmn). → | Error flag is cleared. | Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification. |
| Sets the STmn bit of serial channel stop register m (STm) to 1. → | The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation. | Slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission. |
| Creates stop condition. | | |
| Creates start condition. | | |
| Sets the SSmn bit of serial channel start register m (SSm) to 1. → | The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate. | |

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11)
mn = 00, 01, 10, 11

# CHAPTER 16 SERIAL INTERFACE IICA

The number of channels of the serial Interface IICA differs, depending on the product.

| | Groups A, C-1, and D-1 | Groups B, C-2, D-2, and E |
|---|---|---|
| Channel | – | 1 |

**Remark** Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory

Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory

Group C-1: RL78/F13 (CAN incorporated) products with 30 pins and 32 Kbytes to 128 Kbytes of code flash memory

Group C-2: RL78/F13 (CAN incorporated) products with 32, 48, 64, or 80 pins and 32 Kbytes to 128 Kbytes of code flash memory

Group D-1: 30-pin products of the RL78/F14

Group D-2: RL78/F14 products with 32, 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory

Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

**Cautions 1. Most of the following descriptions in this chapter use the Group E products as an example.**

**2. Products of Groups A and C-1 do not have this function.**

## 16.1 Functions of Serial Interface IICA

Serial interface IICA has the following three modes.

**(1) Operation stop mode**

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

**(2) $I^2C$ bus mode (multimaster supported)**

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLA0) line and a serial data bus (SDAA0) line.

This mode complies with the $I^2C$ bus format and the master device can generated “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the $I^2C$ bus.

Since the SCLA0 and SDAA0 pins are used for open drain outputs, serial interface IICA requires pull-up resistors for the serial clock line and the serial data bus line.

**(3) Wakeup mode**

The STOP mode can be released by generating an interrupt request signal (INTIICA0) when an extension code from the master device or a local address has been received while in STOP mode. This can be set by using the WUP0 bit of IICA control register 01 (IICCTL01).

Figure 16-1 shows a block diagram of serial interface IICA.

**Figure 16-1. Block Diagram of Serial Interface IICA**

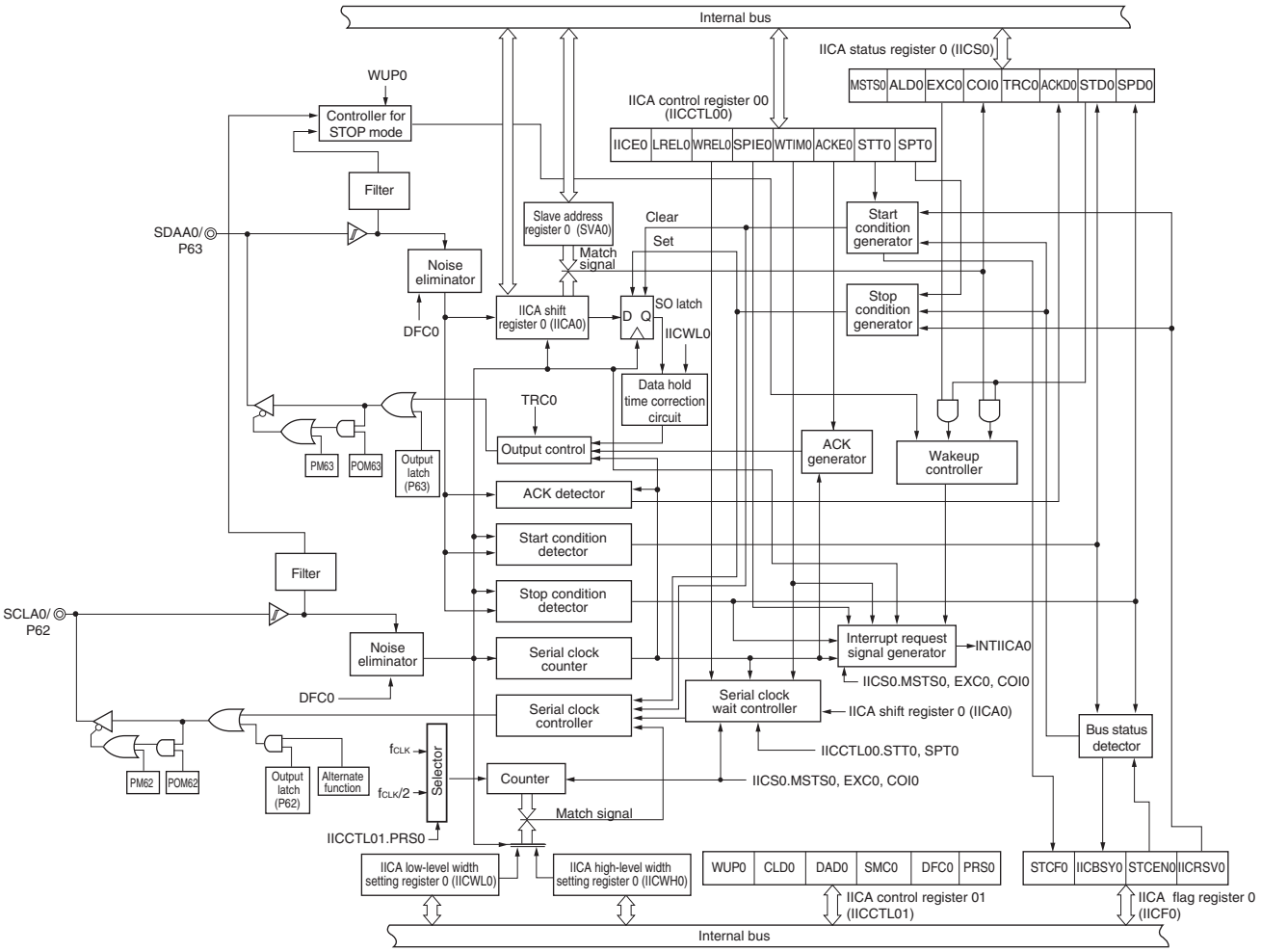
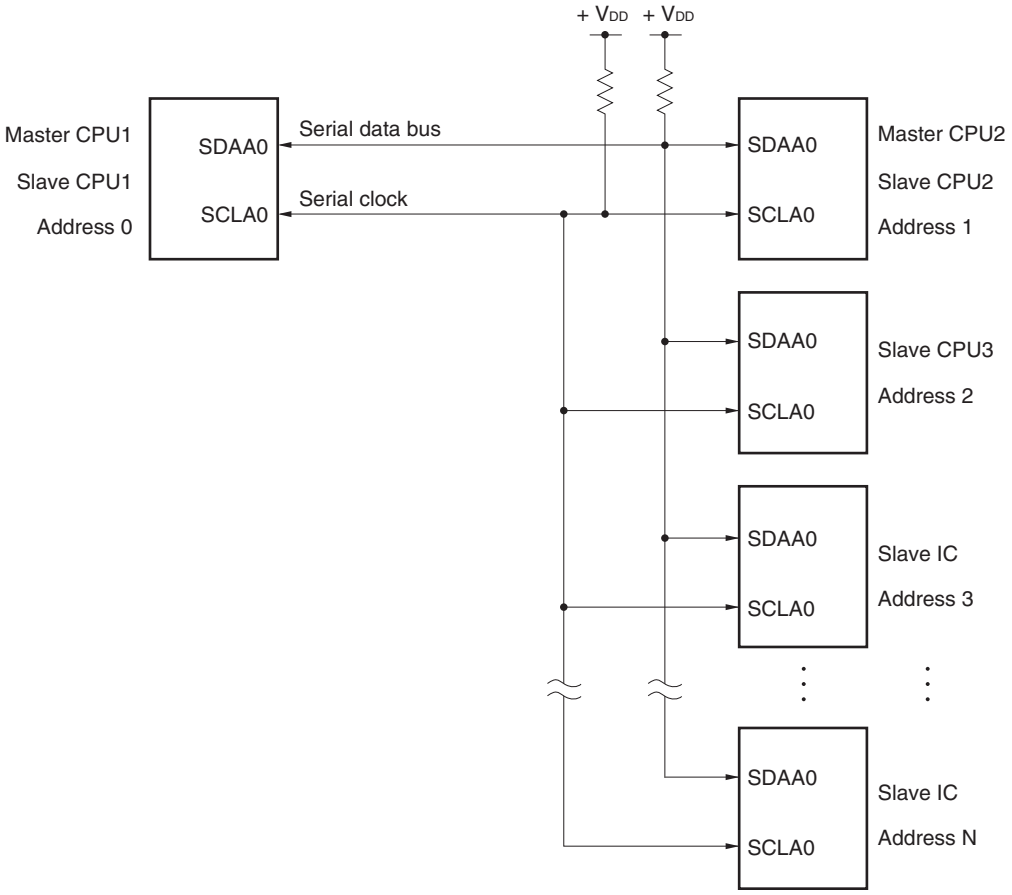


Figure 16-2 shows a serial bus configuration example.

**Figure 16-2. Serial Bus Configuration Example Using I²C Bus**



## 16.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

**Table 16-1. Configuration of Serial Interface IICA**

| Item | Configuration |
|---|---|
| Registers | IICA shift register 0 (IICA0)<br>Slave address register 0 (SVA0) |
| Control registers | Peripheral enable register 0 (PER0)<br>IICA control register 00 (IICCTL00)<br>IICA status register 0 (IICS0)<br>IICA flag register 0 (IICF0)<br>IICA control register 01 (IICCTL01)<br>IICA low-level width setting register 0 (IICWL0)<br>IICA high-level width setting register 0 (IICWH0)<br>Port mode register 6 (PM6)<br>Port register 6 (P6)<br>Port output mode register 6 (POM6) |

### (1) IICA shift register 0 (IICA0)

The IICA0 register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICA0 register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to the IICA0 register.

Cancel the wait state and start data transfer by writing data to the IICA0 register during the wait period.

The IICA0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears IICA0 to 00H.

**Figure 16-3. Format of IICA Shift Register 0 (IICA0)**

Address: FFF50H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| IICA0 | | | | | | | | |

**Cautions 1. Do not write data to the IICA0 register during data transfer.**

**2. Write or read the IICA0 register only during the wait period. Accessing the IICA0 register in a communication state other than during the wait period is prohibited. When the device serves as the master, however, the IICA0 register can be written only once after the communication trigger bit (STT0) is set to 1.**

**3. When communication is reserved, write data to the IICA0 register after the interrupt triggered by a stop condition is detected.**

### (2) Slave address register 0 (SVA0)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

The SVA0 register can be set by an 8-bit memory manipulation instruction.

However, rewriting to this register is prohibited while STD0 = 1 (while the start condition is detected).

Reset signal generation clears the SVA0 register to 00H.

**Figure 16-4. Format of Slave Address Register 0 (SVA0)**

Address: F0234H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| SVA0 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | 0[Note] |

**Note** Bit 0 is fixed to 0.

**(3) SO latch**

The SO latch is used to retain the SDAA0 pin's output level.

**(4) Wakeup controller**

This circuit generates an interrupt request (INTIICA0) when the address received by this register matches the address value set to the slave address register 0 (SVA0) or when an extension code is received.

**(5) Serial clock counter**

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

**(6) Interrupt request signal generator**

This circuit controls the generation of interrupt request signals (INTIICA0).

An $I^2C$ interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by the WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by the SPIE0 bit)

**Remark** WTIM0 bit: Bit 3 of IICA control register 00 (IICCTL00)
SPIE0 bit: Bit 4 of IICA control register 00 (IICCTL00)

**(7) Serial clock controller**

In master mode, this circuit generates the clock output via the SCLA0 pin from a sampling clock.

**(8) Serial clock wait controller**

This circuit controls the wait timing.

**(9) ACK generator, stop condition detector, start condition detector, and ACK detector**

These circuits generate and detect each status.

**(10) Data hold time correction circuit**

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

**(11) Start condition generator**

This circuit generates a start condition when the STT0 bit is set to 1.

However, in the communication reservation disabled status (IICRSV bit = 1), when the bus is not released (IICBSY bit = 1), start condition requests are ignored and the STCF bit is set to 1.

**(12) Stop condition generator**

This circuit generates a stop condition when the SPT0 bit is set to 1.

**(13) Bus status detector**

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCEN0 bit.

**Remark** STT0 bit: Bit 1 of IICA control register 00 (IICCTL00)
SPT0 bit: Bit 0 of IICA control register 00 (IICCTL00)
IICRSV0 bit: Bit 0 of IICA flag register 0 (IICF0)
IICBSY0 bit: Bit 6 of IICA flag register 0 (IICF0)
STCF0 bit: Bit 7 of IICA flag register 0 (IICF0)
STCEN0 bit: Bit 1 of IICA flag register 0 (IICF0)

## 16.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following eight registers.

- Peripheral enable register 0 (PER0)
- IICA control register 00 (IICCTL00)
- IICA flag register 0 (IICF0)
- IICA status register 0 (IICS0)
- IICA control register 01 (IICCTL01)
- IICA low-level width setting register 0 (IICWL0)
- IICA high-level width setting register 0 (IICWH0)
- Port mode register 6 (PM6)
- Port register 6 (P6)
- Port output mode register 6 (POM6)

### 16.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial interface IICA0 is used, be sure to set bit 4 (IICA0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 16-5. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

| Symbol | <7> | 6 | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PER0 | RTCEN | 0 | ADCEN | IICA0EN Notes1, 2 | SAU1EN Note1 | SAU0EN | TAU1EN Note1 | TAU0EN |

| IICA0EN | Control of serial interface IICA0 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• SFR used by serial interface IICA0 cannot be written.<br>• Serial interface IICA0 is in the reset status. |
| 1 | Enables input clock supply.<br>• SFR used by serial interface IICA0 can be read/written. |

**Notes 1.** Not provided in the products of the RL78/F13 (LIN incorporated) with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory.

**2.** Not provided in the 30-pin products of the RL78/F13 (CAN and LIN incorporated) and the 30-pin products of the RL78/F14.

**Cautions 1. When setting serial interface IICA0, be sure to set the IICA0EN bit to 1 first. If IICA0EN = 0, writing to a control register of serial interface IICA0 is ignored, and, even if the register is read, only the default value is read (except for port mode register 6 (PM6) and port register 6 (P6)).**

**2. Be sure to clear the following bits to 0.**

**Bits 1, 3, 4, and 6 in the RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory.**

**Bits 4 and 6 in the 30-pin products of the RL78/F13 (CAN and LIN incorporated) and 30-pin products of the RL78/F14**

**Bit 6 in the products other than above**

### 16.3.2 IICA control register 00 (IICCTL00)

This register is used to enable/stop $I^2C$ operations, set wait timing, and set other $I^2C$ operations.

The IICCTL00 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE0, WTIM0, and ACKE0 bits while IICE0 = 0 or during the wait period. These bits can be set at the same time when the IICE0 bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

**Figure 16-6. Format of IICA Control Register 00 (IICCTL00) (1/4)**

Address: F0230H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IICCTL00 | IICE0 | LREL0 | WREL0 | SPIE0 | WTIM0 | ACKE0 | STT0 | SPT0 |

| IICE0 | $I^2C$ operation enable |
|---|---|
| 0 | Stop operation. Reset the IICA status register 0 (IICS0) Note 1. Stop internal operation. |
| 1 | Enable operation. |
| ★ Be sure to set this bit (1) while the SCLA0 and SDAA0 lines are at high level. | |
| Condition for clearing (IICE0 = 0) | Condition for setting (IICE0 = 1) |
| • Cleared by instruction<br>• Reset | • Set by instruction |

| LREL0 Notes 2, 3 | Exit from communications |
|---|---|
| 0 | Normal operation |
| 1 | This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed.<br>Its uses include cases in which a locally irrelevant extension code has been received.<br>The SCLA0 and SDAA0 lines are set to high impedance.<br>The following flags of IICA control register 00 (IICCTL00) and the IICA status register 0 (IICS0) are cleared to 0.<br>• STT0 • SPT0 • MSTS0 • EXC0 • COI0 • TRC0 • ACKD0 • STD0 |
| The standby mode following exit from communications remains in effect until the following communications entry conditions are met.<br>• After a stop condition is detected, restart is in master mode.<br>• An address match or extension code reception occurs after the start condition. | |
| Condition for clearing (LREL0 = 0) | Condition for setting (LREL0 = 1) |
| • Automatically cleared after execution<br>• Reset | • Set by instruction |

| WREL0 Notes 2, 3 | Wait cancellation |
|---|---|
| 0 | Do not cancel wait |
| 1 | Cancel wait. This setting is automatically cleared after wait is canceled. |
| When the WREL0 bit is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC0 = 1), the SDAA0 line goes into the high impedance state (TRC0 = 0). | |
| Condition for clearing (WREL0 = 0) | Condition for setting (WREL0 = 1) |
| • Automatically cleared after execution<br>• Reset | • Set by instruction |

**Notes 1.** The IICA status register 0 (IICS0), the STCF0 and IICBSY0 bits of the IICA flag register 0 (IICF0), and the CLD0 and DAD0 bits of IICA control register 01 (IICCTL01) are reset.

**2.** The signal of this bit is invalid while IICE0 is 0.

**3.** When the LREL0 and WREL0 bits are read, 0 is always read.

**Caution If the operation of $I^2C$ is enabled (IICE0 = 1) when the SCLA0 line is high level, the SDAA0 line is low level, and the digital filter is turned on (DFC0 bit of IICCTL01 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LREL0 bit by using a 1-bit memory manipulation instruction immediately after enabling operation of $I^2C$ (IICE0 = 1).**

**Figure 16-6. Format of IICA Control Register 00 (IICCTL00) (2/4)**

| SPIE0 Note 1 | Enable/disable generation of interrupt request when stop condition is detected |
|---|---|
| 0 | Disable |
| 1 | Enable |
| If the WUP0 bit of IICA control register 01 (IICCTL01) is 1, no stop condition interrupt will be generated even if SPIE0 = 1. | |
| Condition for clearing (SPIE0 = 0) | Condition for setting (SPIE0 = 1) |
| • Cleared by instruction<br>• Reset | • Set by instruction |

| WTIM0 Note 1 | Control of wait and interrupt request generation |
|---|---|
| 0 | Interrupt request is generated at the eighth clock's falling edge.<br>Master mode: After output of eight clocks, clock output is set to low level and wait is set.<br>Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device. |
| 1 | Interrupt request is generated at the ninth clock's falling edge.<br>Master mode: After output of nine clocks, clock output is set to low level and wait is set.<br>Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device. |
| An interrupt is generated at the falling edge of the ninth clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after an acknowledge ($\overline{ACK}$) is issued. However, when the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock. | |
| Condition for clearing (WTIM0 = 0) | Condition for setting (WTIM0 = 1) |
| • Cleared by instruction<br>• Reset | • Set by instruction |

| ACKE0 Notes 1, 2 | Acknowledgment control |
|---|---|
| 0 | Disable acknowledgment. |
| 1 | Enable acknowledgment. During the ninth clock period, the SDAA0 line is set to low level. |
| Condition for clearing (ACKE0 = 0) | Condition for setting (ACKE0 = 1) |
| • Cleared by instruction<br>• Reset | • Set by instruction |

**Notes 1.** The signal of this bit is invalid while IICE0 is 0. Set this bit during that period.

**2.** The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

**Figure 16-6. Format of IICA Control Register 00 (IICCTL00) (3/4)**

| STT0 Note | Start condition trigger |
|---|---|
| 0 | Do not generate a start condition. |
| 1 | When bus is released (in standby state, when IICBSY0 = 0):<br>If this bit is set (1), a start condition is generated (startup as the master).<br>When a third party is communicating:<br>• When communication reservation function is enabled (IICRSV0 = 0)<br>Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released.<br>• When communication reservation function is disabled (IICRSV0 = 1)<br>Even if this bit is set (1), the STT0 bit is cleared and the STT0 clear flag (STCF0) is set (1). No start condition is generated.<br>In the wait state (when master device):<br>Generates a restart condition after releasing the wait. |

Cautions concerning set timing

- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when the ACKE0 bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as stop condition trigger (SPT0).
- ★ Once STT0 is set (1), setting it again (1) before the clear condition is met is not allowed.

| Condition for clearing (STT0 = 0) | Condition for setting (STT0 = 1) |
|---|---|
| ★ • Cleared by setting the STT0 bit to 1 while communication reservation is prohibited.<br>• Cleared by loss in arbitration<br>• Cleared after start condition is generated by master device<br>• Cleared by LREL0 = 1 (exit from communications)<br>• When IICE0 = 0 (operation stop)<br>• Reset | • Set by instruction |

**Note** The signal of this bit is invalid while IICE0 is 0.

**Remarks 1.** Bit 1 (STT0) becomes 0 when it is read after data setting.

**2.** IICRSV0: Bit 0 of IICA flag register 0 (IICF0)
STCF0: Bit 7 of IICA flag register 0 (IICF0)

**Figure 16-6. Format of IICA Control Register 00 (IICCTL00) (4/4)**

| SPT0 | Stop condition trigger |
|---|---|
| 0 | Stop condition is not generated. |
| 1 | Stop condition is generated (termination of master device's transfer). |

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when the ACKE0 bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as start condition trigger (STT0).
- The SPT0 bit can be set to 1 only when in master mode.
- When the WTIM0 bit has been cleared to 0, if the SPT0 bit is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. The WTIM0 bit should be changed from 0 to 1 during the wait period following the output of eight clocks, and the SPT0 bit should be set to 1 during the wait period that follows the output of the ninth clock.
- Once STT0 is set (1), setting it again (1) before the clear condition is met is not allowed.

| Condition for clearing (SPT0 = 0) | Condition for setting (SPT0 = 1) |
|---|---|
| • Cleared by loss in arbitration<br>• Automatically cleared after stop condition is detected<br>• Cleared by LREL0 = 1 (exit from communications)<br>• When IICE0 = 0 (operation stop)<br>• Reset | • Set by instruction |

**Caution When bit 3 (TRC0) of the IICA status register 0 (IICS0) is set to 1 (transmission status), bit 5 (WREL0) of IICA control register 00 (IICCTL00) is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared (reception status) and the SDAA0 line is set to high impedance. Release the wait performed while the TRC0 bit is 1 (transmission status) by writing to the IICA shift register 0.**

**Remark** Bit 0 (SPT0) becomes 0 when it is read after data setting.

### 16.3.3 IICA status register 0 (IICS0)

This register indicates the status of $I^2C$.

The IICS0 register is read by a 1-bit or 8-bit memory manipulation instruction only when STT0 = 1 and during the wait period.

Reset signal generation clears this register to 00H.

**Caution Reading the IICS0 register while the address match wakeup function is enabled (WUP0 = 1) in STOP mode is prohibited. When the WUP0 bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA0 interrupt request, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (SPIE0 = 1) the interrupt generated by detecting a stop condition and read the IICS0 register after the interrupt has been detected.**

**Remark** STT0: bit 1 of IICA control register 00 (IICCTL00)
WUP0: bit 7 of IICA control register 01 (IICCTL01)

**Figure 16-7. Format of IICA Status Register 0 (IICS0) (1/3)**

Address: FFF51H After reset: 00H R

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IICS0 | MSTS0 | ALD0 | EXC0 | COI0 | TRC0 | ACKD0 | STD0 | SPD0 |

| MSTS0 | Master status check flag |
|---|---|
| 0 | Slave device status or communication standby status |
| 1 | Master device communication status |

| Condition for clearing (MSTS0 = 0) | Condition for setting (MSTS0 = 1) |
|---|---|
| • When a stop condition is detected<br>• When ALD0 = 1 (arbitration loss)<br>• Cleared by LREL0 = 1 (exit from communications)<br>• When the IICE0 bit changes from 1 to 0 (operation stop)<br>• Reset | • When a start condition is generated |

| ALD0 | Detection of arbitration loss |
|---|---|
| 0 | This status means either that there was no arbitration or that the arbitration result was a “win”. |
| 1 | This status indicates the arbitration result was a “loss”. The MSTS0 bit is cleared. |

| Condition for clearing (ALD0 = 0) | Condition for setting (ALD0 = 1) |
|---|---|
| • Automatically cleared after the IICS0 register is read **Note**<br>• When the IICE0 bit changes from 1 to 0 (operation stop)<br>• Reset | • When the arbitration result is a “loss”. |

**Note** This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than the IICS0 register. Therefore, when using the ALD0 bit, read the data of this bit before the data of the other bits.

**Remark** LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

**Figure 16-7. Format of IICA Status Register 0 (IICS0) (2/3)**

| EXC0 | Detection of extension code reception |
|---|---|
| 0 | Extension code was not received. |
| 1 | Extension code was received. |

| Condition for clearing (EXC0 = 0) | Condition for setting (EXC0 = 1) |
|---|---|
| • When a start condition is detected<br>• When a stop condition is detected<br>• Cleared by LREL0 = 1 (exit from communications)<br>• When the IICE0 bit changes from 1 to 0 (operation stop)<br>• Reset | • When the higher four bits of the received address data is either 0000 or 1111 (set at the rising edge of the eighth clock). |

| COI0 | Detection of matching addresses |
|---|---|
| 0 | Addresses do not match. |
| 1 | Addresses match. |

| Condition for clearing (COI0 = 0) | Condition for setting (COI0 = 1) |
|---|---|
| • When a start condition is detected<br>• When a stop condition is detected<br>• Cleared by LREL0 = 1 (exit from communications)<br>• When the IICE0 bit changes from 1 to 0 (operation stop)<br>• Reset | • When the received address matches the local address (slave address register 0 (SVA0)) (set at the rising edge of the eighth clock). |

| TRC0 | Detection of transmit/receive status |
|---|---|
| 0 | Receive status (other than transmit status). The SDAA0 line is set for high impedance. |
| 1 | Transmit status. The value in the SO0 latch is enabled for output to the SDAA0 line (valid starting at the falling edge of the first byte's ninth clock). |

| Condition for clearing (TRC0 = 0) | Condition for setting (TRC0 = 1) |
|---|---|
| <Both master and slave><br>• When a stop condition is detected<br>• Cleared by LREL0 = 1 (exit from communications)<br>• When the IICE0 bit changes from 1 to 0 (operation stop)<br>• Cleared by WREL0 = 1 Note (wait cancel)<br>• When the ALD0 bit changes from 0 to 1 (arbitration loss)<br>• Reset<br>• When not used for communication (MSTS0, EXC0, COI0 = 0)<br><Master><br>• When 1 is output to the LSB (transfer direction specification bit) of the first byte<br><Slave><br>• When a start condition is detected<br>• When 0 is input to the LSB (transfer direction specification bit) of the first byte. | <Master><br>• When a start condition is generated<br>• When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer).<br><Slave><br>• When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte (during address transfer) from the master. |

**Note** When bit 3 (TRC0) of the IICA status register 0 (IICS0) is set to 1 (transmission status), bit 5 (WREL0) of IICA control register 00 (IICCTL00) is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared (reception status) and the SDAA0 line is set to high impedance. Release the wait performed while the TRC0 bit is 1 (transmission status) by writing to the IICA shift register 0.

**Remark** LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

**Figure 16-7. Format of IICA Status Register 0 (IICS0) (3/3)**

| ACKD0 | Detection of acknowledge ($\overline{ACK}$) |
|---|---|
| 0 | Acknowledge was not detected. |
| 1 | Acknowledge was detected. |

| Condition for clearing (ACKD0 = 0) | Condition for setting (ACKD0 = 1) |
|---|---|
| • When a stop condition is detected<br>• At the rising edge of the next byte's first clock<br>• Cleared by LREL0 = 1 (exit from communications)<br>• When the IICE0 bit changes from 1 to 0 (operation stop)<br>• Reset | • After the SDAA0 line is set to low level at the rising edge of SCLA0 line's ninth clock |

| STD0 | Detection of start condition |
|---|---|
| 0 | Start condition was not detected. |
| 1 | Start condition was detected. This indicates that the address transfer period is in effect. |

| Condition for clearing (STD0 = 0) | Condition for setting (STD0 = 1) |
|---|---|
| • When a stop condition is detected<br>• At the rising edge of the next byte's first clock following address transfer<br>• Cleared by LREL0 = 1 (exit from communications)<br>• When the IICE0 bit changes from 1 to 0 (operation stop)<br>• Reset | • When a start condition is detected |

| SPD0 | Detection of stop condition |
|---|---|
| 0 | Stop condition was not detected. |
| 1 | Stop condition was detected. The master device's communication is terminated and the bus is released. |

| Condition for clearing (SPD0 = 0) | Condition for setting (SPD0 = 1) |
|---|---|
| • At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition<br>• When the WUP0 bit changes from 1 to 0<br>• When the IICE0 bit changes from 1 to 0 (operation stop)<br>• Reset | • When a stop condition is detected |

**Remark** LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

### 16.3.4 IICA flag register 0 (IICF0)

This register sets the operation mode of $I^2C$ and indicates the status of the $I^2C$ bus.

The IICF0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STT0 clear flag (STCF0) and $I^2C$ bus status flag (IICBSY0) bits are read-only.

The IICRSV0 bit can be used to enable/disable the communication reservation function.

The STCEN0 bit can be used to set the initial value of the IICBSY bit.

The IICRSV0 and STCEN0 bits can be written only when the operation of $I^2C$ is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) = 0). When operation is enabled, the IICF0 register can be read.

Reset signal generation clears this register to 00H.

**Figure 16-8. Format of IICA Flag Register 0 (IICF0)**

Address: FFF52H After reset: 00H R/W[Note]

| Symbol | <7> | <6> | 5 | 4 | 3 | 2 | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IICF0 | STCF0 | IICBSY0 | 0 | 0 | 0 | 0 | STCEN0 | IICRSV0 |

| STCF0 | STT0 clear flag |
|---|---|
| 0 | Generate start condition |
| 1 | Start condition generation unsuccessful: clear the STT0 flag |

| Condition for clearing (STCF0 = 0) | Condition for setting (STCF0 = 1) |
|---|---|
| • Cleared by STT0 = 1<br>• When IICE0 = 0 (operation stop)<br>• Reset | • Generating start condition unsuccessful and the STT0 bit cleared to 0 when communication reservation is disabled (IICRSV0 = 1). |

| IICBSY0 | $I^2C$ bus status flag |
|---|---|
| 0 | Bus release status (communication initial status when STCEN0 = 1) |
| 1 | Bus communication status (communication initial status when STCEN0 = 0) |

| Condition for clearing (IICBSY0 = 0) | Condition for setting (IICBSY0 = 1) |
|---|---|
| • Detection of stop condition<br>• When IICE0 = 0 (operation stop)<br>• Reset | • Detection of start condition<br>• Setting of the IICE0 bit when STCEN0 = 0 |

| STCEN0 | Initial start enable trigger |
|---|---|
| 0 | After operation is enabled (IICE0 = 1), enable generation of a start condition upon detection of a stop condition. |
| 1 | After operation is enabled (IICE0 = 1), enable generation of a start condition without detecting a stop condition. |

| Condition for clearing (STCEN0 = 0) | Condition for setting (STCEN0 = 1) |
|---|---|
| • Cleared by instruction<br>• Detection of start condition<br>• Reset | • Set by instruction |

| IICRSV0 | Communication reservation function disable bit |
|---|---|
| 0 | Enable communication reservation |
| 1 | Disable communication reservation |

| Condition for clearing (IICRSV0 = 0) | Condition for setting (IICRSV0 = 1) |
|---|---|
| • Cleared by instruction<br>• Reset | • Set by instruction |

**Note** Bits 6 and 7 are read-only.

**Cautions 1. Write to the STCEN bit only when the operation is stopped (IICE0 = 0).**

**2. As the bus release status (IICBSY0 = 0) is recognized regardless of the actual bus status when STCEN0 = 1, when generating the first start condition (STT0 = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.**

**3. Write to IICRSV0 only when the operation is stopped (IICE0 = 0).**

**Remark** STT0: Bit 1 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

### 16.3.5 IICA control register 01 (IICCTL01)

This register is used to set the operation mode of $I^2C$ and detect the statuses of the SCLA0 and SDAA0 pins.

The IICCTL01 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD0 and DAD0 bits are read-only.

Set the IICCTL01 register, except the WUP0 bit, while operation of $I^2C$ is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation clears this register to 00H.

**Figure 16-9. Format of IICA Control Register 01 (IICCTL01) (1/2)**

Address: F0231H After reset: 00H R/W Note 1

| Symbol | <7> | 6 | <5> | <4> | <3> | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| IICCTL01 | WUP0 | 0 | CLD0 | DAD0 | SMC0 | DFC0 | 0 | PRS0 |

| WUP0 | Control of address match wakeup |
|---|---|
| 0 | Stops operation of address match wakeup function in STOP mode. |
| 1 | Enables operation of address match wakeup function in STOP mode. |

To shift to STOP mode when WUP0 = 1, execute the STOP instruction at least three cycles of the operation clock ($f_{MCK}$) after setting (1) the WUP0 bit (see **Figure 16-23 Flow When Setting WUP0 = 1**).
Clear (0) the WUP0 bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUP0 bit. (The wait must be released and transmit data must be written after the WUP0 bit has been cleared (0).)
The interrupt timing when the address has matched or when an extension code has been received, while WUP0 = 1, is identical to the interrupt timing when WUP0 = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUP0 = 1, a stop condition interrupt is not generated even if the SPIE0 bit is set to 1.
When WUP0 = 0 is set by a source other than an interrupt from serial interface IICA, operation as the master device cannot be performed until the subsequent start condition or stop condition is detected. Do not output a start condition by setting (1) the STT0 bit, without waiting for the detection of the subsequent start condition or stop condition.

| Condition for clearing (WUP0 = 0) | Condition for setting (WUP0 = 1) |
|---|---|
| • Cleared by instruction (after address match or extension code reception) | • Set by instruction (when the MSTS0, EXC0, and COI0 bits are “0”, and the STD0 bit also “0” (communication not entered)) Note 2 |

**Notes 1.** Bits 4 and 5 are read-only.

**2.** The status of the IICA status register 0 (IICS0) must be checked and the WUP0 bit must be set during the period shown below.

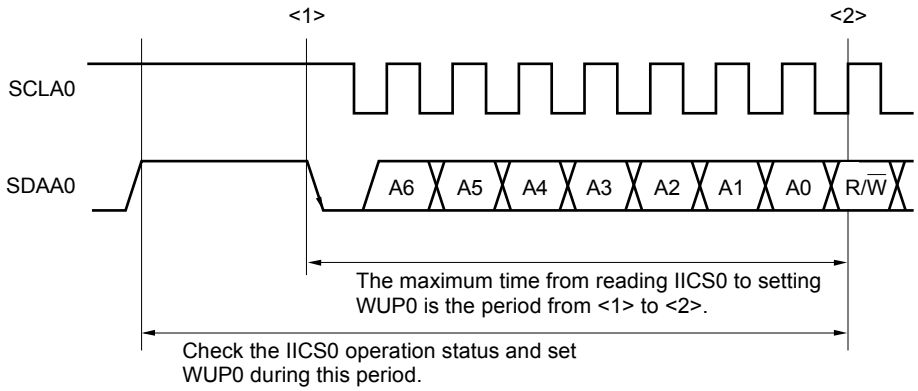


**Figure 16-9. Format of IICA Control Register 01 (IICCTL01) (2/2)**

| CLD0 | Detection of SCLA0 pin level (valid only when IICE0 = 1) |
|---|---|
| 0 | The SCLA0 pin was detected at low level. |
| 1 | The SCLA0 pin was detected at high level. |

| Condition for clearing (CLD0 = 0) | Condition for setting (CLD0 = 1) |
|---|---|
| • When the SCLA0 pin is at low level<br>• When IICE0 = 0 (operation stop)<br>• Reset | • When the SCLA0 pin is at high level |

| DAD0 | Detection of SDAA0 pin level (valid only when IICE0 = 1) |
|---|---|
| 0 | The SDAA0 pin was detected at low level. |
| 1 | The SDAA0 pin was detected at high level. |

| Condition for clearing (DAD0 = 0) | Condition for setting (DAD0 = 1) |
|---|---|
| • When the SDAA0 pin is at low level<br>• When IICE0 = 0 (operation stop)<br>• Reset | • When the SDAA0 pin is at high level |

| SMC0 | Operation mode switching |
|---|---|
| 0 | Operates in standard mode (fastest transfer rate: 100 kbps). |
| 1 | Operates in fast mode (fastest transfer rate: 400 kbps) or fast mode plus (fastest transfer rate: 1 Mbps). |

| DFC0 | Digital filter operation control |
|---|---|
| 0 | Digital filter off. |
| 1 | Digital filter on. |
| Digital filter can be used only in fast mode.<br>In fast mode and fast mode plus, the transfer clock does not vary, regardless of the DFC0 bit being set (1) or cleared (0).<br>The digital filter is used for noise elimination in fast mode and fast mode plus. | |

| PRS0 | Control of the operation clock for IICA ($f_{MCK}$) |
|---|---|
| 0 | Selects $f_{CLK}$ (1 MHz ≤ $f_{CLK}$ ≤ 20 MHz). |
| 1 | Selects $f_{CLK}/2$ (20 MHz < $f_{CLK}$). |

**Cautions 1. The fastest operation frequency of the operation clock for IICA ($f_{MCK}$) is 20 MHz (max.). Set bit 0 (PRS0) of the IICA control register 01 (IICCTL01) to 1 only when the $f_{CLK}$ exceeds 20 MHz.**

**2. Note the minimum $f_{CLK}$ operation frequency when setting the transfer clock. The minimum $f_{CLK}$ operation frequency for serial interface IICA is determined according to the mode.**

**Fast mode: $f_{CLK}$ = 3.5 MHz (min.)**
**Fast mode plus: $f_{CLK}$ = 10 MHz (min.)**
**Normal mode: $f_{CLK}$ = 1 MHz (min.)**

**Remark** IICE0: Bit 7 of IICA control register 00 (IICCTL00)

### 16.3.6 IICA low-level width setting register 0 (IICWL0)

This register is used to set the low-level width ($t_{LOW}$) of the SCLA0 pin signal that is output by serial interface IICA.

The IICWL0 register can be set by an 8-bit memory manipulation instruction.

Set the IICWL0 register while operation of $I^2C$ is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation sets this register to FFH.

For details about setting the IICWL0 register, see **16.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers**.

**Figure 16-10. Format of IICA Low-Level Width Setting Register 0 (IICWL0)**

Address: F0232H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| IICWL0 | | | | | | | | |

### 16.3.7 IICA high-level width setting register 0 (IICWH0)

This register is used to set the high-level width of the SCLA0 pin signal that is output by serial interface IICA.

The IICWH0 register can be set by an 8-bit memory manipulation instruction.

Set the IICWH0 register while operation of $I^2C$ is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation sets this register to FFH.

**Figure 16-11. Format of IICA High-Level Width Setting Register 0 (IICWH0)**

Address: F0233H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| IICWH0 | | | | | | | | |

**Remark** For how to set the transfer clock by using the IICWL0 and IICWH0 registers, see **16.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers**.

### 16.3.8 Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P62/SCLA0 pin as clock I/O and the P63/SDAA0 pin as serial data I/O, clear P62, P63, and the output latches of P62 and P63 to 0.

Set the IICE0 bit (bit 7 of IICA control register 00 (IICCTL00)) to 1 before setting the output mode because the P62/SCLA0 and P63/SDAA0 pins output a low level (fixed) when the IICE0 bit is 0.

The PM6 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

**Figure 16-12. Format of Port Mode Register 6 (PM6)**

Address: FFF26H After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PM6 | 1 | 1 | 1 | 1 | PM63 | PM62 | PM61 | PM60 |

| PM6n | P6n pin I/O mode selection (n = 0 to 3) |
|---|---|
| 0 | Output mode (output buffer on) |
| 1 | Input mode (output buffer off) |

**Caution PM62 and PM63 are used for the IICA serial interface.**

### 16.3.9 Port output mode register (POM6)

This register sets the output mode of P60 to P63 in 1-bit units.

N-ch open drain output ($EV_{DD0}$ tolerance) mode can be selected for the SCLA0 and SDAA0 pins during $I^2C$ communication.

When using the P62/SCLA0 pin as clock I/O and the P63/SDAA0 pin as serial data I/O, set POM62 and POM63 to 1.

Set the IICE0 bit (bit 7 of IICA control register 00 (IICCTL00)) to 1 before setting the output mode because the P62/SCLA0 and P63/SDAA0 pins output a low level (fixed) when the IICE0 bit is 0.

The POM6 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 16-13. Format of Port Output Mode Register 6 (POM6)**

Address: F0056H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| POM6 | 0 | 0 | 0 | 0 | POM63 | POM62 | POM61 | POM60 |

| POMmn | P6n pin output mode selection (n = 0 to 3) |
|---|---|
| 0 | Normal output mode |
| 1 | N-ch open-drain output ($EV_{DD0}$ tolerance) mode |

**Caution POM62 and POM63 are used for the IICA serial interface.**

## 16.4 I²C Bus Mode Functions

### 16.4.1 Pin configuration

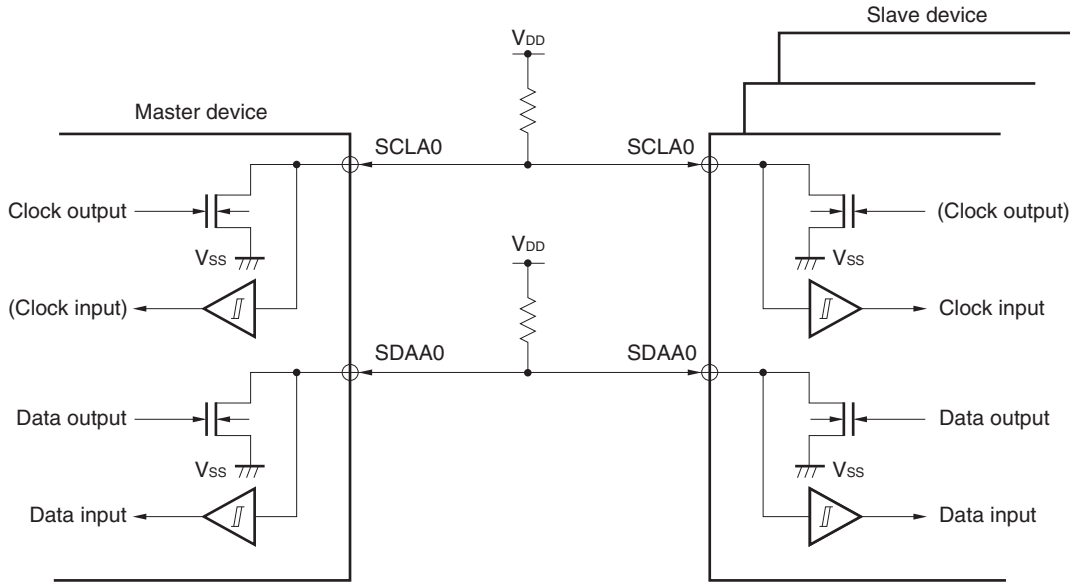
The serial clock pin (SCLA0) and the serial data bus pin (SDAA0) are configured as follows.

(1) SCLA0 .... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDAA0 .... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

**Figure 16-14. Pin Configuration Diagram**



### 16.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers

#### (1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{MCK}}{IICWL0 + IICWH0 + f_{MCK}(t_R + t_F)}$$

At this time, the optimal setting values of the IICWL0 and IICWH0 registers are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICWL0 = \frac{0.52}{\text{Transfer clock}} \times f_{MCK}$$
$$IICWH0 = (\frac{0.48}{\text{Transfer clock}} - t_R - t_F) \times f_{MCK}$$

- When the normal mode

$$IICWL0 = \frac{0.47}{\text{Transfer clock}} \times f_{MCK}$$
$$IICWH0 = (\frac{0.53}{\text{Transfer clock}} - t_R - t_F) \times f_{MCK}$$

- When the fast mode plus

$$IICWL0 = \frac{0.50}{\text{Transfer clock}} \times f_{MCK}$$
$$IICWH0 = (\frac{0.50}{\text{Transfer clock}} - t_R - t_F) \times f_{MCK}$$

#### (2) Setting IICWL0 and IICWH0 registers on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$IICWL0 = 1.3\ \mu s \times f_{MCK}$$
$$IICWH0 = (1.2\ \mu s - t_R - t_F) \times f_{MCK}$$

- When the normal mode

$$IICWL0 = 4.7\ \mu s \times f_{MCK}$$
$$IICWH0 = (5.3\ \mu s - t_R - t_F) \times f_{MCK}$$

- When the fast mode plus

$$IICWL0 = 0.50\ \mu s \times f_{MCK}$$
$$IICWH0 = (0.50\ \mu s - t_R - t_F) \times f_{MCK}$$

(**Cautions** and **Remarks** are listed on the next page.)

**Cautions 1. The fastest operation frequency of the operation clock for IICA ($f_{MCK}$) is 20 MHz (max.). Set bit 0 (PRS0) of the IICA control register 01 (IICCTL01) to 1 only when the $f_{CLK}$ exceeds 20 MHz.**

**2. Note the minimum $f_{CLK}$ operation frequency when setting the transfer clock. The minimum $f_{CLK}$ operation frequency for serial interface IICA is determined according to the mode.**

**Fast mode: $f_{CLK}$ = 3.5 MHz (min.)**

**Fast mode plus: $f_{CLK}$ = 10 MHz (min.)**

**Normal mode: $f_{CLK}$ = 1 MHz (min.)**

**Remarks 1.** Calculate the rise time ($t_R$) and fall time ($t_F$) of the SDAA0 and SCLA0 signals separately, because they differ depending on the pull-up resistance and wire load.

**2.** IICWL0: IICA low-level width setting register 0

IICWH0: IICA high-level width setting register 0

$t_F$: SDAA0 and SCLA0 signal falling times

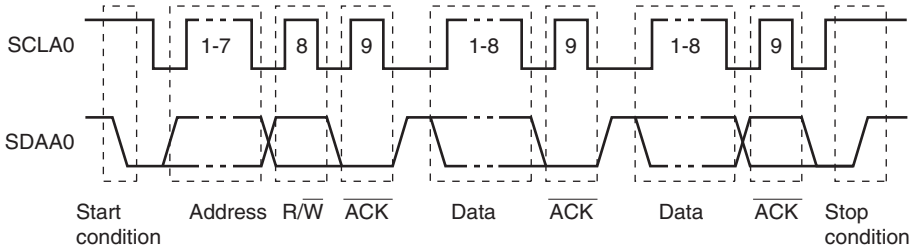
$t_R$: SDAA0 and SCLA0 signal rising times

$f_{MCK}$: Frequency of the IICA operation clock

## 16.5 I²C Bus Definitions and Control Methods

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus. Figure 16-15 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

**Figure 16-15. I²C Bus Serial Data Transfer Timing**



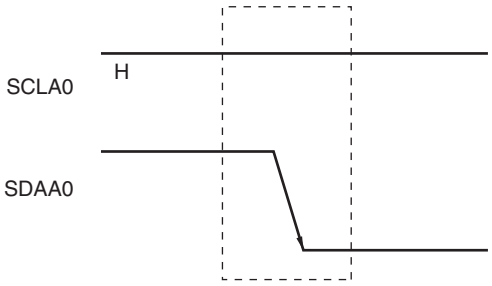

The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLA0) is continuously output by the master device. However, in the slave device, the SCLA0 pin low level period can be extended and a wait can be inserted.

### 16.5.1 Start conditions

A start condition is met when the SCLA0 pin is at high level and the SDAA0 pin changes from high level to low level. The start conditions for the SCLA0 pin and SDAA0 pin are signals that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

**Figure 16-16. Start Conditions**




A start condition is output when bit 1 (STT0) of IICA control register 00 (IICCTL00) is set (1) after a stop condition has been detected (SPD0: Bit 0 of the IICA status register 0 (IICS0) = 1). When a start condition is detected, bit 1 (STD0) of the IICS0 register is set (1).

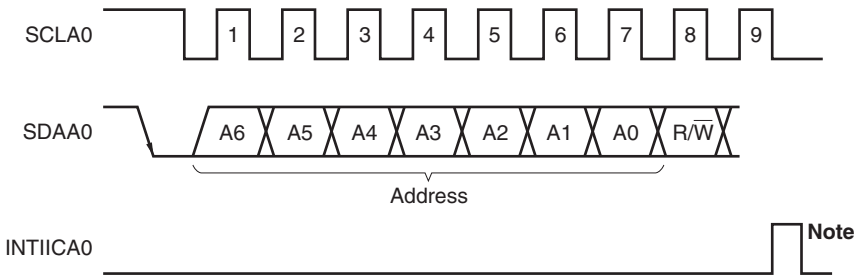
### 16.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register 0 (SVA0). If the address data matches the SVA0 register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

**Figure 16-17. Address**




**Note** INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **16.5.3 Transfer direction specification** are written to the IICA shift register 0 (IICA0). The received addresses are written to the IICA0 register.

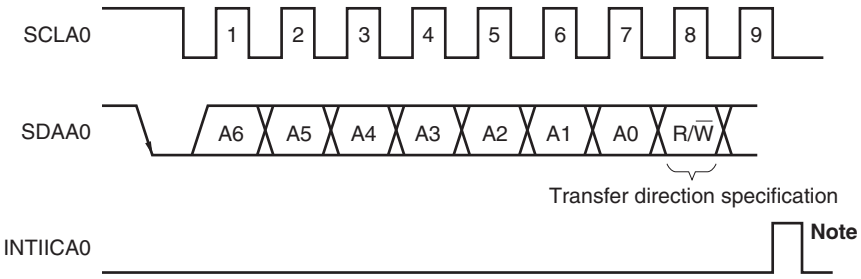
The slave address is assigned to the higher 7 bits of the IICA0 register.

### 16.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit has a value of “0”, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of “1”, it indicates that the master device is receiving data from a slave device.

**Figure 16-18. Transfer Direction Specification**




**Note** INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

### 16.5.4 Acknowledge ($\overline{ACK}$)

$\overline{ACK}$ is used to check the status of serial data at the transmission and reception sides.

The reception side returns $\overline{ACK}$ each time it has received 8-bit data.

The transmission side usually receives $\overline{ACK}$ after transmitting 8-bit data. When $\overline{ACK}$ is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether $\overline{ACK}$ has been detected can be checked by using bit 2 (ACKD0) of the IICA status register 0 (IICS0).

When the master receives the last data item, it does not return $\overline{ACK}$ and instead generates a stop condition. If a slave does not return $\overline{ACK}$ after receiving data, the master outputs a stop condition or restart condition and stops transmission. If $\overline{ACK}$ is not returned, the possible causes are as follows.

<1> Reception was not performed normally.
<2> The final data item was received.
<3> The reception side specified by the address does not exist.

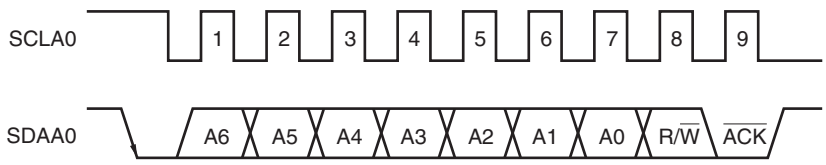
To generate $\overline{ACK}$, the reception side makes the SDAA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{ACK}$ is enabled by setting bit 2 (ACKE0) of IICA control register 00 (IICCTL00) to 1. Bit 3 (TRC0) of the IICS0 register is set by the data of the eighth bit that follows 7-bit address information. Usually, set the ACKE0 bit to 1 for reception (TRC0 = 0).

If a slave can receive no more data during reception (TRC0 = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKE0 bit to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC0 = 0), it must clear the ACKE0 bit to 0 so that $\overline{ACK}$ is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

**Figure 16-19. $\overline{ACK}$**




When the local address is received, $\overline{ACK}$ is automatically generated, regardless of the value of the ACKE0 bit. When an address other than that of the local address is received, $\overline{ACK}$ is not generated (NACK).

When an extension code is received, $\overline{ACK}$ is generated if the ACKE0 bit is set to 1 in advance.

How $\overline{ACK}$ is generated when data is received differs as follows depending on the setting of the wait timing.

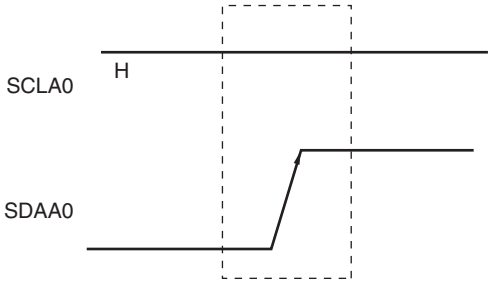
- When 8-clock wait state is selected (bit 3 (WTIM0) of IICCTL00 register = 0):
  By setting the ACKE0 bit to 1 before releasing the wait state, $\overline{ACK}$ is generated at the falling edge of the eighth clock of the SCLA0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM0) of IICCTL00 register = 1):
  $\overline{ACK}$ is generated by setting the ACKE0 bit to 1 in advance.

### 16.5.5 Stop condition

When the SCLA0 pin is at high level, changing the SDAA0 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

**Figure 16-20. Stop Condition**




A stop condition is generated when bit 0 (SPT0) of IICA control register 00 (IICCTL00) is set to 1. When the stop condition is detected, bit 0 (SPD0) of the IICA status register 0 (IICS0) is set to 1 and INTIICA0 is generated when bit 4 (SPIE0) of the IICCTL00 register is set to 1.

### 16.5.6 Wait

The wait is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCLA0 pin to low level notifies the communication partner of the wait state. When wait state has been canceled for both the master and slave devices, the next data transfer can begin.

**Figure 16-21. Wait (1/2)**

**(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKE0 = 1)**

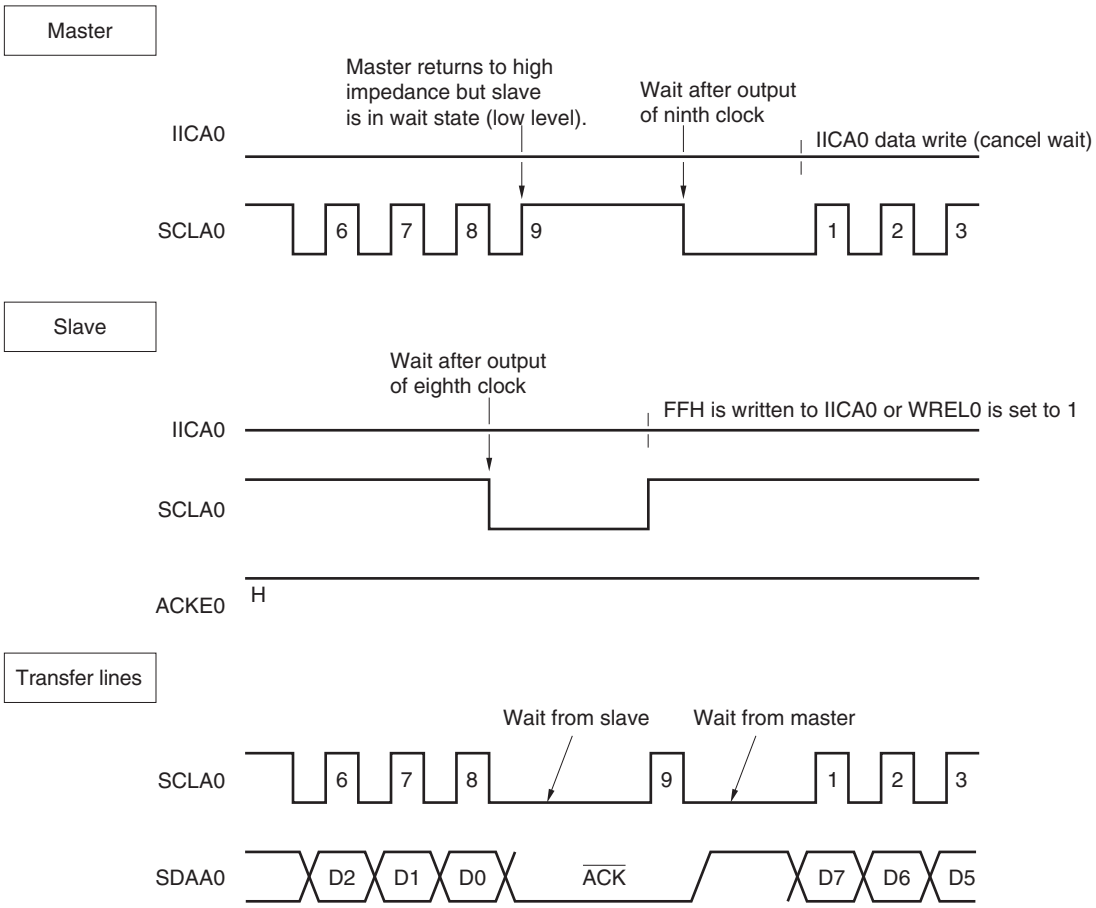
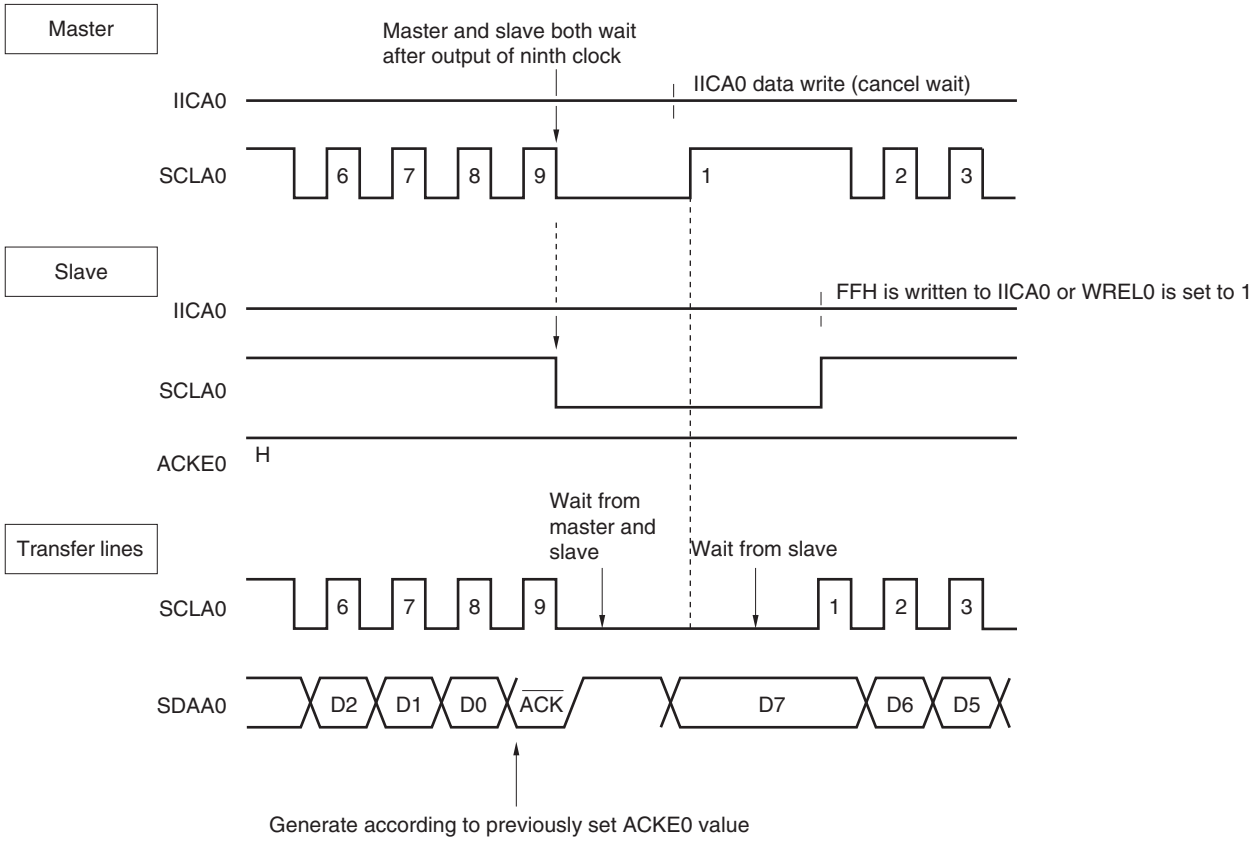


**Figure 16-21. Wait (2/2)**

**(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE0 = 1)**




**Remark** ACKE0: Bit 2 of IICA control register 00 (IICCTL00)
WREL0: Bit 5 of IICA control register 00 (IICCTL00)

A wait may be automatically generated depending on the setting of bit 3 (WTIM0) of IICA control register 00 (IICCTL00).

Normally, the receiving side cancels the wait state when bit 5 (WREL0) of the IICCTL00 register is set to 1 or when FFH is written to the IICA shift register 0 (IICA0), and the transmitting side cancels the wait state when data is written to the IICA0 register.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT0) of the IICCTL00 register to 1
- By setting bit 0 (SPT0) of the IICCTL00 register to 1

### 16.5.7 Canceling wait

The $I^2C$ usually cancels a wait state by the following processing.

- Writing data to the IICA shift register 0 (IICA0)
- Setting bit 5 (WREL0) of IICA control register 00 (IICCTL00) (canceling wait)
- Setting bit 1 (STT0) of the IICCTL00 register (generating start condition) **Note**
- Setting bit 0 (SPT0) of the IICCTL00 register (generating stop condition) **Note**

**Note** Master only

When the above wait canceling processing is executed, the $I^2C$ cancels the wait state and communication is resumed.

To cancel a wait state and transmit data (including addresses), write the data to the IICA0 register.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL0) of the IICCTL00 register to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT0) of the IICCTL00 register to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT0) of the IICCTL00 register to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to the IICA0 register after canceling a wait state by setting the WREL0 bit to 1, an incorrect value may be output to SDAA0 line because the timing for changing the SDAA0 line conflicts with the timing for writing the IICA0 register.

In addition to the above, communication is stopped if the IICE0 bit is cleared to 0 when communication has been aborted, so that the wait state can be canceled.

If the $I^2C$ bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LREL0) of the IICCTL00 register, so that the wait state can be canceled.

**Caution If a processing to cancel a wait state is executed when WUP0 = 1, the wait state will not be canceled.**

### 16.5.8 Interrupt request (INTIICA0) generation timing and wait control

The setting of bit 3 (WTIM0) of IICA control register 00 (IICCTL00) determines the timing by which INTIICA0 is generated and the corresponding wait control, as shown in Table 16-2.

**Table 16-2. INTIICA0 Generation Timing and Wait Control**

| WTIM0 | During Slave Device Operation | | | During Master Device Operation | | |
|---|---|---|---|---|---|---|
| | Address | Data Reception | Data Transmission | Address | Data Reception | Data Transmission |
| 0 | 9 Notes 1, 2 | 8 Note 2 | 8 Note 2 | 9 | 8 | 8 |
| 1 | 9 Notes 1, 2 | 9 Note 2 | 9 Note 2 | 9 | 9 | 9 |

**Notes 1.** The slave device's INTIICA0 signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to the slave address register 0 (SVA0).
At this point, $\overline{ACK}$ is generated regardless of the value set to the IICCTL00 register's bit 2 (ACKE0). For a slave device that has received an extension code, INTIICA0 occurs at the falling edge of the eighth clock. However, if the address does not match after restart, INTIICA0 is generated at the falling edge of the 9th clock, but wait does not occur.

**2.** If the received address does not match the contents of the slave address register 0 (SVA0) and extension code is not received, neither INTIICA0 nor a wait occurs.

**Remark** The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

#### (1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIM0 bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

#### (2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

#### (3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

#### (4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to the IICA shift register 0 (IICA0)
- Setting bit 5 (WREL0) of IICA control register 00 (IICCTL00) (canceling wait)
- Setting bit 1 (STT0) of IICCTL00 register (generating start condition)Note
- Setting bit 0 (SPT0) of IICCTL00 register (generating stop condition)Note

**Note** Master only.

When an 8-clock wait has been selected (WTIM0 = 0), the presence/absence of $\overline{ACK}$ generation must be determined prior to wait cancellation.

#### (5) Stop condition detection

INTIICA0 is generated when a stop condition is detected (only when SPIE0 = 1).

### 16.5.9 Address match detection method

In $I^2C$ bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt request (INTIICA0) occurs when the address set to the slave address register 0 (SVA0) matches the slave address sent by the master device, or when an extension code has been received.

### 16.5.10 Error detection

In $I^2C$ bus mode, the status of the serial data bus (SDAA0) during data transmission is captured by the IICA shift register 0 (IICA0) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

### 16.5.11 Extension code

(1) When the higher 4 bits of the receive address are either 0000 or 1111, the extension code reception flag (EXC0) is set to 1 for extension code reception and an interrupt request (INTIICA0) is issued at the falling edge of the eighth clock. The local address stored in the slave address register 0 (SVA0) is not affected.

(2) The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer when the SVA0 register is set to 11110xx0. Note that INTIICA0 occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC0 = 1
- Seven bits of data match: COI0 = 1

**Remark** EXC0: Bit 5 of IICA status register 0 (IICS0)
COI0: Bit 4 of IICA status register 0 (IICS0)

(3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
If the extension code is received while a slave device is operating, then the slave device is participating in communication even if its address does not match.
For example, after the extension code is received, if you do not wish to operate the target device as a slave device, set bit 6 (LREL0) of IICA control register 00 (IICCTL00) to 1 to set the standby mode for the next communication operation.

**Table 16-3. Bit Definitions of Major Extension Codes**

| Slave Address | R/W Bit | Description |
|---|---|---|
| 0 0 0 0 0 0 0 | 0 | General call address |
| 1 1 1 1 0 x x | 0 | 10-bit slave address specification (during address authentication) |
| 1 1 1 1 0 x x | 1 | 10-bit slave address specification (after address match, when read command is issued) |

**Remark** See the $I^2C$ bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

### 16.5.12 Arbitration

When several master devices simultaneously generate a start condition (when the STT0 bit is set to 1 before the STD0 bit is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (ALD0) in the IICA status register 0 (IICS0) is set (1) via the timing by which the arbitration loss occurred, and the SCLA0 and SDAA0 lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 = 1 setting that has been made by software.

For details of interrupt request timing, see **16.5.8 Interrupt request (INTIICA0) generation timing and wait control**.

**Remark** STD0: Bit 1 of IICA status register 0 (IICS0)
STT0: Bit 1 of IICA control register 00 (IICCTL00)

**Figure 16-22. Arbitration Timing Example**

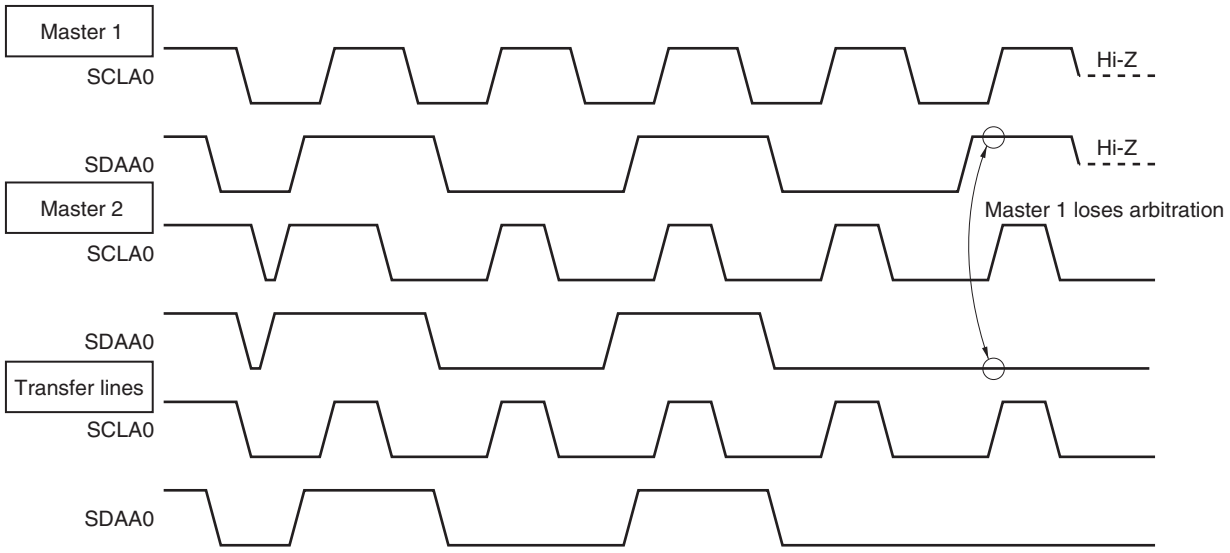


**Table 16-4. Status During Arbitration and Interrupt Request Generation Timing**

| Status During Arbitration | Interrupt Request Generation Timing |
|---|---|
| During address transmission | At falling edge of eighth or ninth clock following byte transfer Note 1 |
| Read/write data after address transmission | |
| During extension code transmission | |
| Read/write data after extension code transmission | |
| During data transmission | |
| During $\overline{\text{ACK}}$ transfer period after data transmission | |
| When restart condition is detected during data transfer | |
| When stop condition is detected during data transfer | When stop condition is generated (when SPIE0 = 1) Note 2 |
| When data is at low level while attempting to generate a restart condition | At falling edge of eighth or ninth clock following byte transfer Note 1 |
| When stop condition is detected while attempting to generate a restart condition | When stop condition is generated (when SPIE0 = 1) Note 2 |
| When data is at low level while attempting to generate a stop condition | At falling edge of eighth or ninth clock following byte transfer Note 1 |
| When SCLA0 is at low level while attempting to generate a restart condition | |

**Notes 1.** When the WTIM0 bit (bit 3 of IICA control register 00 (IICCTL00)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0 = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

**2.** When there is a chance that arbitration will occur, set SPIE0 = 1 for master device operation.

**Remark** SPIE0: Bit 4 of IICA control register 00 (IICCTL00)

### 16.5.13 Wakeup function

The $I^2C$ bus slave function is a function that generates an interrupt request signal (INTIICA0) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIICA0 signal from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

To use the wakeup function in the STOP mode, set the WUP0 bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICA0) is also generated when a local address and extension code have been received. Operation returns to normal operation by using an instruction to clear (0) the WUP0 bit after this interrupt has been generated.

Figure 16-23 shows the flow for setting WUP0 = 1 and Figure 16-24 shows the flow for setting WUP0 = 0 upon an address match.

**Figure 16-23. Flow When Setting WUP0 = 1**

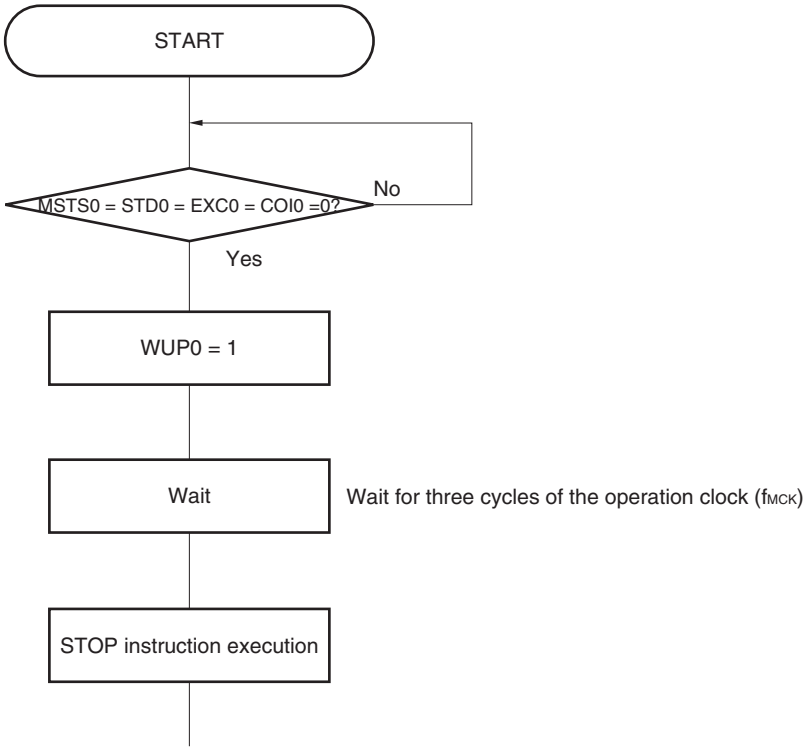
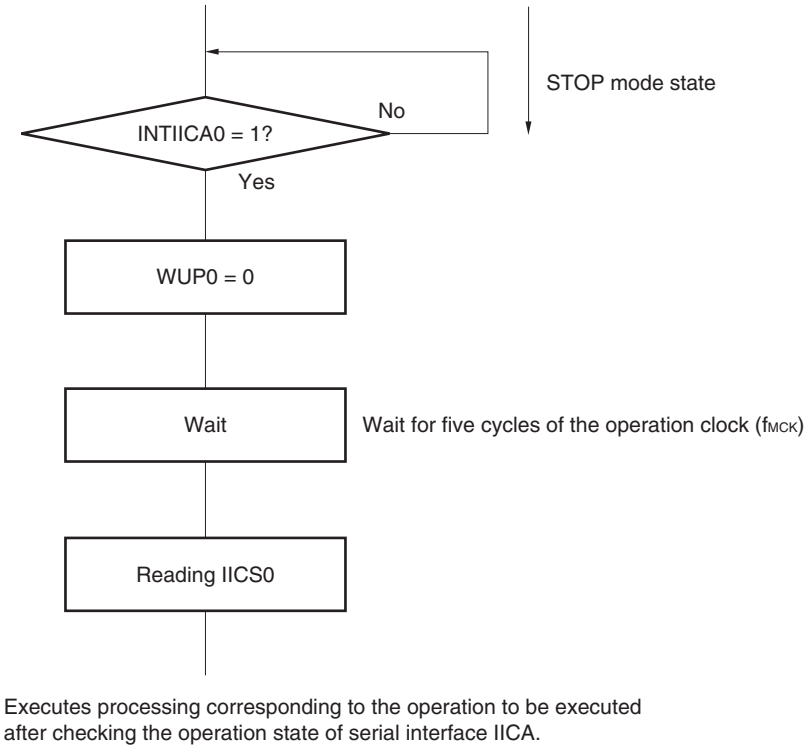


**Figure 16-24. Flow When Setting WUP0 = 0 upon Address Match (Including Extension Code Reception)**

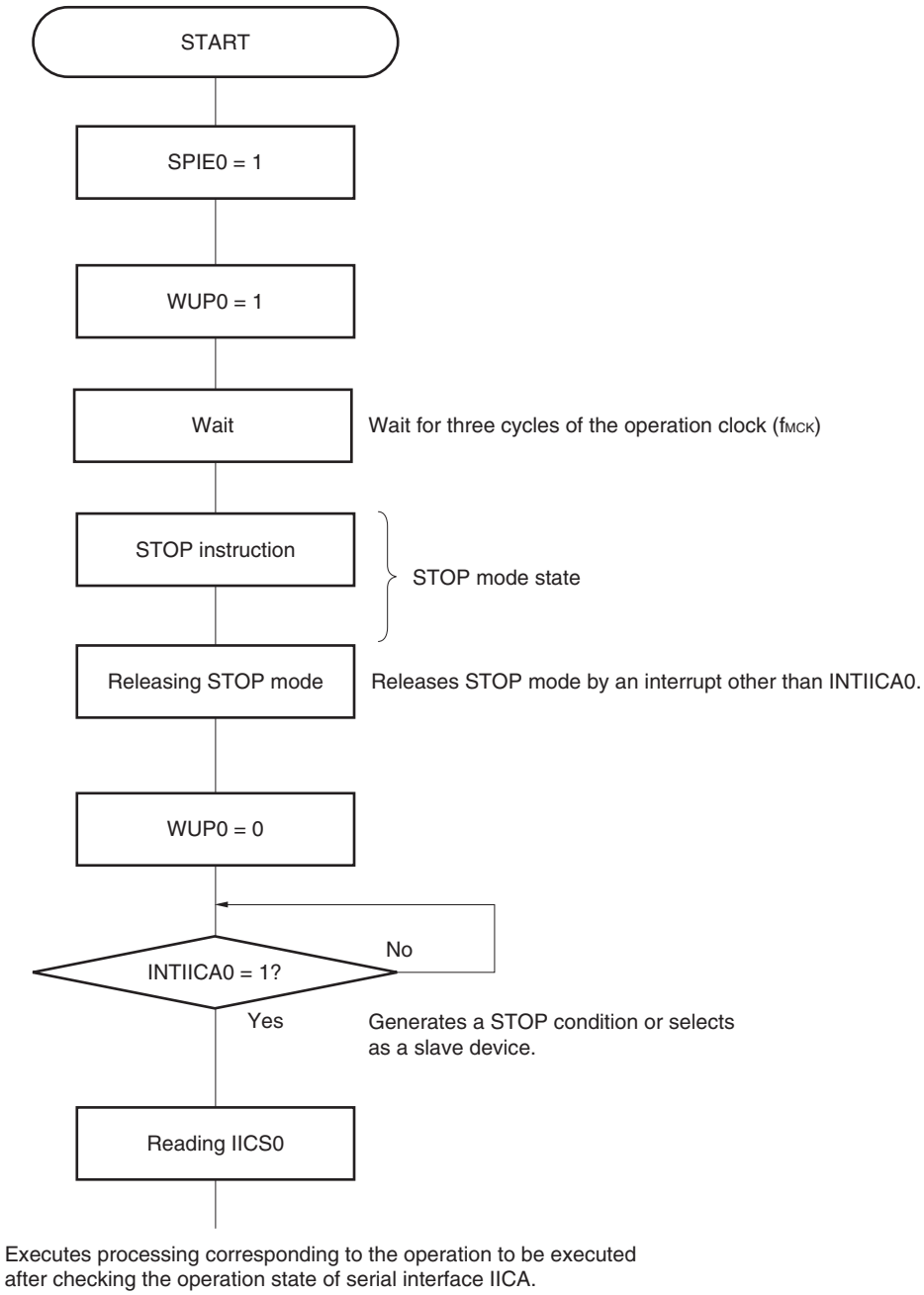



Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICA0) generated from serial interface IICA.

- Master device operation: Flow shown in Figure 16-25
- Slave device operation: Same as the flow in Figure 16-24. The value of WUP0 must be kept 1 until the INTIICA0 is set to 1.

**Figure 16-25. When Operating as Master Device after Releasing STOP Mode other than by INTIICA0**



### 16.5.14 Communication reservation

**(1) When communication reservation function is enabled (bit 0 (IICRSV0) of IICA flag register 0 (IICF0) = 0)**

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL0) of IICA control register 00 (IICCTL00) to 1 and saving communication).

If bit 1 (STT0) of the IICCTL00 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.
If an address is written to the IICA shift register 0 (IICA0) after bit 4 (SPIE0) of the IICCTL00 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICA0) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICA0 register before the stop condition is detected is invalid.
When the STT0 bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released ..... a start condition is generated
- If the bus has not been released (standby mode) ..... communication reservation

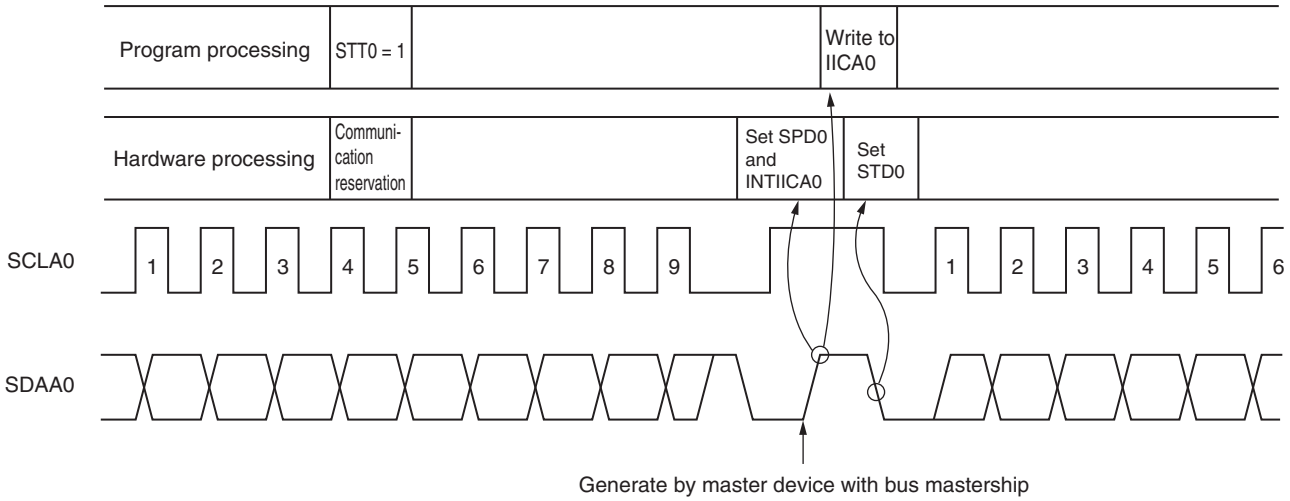
Check whether the communication reservation operates or not by using the MSTS0 bit (bit 7 of the IICA status register 0 (IICS0)) after the STT0 bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

| Wait time from setting STT0 = 1 to checking the MSTS0 flag: |
|---|
| (IICWL0 setting value + IICWH0 setting value + 4) / $f_{MCK}$ + $t_F \times 2$ |

**Remark** IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
$t_F$: SDAA0 and SCLA0 signal falling times
$f_{MCK}$: Frequency of the IICA operation clock

Figure 16-26 shows the communication reservation timing.

**Figure 16-26. Communication Reservation Timing**




**Remark** IICA0: IICA shift register 0
STT0: Bit 1 of IICA control register 00 (IICCTL00)
STD0: Bit 1 of IICA status register 0 (IICS0)
SPD0: Bit 0 of IICA status register 0 (IICS0)

Communication reservations are accepted via the timing shown in Figure 16-27. After bit 1 (STD0) of the IICA status register 0 (IICS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IICA control register 00 (IICCTL00) to 1 before a stop condition is detected.

**Figure 16-27. Timing for Accepting Communication Reservations**

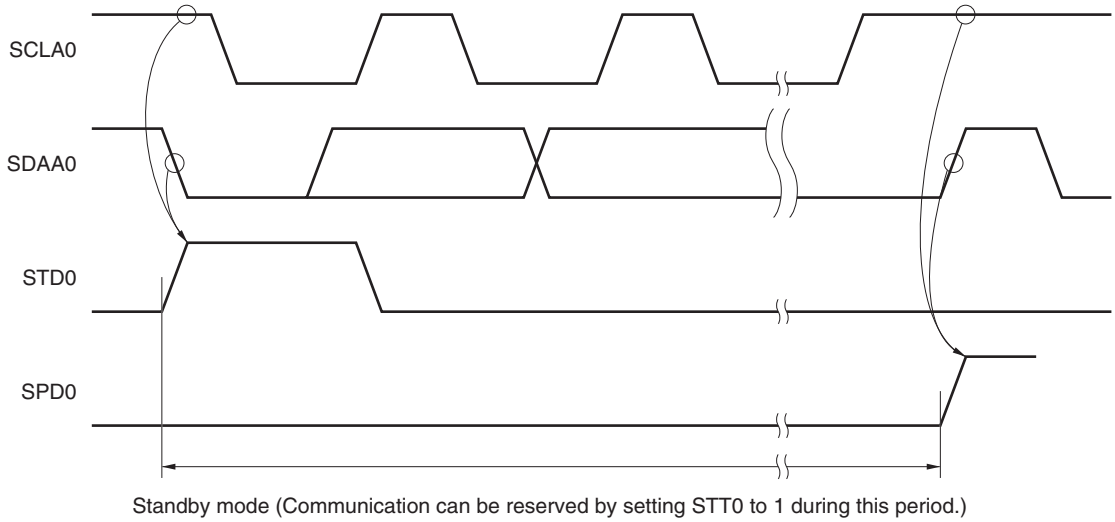
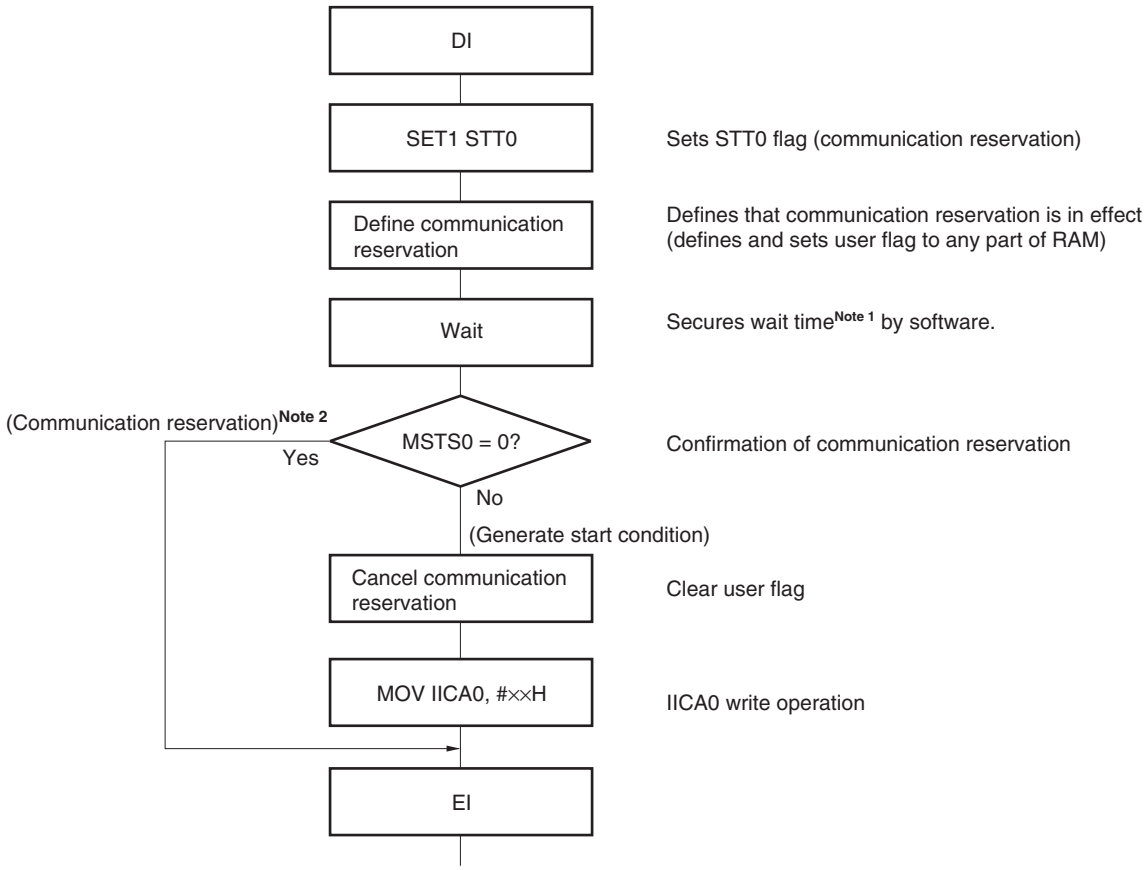



Figure 16-28 shows the communication reservation protocol.

**Figure 16-28. Communication Reservation Protocol**




**Notes 1.** The wait time is calculated as follows.
(IICWL0 setting value + IICWH0 setting value + 4) / $f_{MCK}$ + $t_F \times 2$

**2.** The communication reservation operation executes a write to the IICA shift register 0 (IICA0) when a stop condition interrupt request occurs.

**Remark** STT0: Bit 1 of IICA control register 00 (IICCTL00)
MSTS0: Bit 7 of IICA status register 0 (IICS0)
IICA0: IICA shift register 0
IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
$t_F$: SDAA0 and SCLA0 signal falling times
$f_{MCK}$: Frequency of the IICA operation clock

**(2) When communication reservation function is disabled (bit 0 (IICRSV0) of IICA flag register 0 (IICF0) = 1)**

When bit 1 (STT0) of IICA control register 00 (IICCTL00) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL0) of the IICCTL00 register to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCF0 (bit 7 of the IICF0 register). It takes up to 5 cycles of the operation clock ($f_{MCK}$) until the STCF0 bit is set to 1 after setting STT0 = 1. Therefore, secure the time by software.

### 16.5.15 Cautions

(1) When STCEN0 = 0

Immediately after $I^2C$ operation is enabled (IICE0 = 1), the bus communication status (IICBSY0 = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

<1> Set IICA control register 01 (IICCTL01).
<2> Set bit 7 (IICE0) of IICA control register 00 (IICCTL00) to 1.
<3> Set bit 0 (SPT0) of the IICCTL00 register to 1.

(2) When STCEN0 = 1

Immediately after $I^2C$ operation is enabled (IICE0 = 1), the bus released status (IICBSY0 = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT0 = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other $I^2C$ communications are already in progress

If $I^2C$ operation is enabled and the device participates in communication already in progress when the SDAA0 pin is low and the SCLA0 pin is high, the macro of $I^2C$ recognizes that the SDAA0 pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, $\overline{ACK}$ is returned, but this interferes with other $I^2C$ communications. To avoid this, start $I^2C$ in the following sequence.

<1> Clear bit 4 (SPIE0) of the IICCTL00 register to 0 to disable generation of an interrupt request signal (INTIICA0) when the stop condition is detected.
<2> Set bit 7 (IICE0) of the IICCTL00 register to 1 to enable the operation of $I^2C$.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL0) of the IICCTL00 register to 1 before $\overline{ACK}$ is returned (4 to 80 cycles of the operation clock ($f_{MCK}$) after setting the IICE0 bit to 1), to forcibly disable detection.

(4) Setting the STT0 and SPT0 bits (bits 1 and 0 of the IICCTL00 register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set the SPIE0 bit (bit 4 of the IICTL0 register) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to the IICA shift register 0 (IICA0) after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set the SPIE0 bit to 1 when the MSTS0 bit (bit 7 of the IICA status register 0 (IICS0)) is detected by software.

### 16.5.16 Communication operations

The following shows three operation procedures with the flowchart.

#### (1) Master operation in single master system

The flowchart when using the RL78/F13 and RL78/F14 as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

#### (2) Master operation in multimaster system

In the $I^2C$ bus multimaster system, whether the bus is released or used cannot be judged by the $I^2C$ bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the RL78/F13 and RL78/F14 take part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the RL78/F13 and RL78/F14 lose in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

#### (3) Slave operation

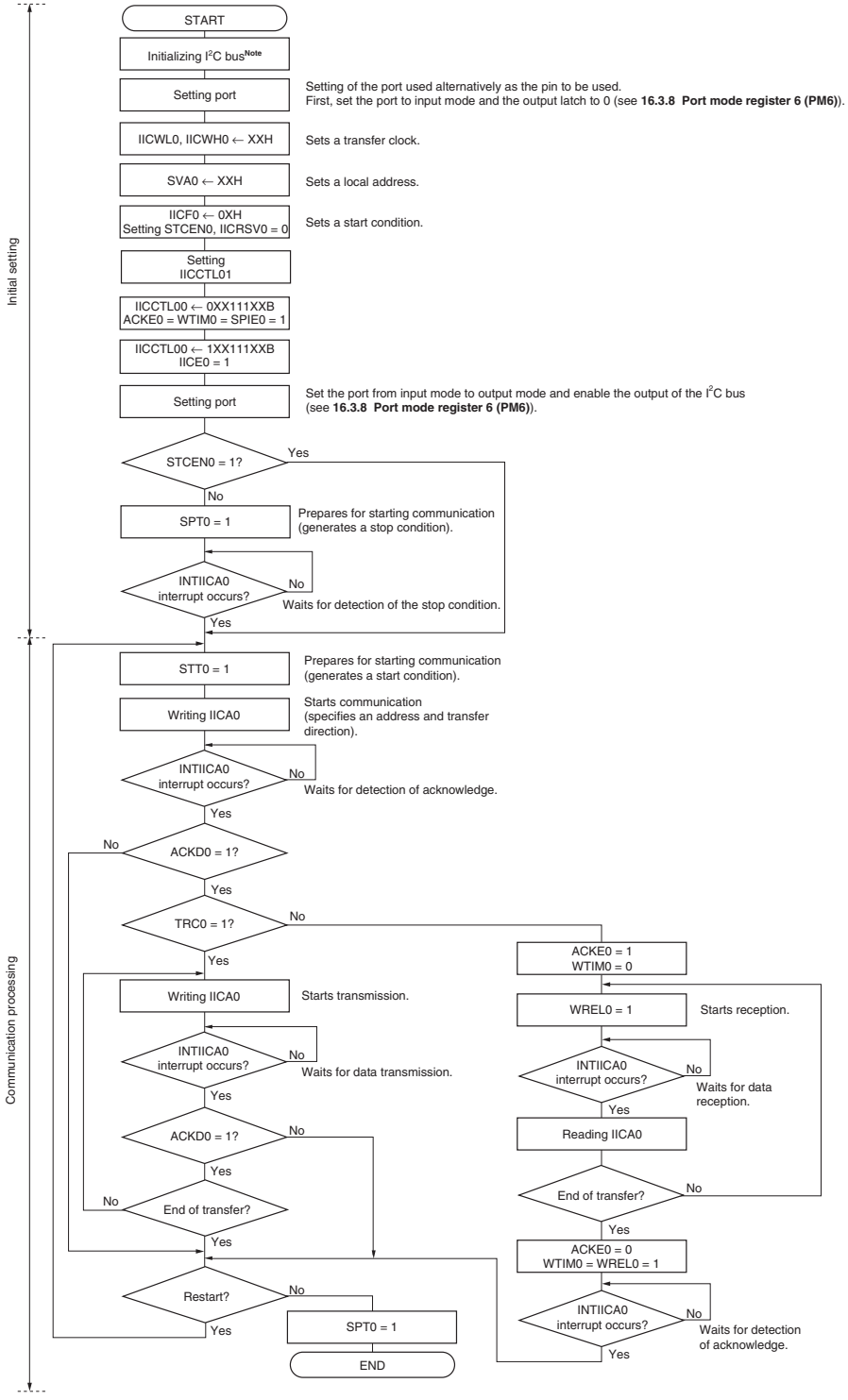
An example of when the RL78/F13 and RL78/F14 are used as the $I^2C$ bus slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICA0 interrupt occurrence (communication waiting). When an INTIICA0 interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

### (1) Master operation in single-master system

**Figure 16-29. Master Operation in Single-Master System**

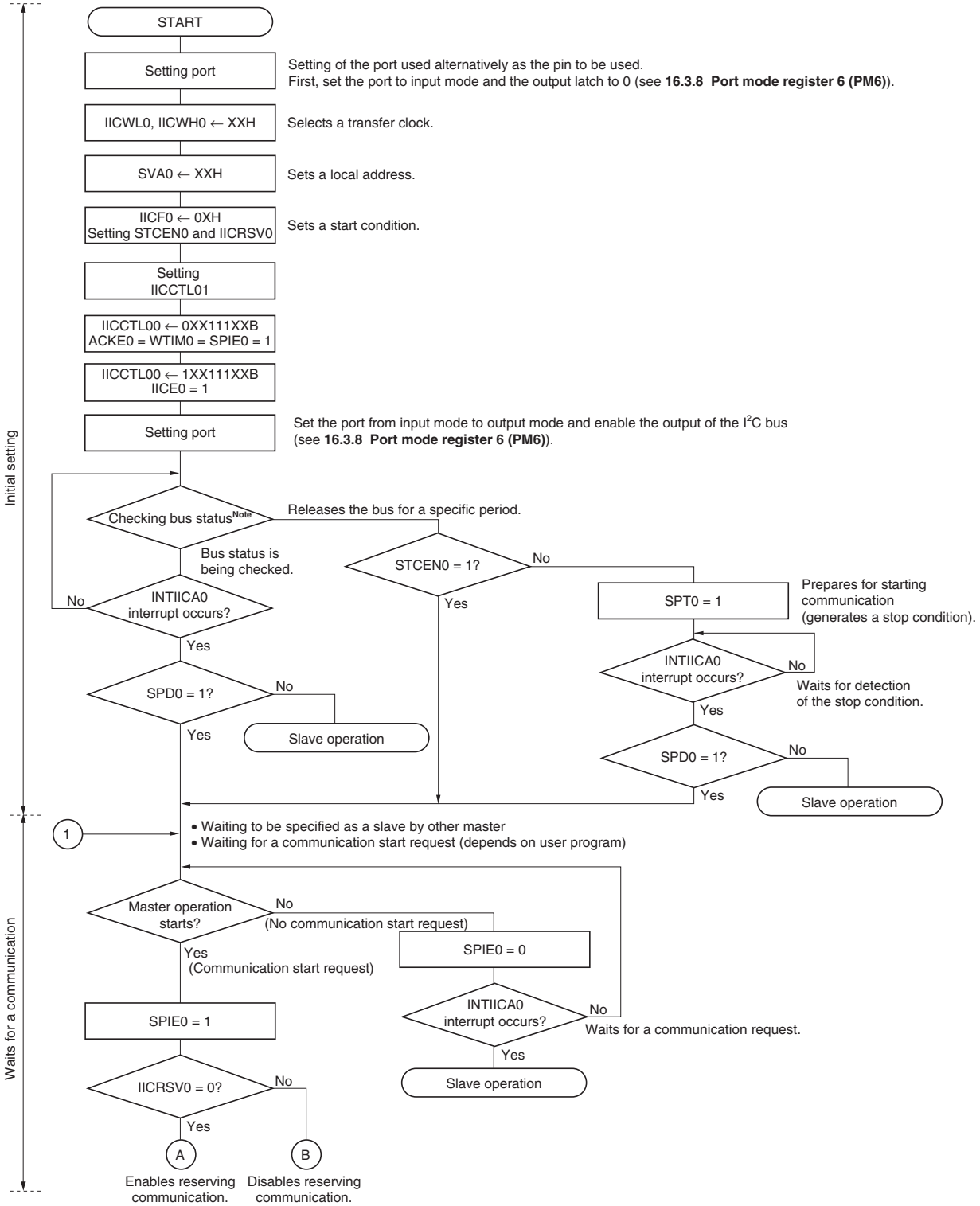



**Note** Release (SCLA0 and SDAA0 pins = high level) the I²C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAA0 pin, for example, set the SCLA0 pin in the output port mode, and output a clock pulse from the output port until the SDAA0 pin is constantly at high level.

**Remark** Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

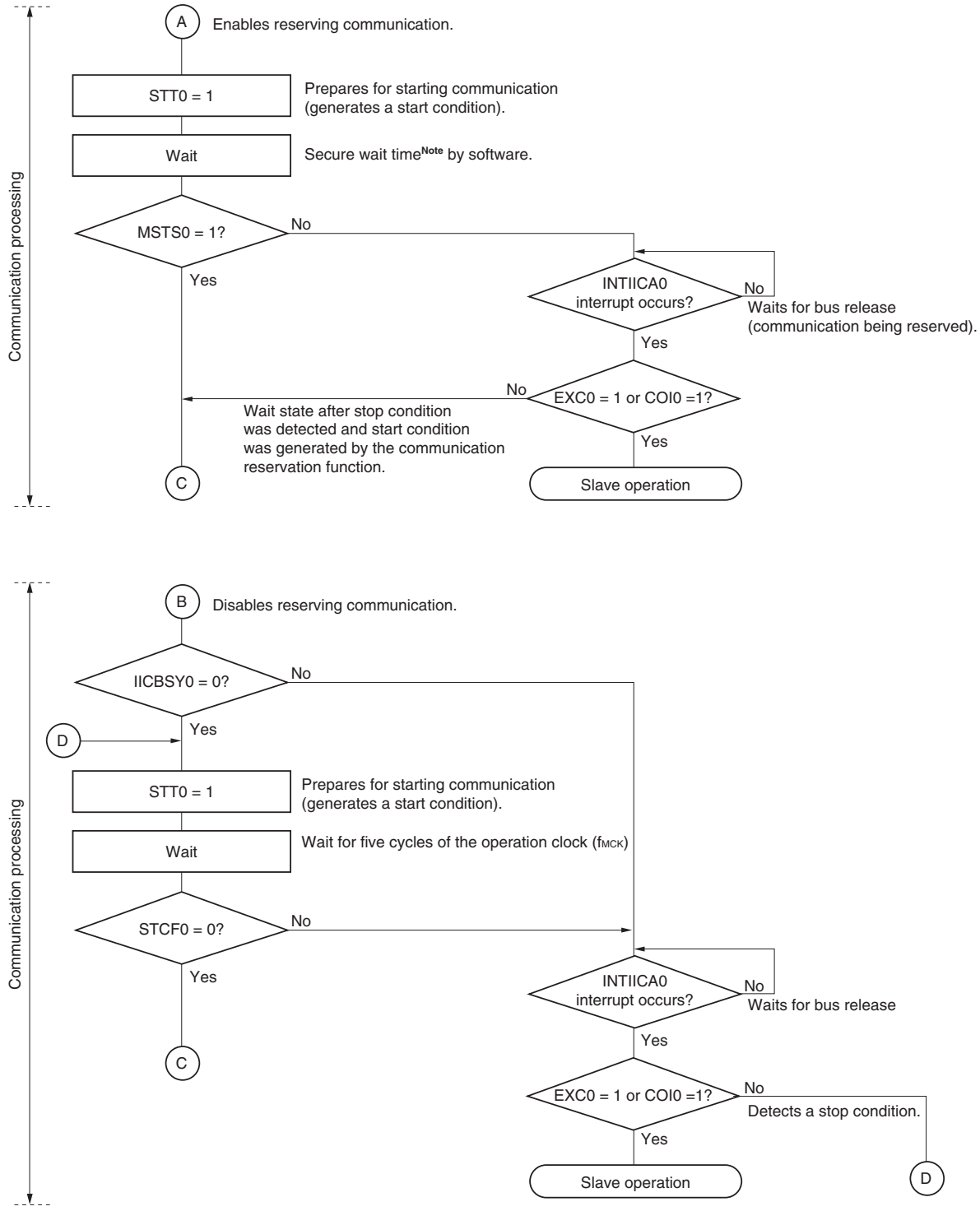
### (2) Master operation in multi-master system

**Figure 16-30. Master Operation in Multi-Master System (1/3)**




**Note** Confirm that the bus is released (CLD0 bit = 1, DAD0 bit = 1) for a specific period (for example, for a period of one frame). If the SDAA0 pin is constantly at low level, decide whether to release the I²C bus (SCLA0 and SDAA0 pins = high level) in conformance with the specifications of the product that is communicating.

**Figure 16-30. Master Operation in Multi-Master System (2/3)**




**Note** The wait time is calculated as follows.

(IICWL0 setting value + IICWH0 setting value + 4) / $f_{MCK}$ + $t_F \times 2$

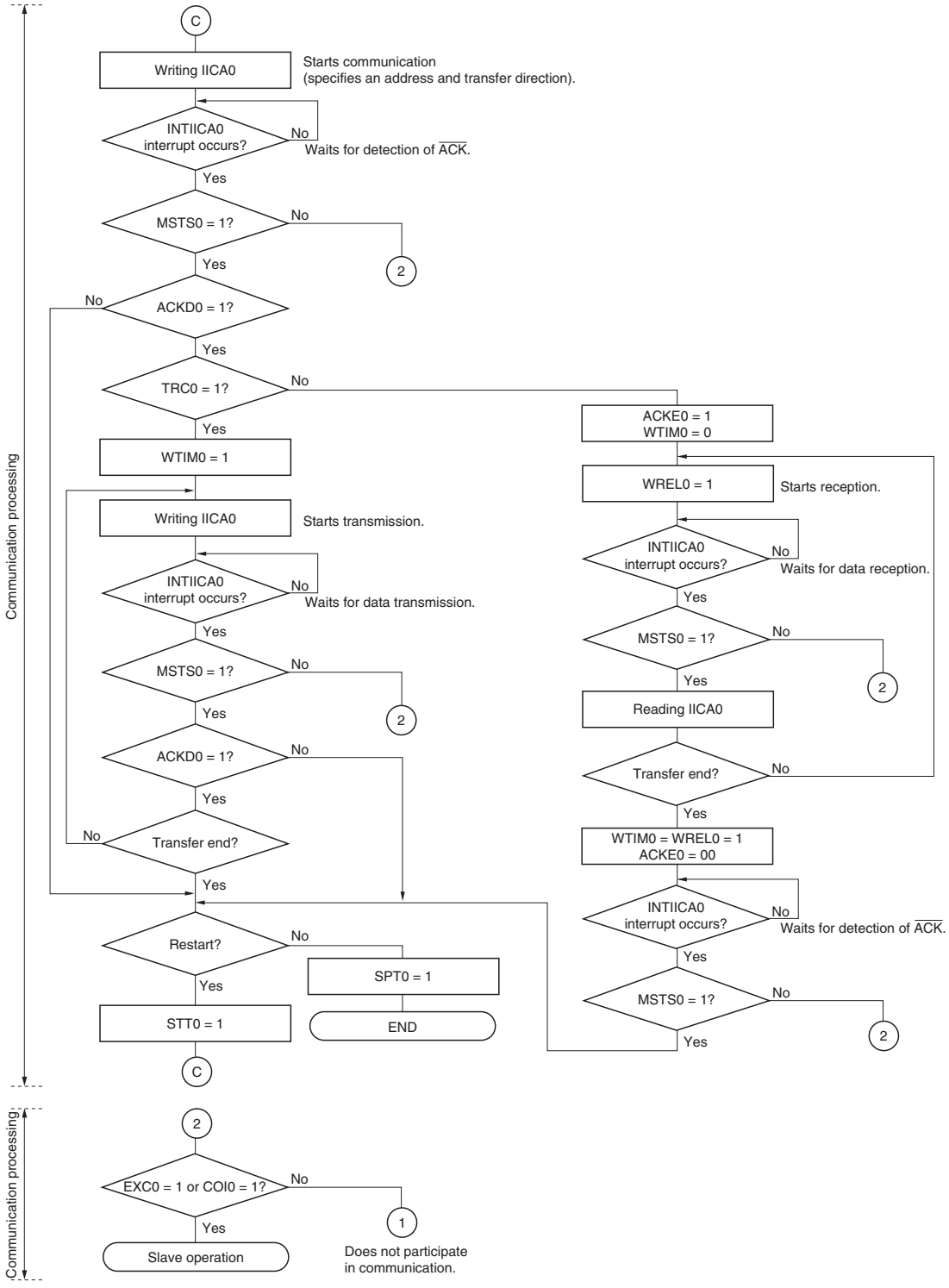
**Remark** IICWL0: IICA low-level width setting register 0

IICWH0: IICA high-level width setting register 0

$t_F$: SDAA0 and SCLA0 signal falling times

$f_{MCK}$: Frequency of the IICA operation clock

**Figure 16-30. Master Operation in Multi-Master System (3/3)**




**Remarks 1.** Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

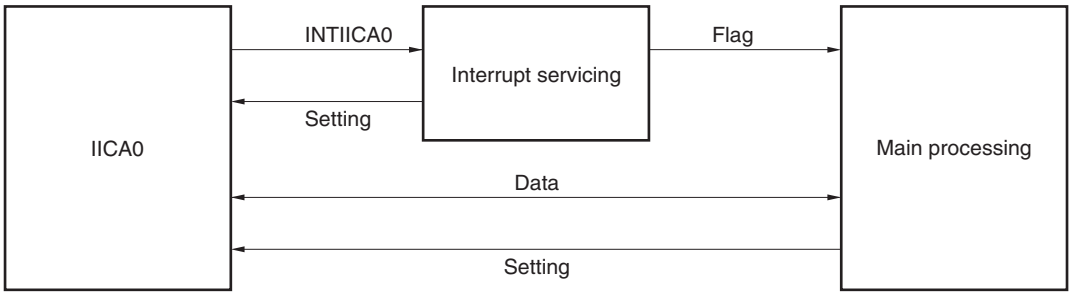
**2.** To use the device as a master in a multi-master system, read the MSTS0 bit each time interrupt INTIICA0 has occurred to check the arbitration result.

**3.** To use the device as a slave in a multi-master system, check the status by using the IICA status register 0 (IICS0) and IICA flag register 0 (IICF0) each time interrupt INTIICA0 has occurred, and determine the processing to be performed next.

### (3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICA0 interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary. In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIICA0 interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.




Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICA0.

**<1> Communication mode flag**

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of $\overline{\text{ACK}}$ from master, address mismatch)

**<2> Ready flag**

This flag indicates that data communication is enabled. Its function is the same as the INTIICA0 interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

**<3> Communication direction flag**

This flag indicates the direction of communication. Its value is the same as the TRC0 bit.

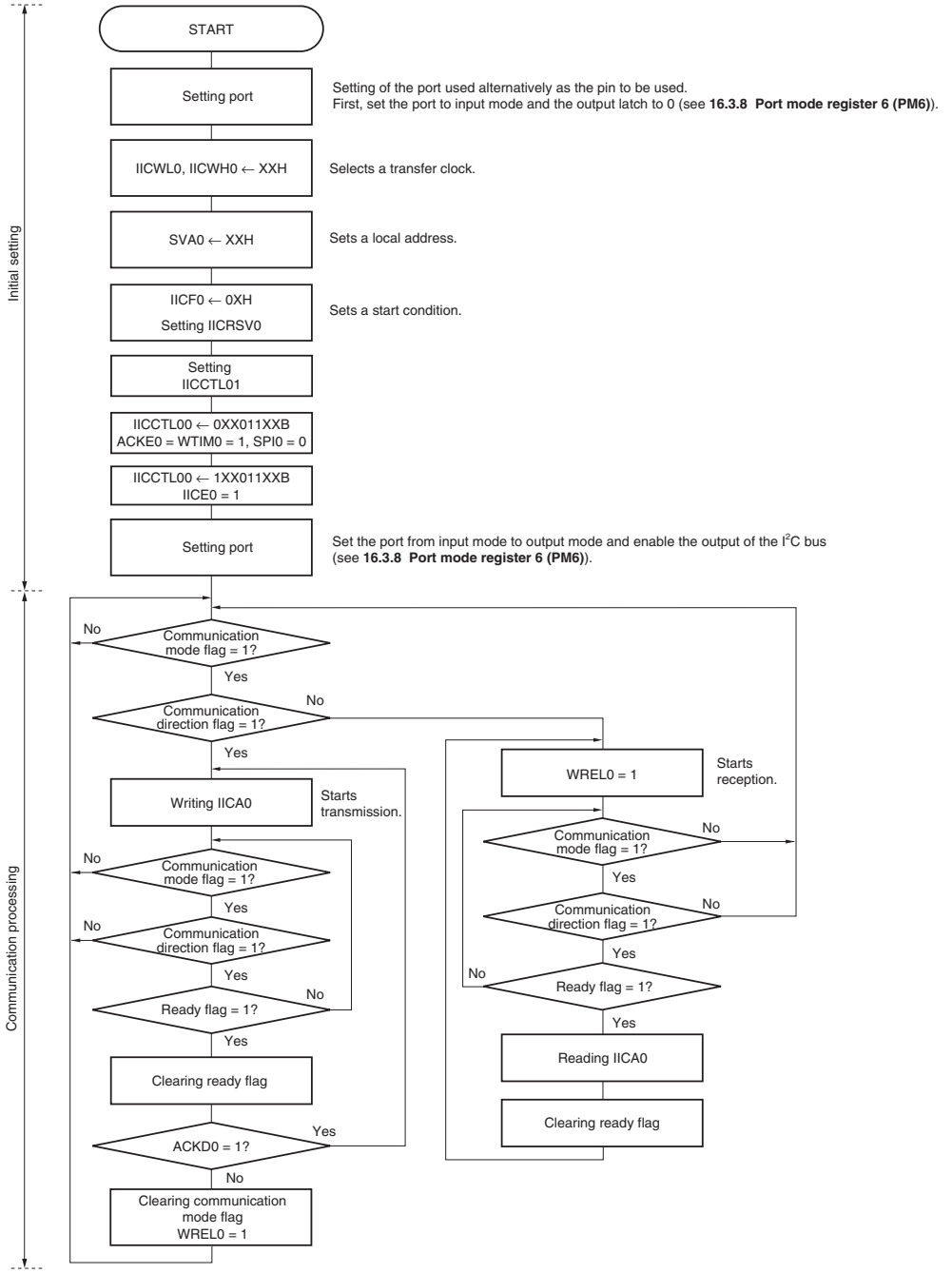
The main processing of the slave operation is explained next.

Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns $\overline{ACK}$. If $\overline{ACK}$ is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, $\overline{ACK}$ is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

**Figure 16-31. Slave Operation Flowchart (1)**



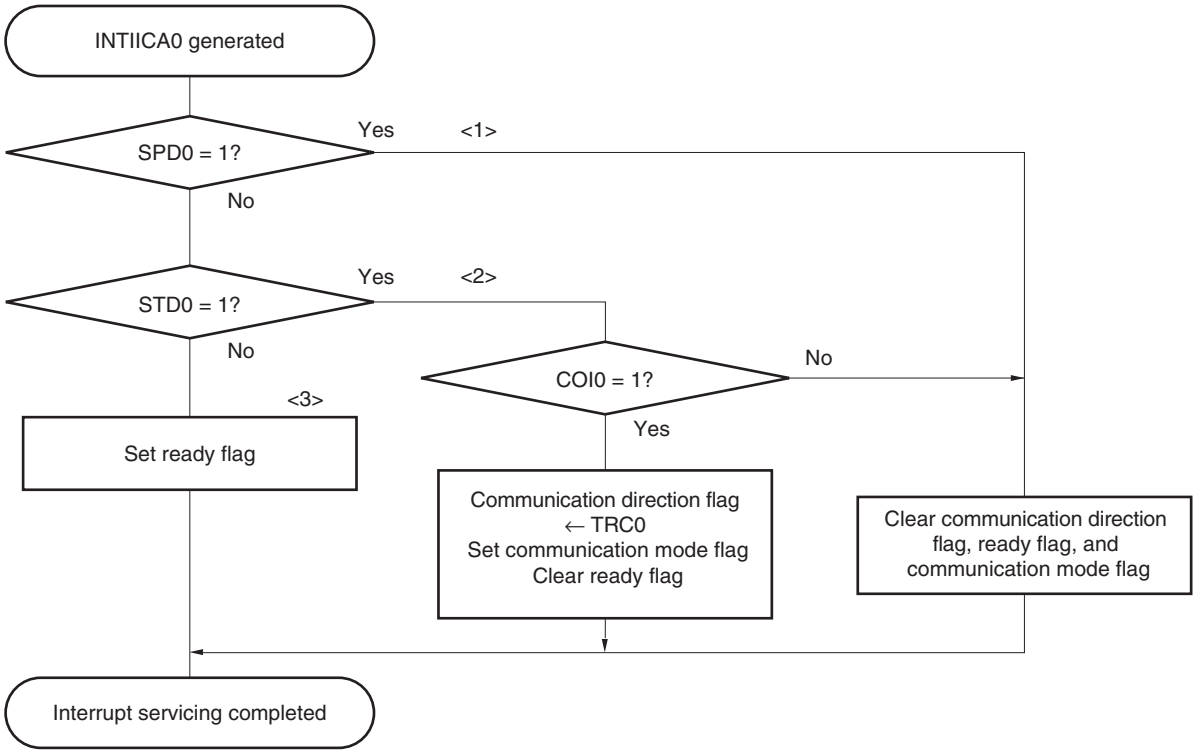

**Remark** Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

An example of the processing procedure of the slave with the INTIICA0 interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICA0 interrupt checks the status, and the following operations are performed.

<1> Communication is stopped if the stop condition is issued.
<2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).
<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the $I^2C$ bus remaining in the wait state.

**Remark** <1> to <3> above correspond to <1> to <3> in Figure 16-31 Slave Operation Flowchart (2).

**Figure 16-31. Slave Operation Flowchart (2)**




### 16.5.17 Timing of $I^2C$ interrupt request (INTIICA0) occurrence

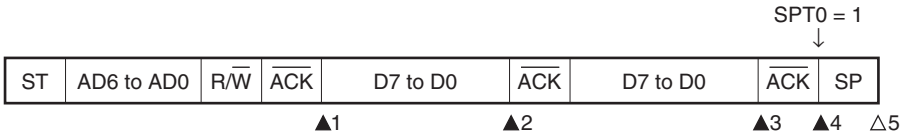
The timing of transmitting or receiving data and generation of interrupt request signal INTIICA0, and the value of the IICA status register 0 (IICS0) when the INTIICA0 signal is generated are shown below.

**Remark** ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

### (1) Master device operation

#### (a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

##### (i) When WTIM0 = 0

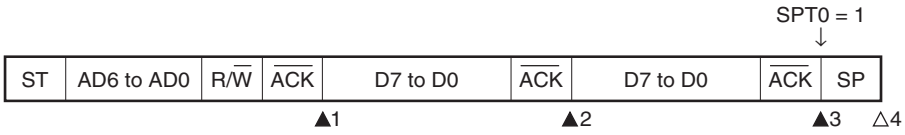



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B
▲3: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)Note
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)Note
△5: IICS0 = 00000001B

**Note** To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

##### (ii) When WTIM0 = 1

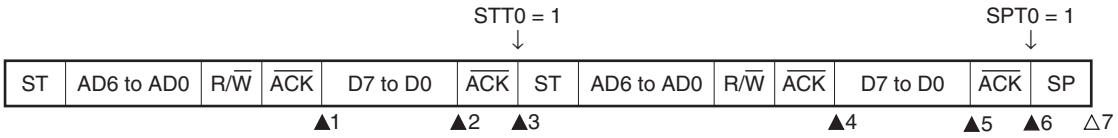



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B
▲3: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)**

**(i) When WTIM0 = 0**

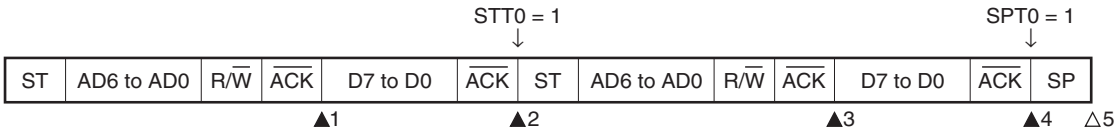



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)Note 1
▲3: IICS0 = 1000××00B (Clears the WTIM0 bit to 0Note 2, sets the STT0 bit to 1)
▲4: IICS0 = 1000×110B
▲5: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)Note 3
▲6: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△7: IICS0 = 00000001B

**Notes 1.** To generate a start condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.
**2.** Clear the WTIM0 bit to 0 to restore the original setting.
**3.** To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1**

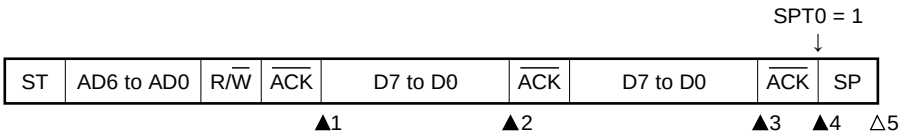



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
▲3: IICS0 = 1000×110B
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)**

**(i) When WTIM0 = 0**

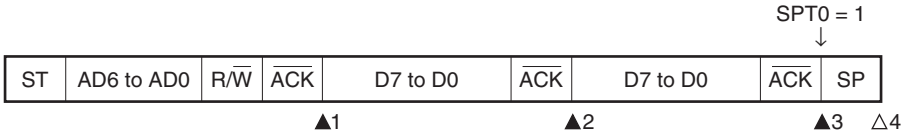



▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×000B
▲3: IICS0 = 1010×000B (Sets the WTIM0 bit to 1)**Note**
▲4: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

**Note** To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1**



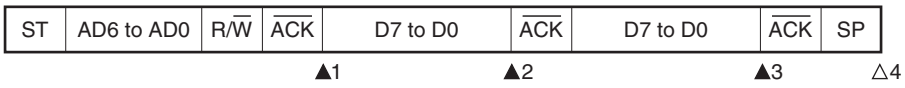

▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×100B
▲3: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00001001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(2) Slave device operation (slave address data reception)**

**(a) Start ~ Address ~ Data ~ Data ~ Stop**

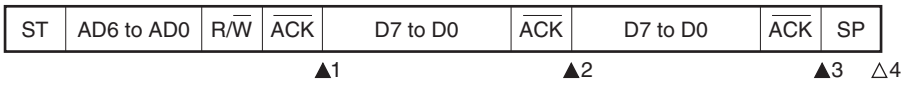
**(i) When WTIM0 = 0**




▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1**




▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop**

**(i) When WTIM0 = 0 (after restart, matches with SVA0)**

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1 (after restart, matches with SVA0)**

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001××00B
△5: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

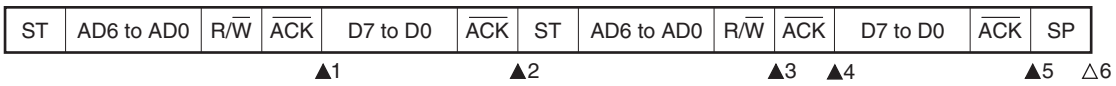
**(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop**

**(i) When WTIM0 = 0 (after restart, does not match address (= extension code))**

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1 (after restart, does not match address (= extension code))**




▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×110B
▲5: IICS0 = 0010××00B
△6: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop**

**(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))**

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))**

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

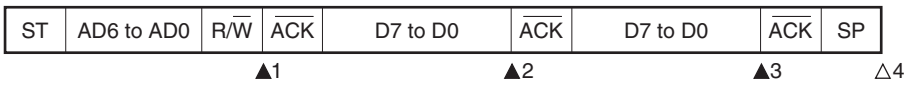
**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(3) Slave device operation (when receiving extension code)**

The device is always participating in communication when it receives an extension code.

**(a) Start ~ Code ~ Data ~ Data ~ Stop**

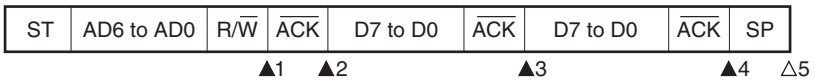
**(i) When WTIM0 = 0**




▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1**




▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop**

**(i) When WTIM0 = 0 (after restart, matches SVA0)**

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1 (after restart, matches SVA0)**

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0001×110B
▲5: IICS0 = 0001××00B
△6: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop**

**(i) When WTIM0 = 0 (after restart, extension code reception)**

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1 (after restart, extension code reception)**

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0010×010B
▲5: IICS0 = 0010×110B
▲6: IICS0 = 0010××00B
△7: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop**

**(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))**

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

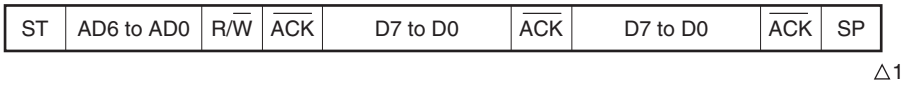
**(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))**

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 00000×10B
△5: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(4) Operation without communication**

**(a) Start ~ Code ~ Data ~ Data ~ Stop**




△1: IICS0 = 00000001B

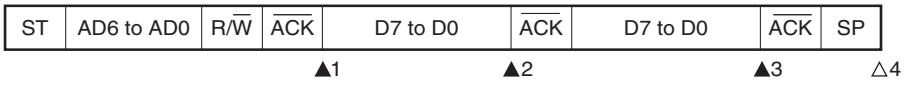
**Remark** △: Generated only when SPIE0 = 1

**(5) Arbitration loss operation (operation as slave after arbitration loss)**

When the device is used as a master in a multi-master system, read the MSTS0 bit each time interrupt request signal INTIICA0 has occurred to check the arbitration result.

**(a) When arbitration loss occurs during transmission of slave address data**

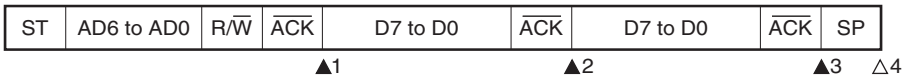
**(i) When WTIM0 = 0**




▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1**

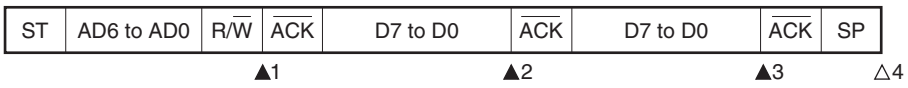



▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(b) When arbitration loss occurs during transmission of extension code**

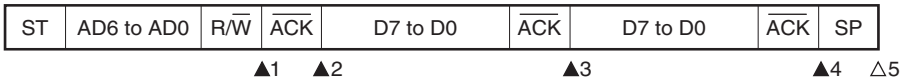
**(i) When WTIM0 = 0**




▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1**




▲1 ▲2 ▲3 ▲4 △5

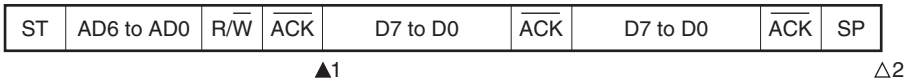
▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(6) Operation when arbitration loss occurs (no communication after arbitration loss)**

When the device is used as a master in a multi-master system, read the MSTS0 bit each time interrupt request signal INTIICA0 has occurred to check the arbitration result.

**(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)**

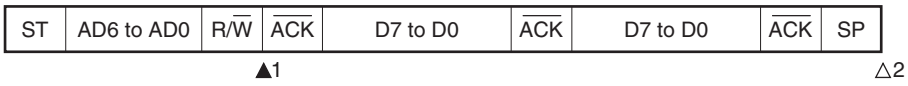



▲1 △2

▲1: IICS0 = 01000110B
△2: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1

**(b) When arbitration loss occurs during transmission of extension code**

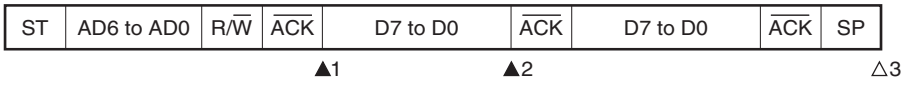



▲1: IICS0 = 0110×010B
Sets LREL0 = 1 by software
△2: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(c) When arbitration loss occurs during transmission of data**

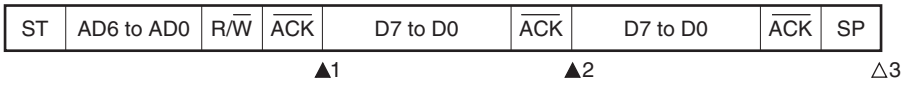
**(i) When WTIM0 = 0**




▲1: IICS0 = 10001110B
▲2: IICS0 = 01000000B
△3: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1

**(ii) When WTIM0 = 1**

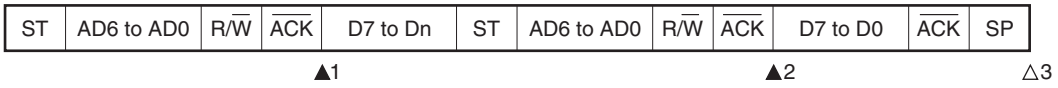



▲1: IICS0 = 10001110B
▲2: IICS0 = 01000100B
△3: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1

**(d) When loss occurs due to restart condition during data transfer**

**(i) Not extension code (Example: unmatches with SVA0)**




▲1: IICS0 = 1000×110B
▲2: IICS0 = 01000110B
△3: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

**(ii) Extension code**

▲1: IICS0 = 1000×110B
▲2: IICS0 = 01100010B
Sets LREL0 = 1 by software
△3: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

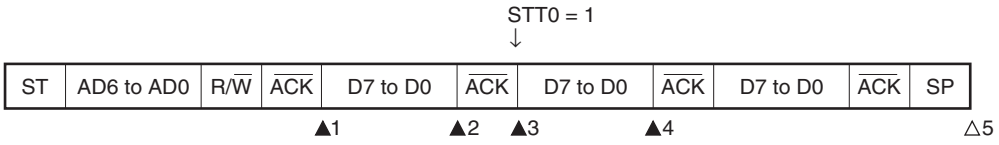
**(e) When loss occurs due to stop condition during data transfer**

▲1: IICS0 = 10000110B
△2: IICS0 = 01000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

**(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition**

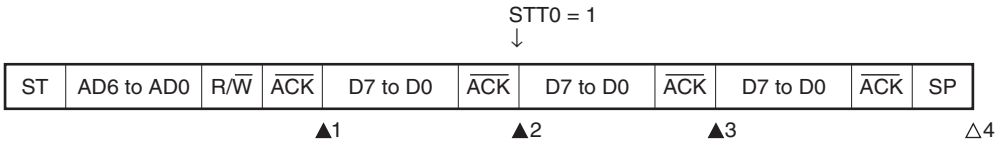
**(i) When WTIM0 = 0**




▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000×100B (Clears the WTIM0 bit to 0)
▲4: IICS0 = 01000000B
△5: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1**

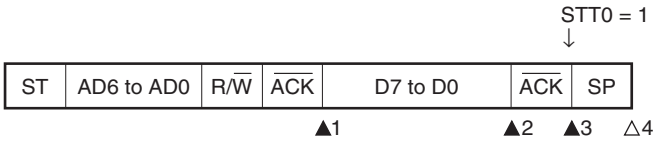



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets the STT0 bit to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition**

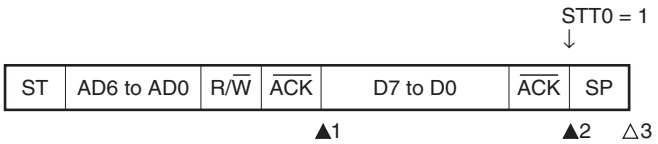
**(i) When WTIM0 = 0**




▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000××00B (Sets the STT0 bit to 1)
△4: IICS0 = 01000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1**

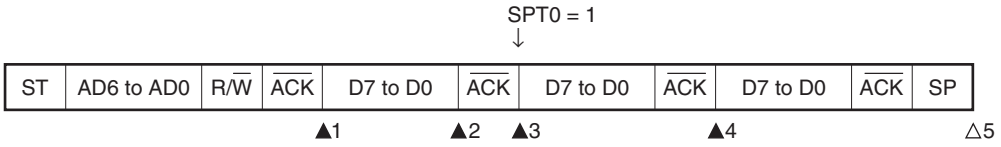



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
△3: IICS0 = 01000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition**

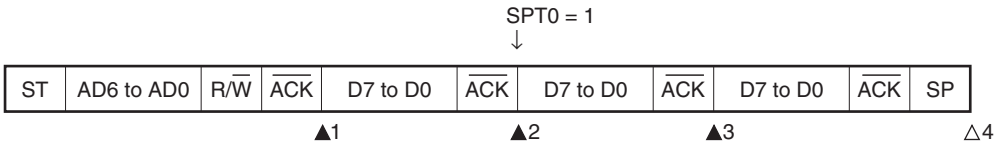
**(i) When WTIM0 = 0**




▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000×100B (Clears the WTIM0 bit to 0)
▲4: IICS0 = 01000100B
△5: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

**(ii) When WTIM0 = 1**




▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets the SPT0 bit to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

**Remark** ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

## 16.6 Timing Charts

When using the $I^2C$ bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC0 bit (bit 3 of the IICA status register 0 (IICS0)), which specifies the data transfer direction, and then starts serial communication with the slave device.

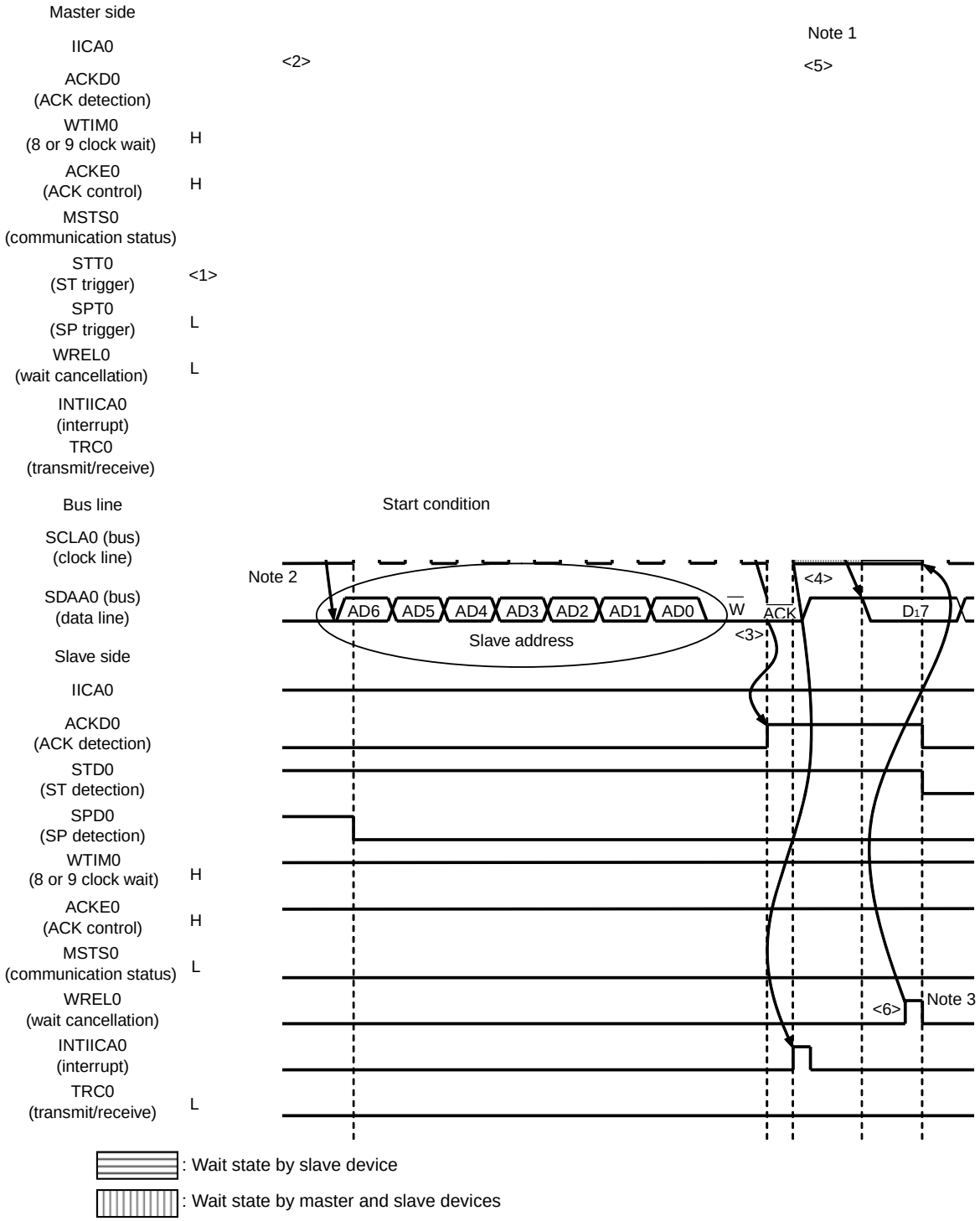
Figures 16-32 and 16-33 show timing charts of the data communication.

The IICA shift register 0 (IICA0)'s shift operation is synchronized with the falling edge of the serial clock (SCLA0). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAA0 pin.

Data input via the SDAA0 pin is captured into IICA0 at the rising edge of SCLA0.

**Figure 16-32. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/4)**

**(1) Start condition ~ address ~ data**




**Notes 1.** Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission by a master device.

**2.** Make sure that the time between the fall of the SDAA0 pin signal and the fall of the SCLA0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

**3.** For releasing wait state during reception of a slave device, write “FFH” to IICA0 or set the WREL0 bit.

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 16-32 are explained below.

<1> The start condition trigger is set by the master device (STT0 = 1) and a start condition (i.e. SCLA0 = 1 changes SDAA0 from 1 to 0) is generated once the bus data line goes low (SDAA0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS0 = 1). The master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register 0 (IICA0) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVA0 value) of a slave device**Note**, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match)**Note**.

<5> The master device writes the data to transmit to the IICA0 register and releases the wait status that it set by the master device.

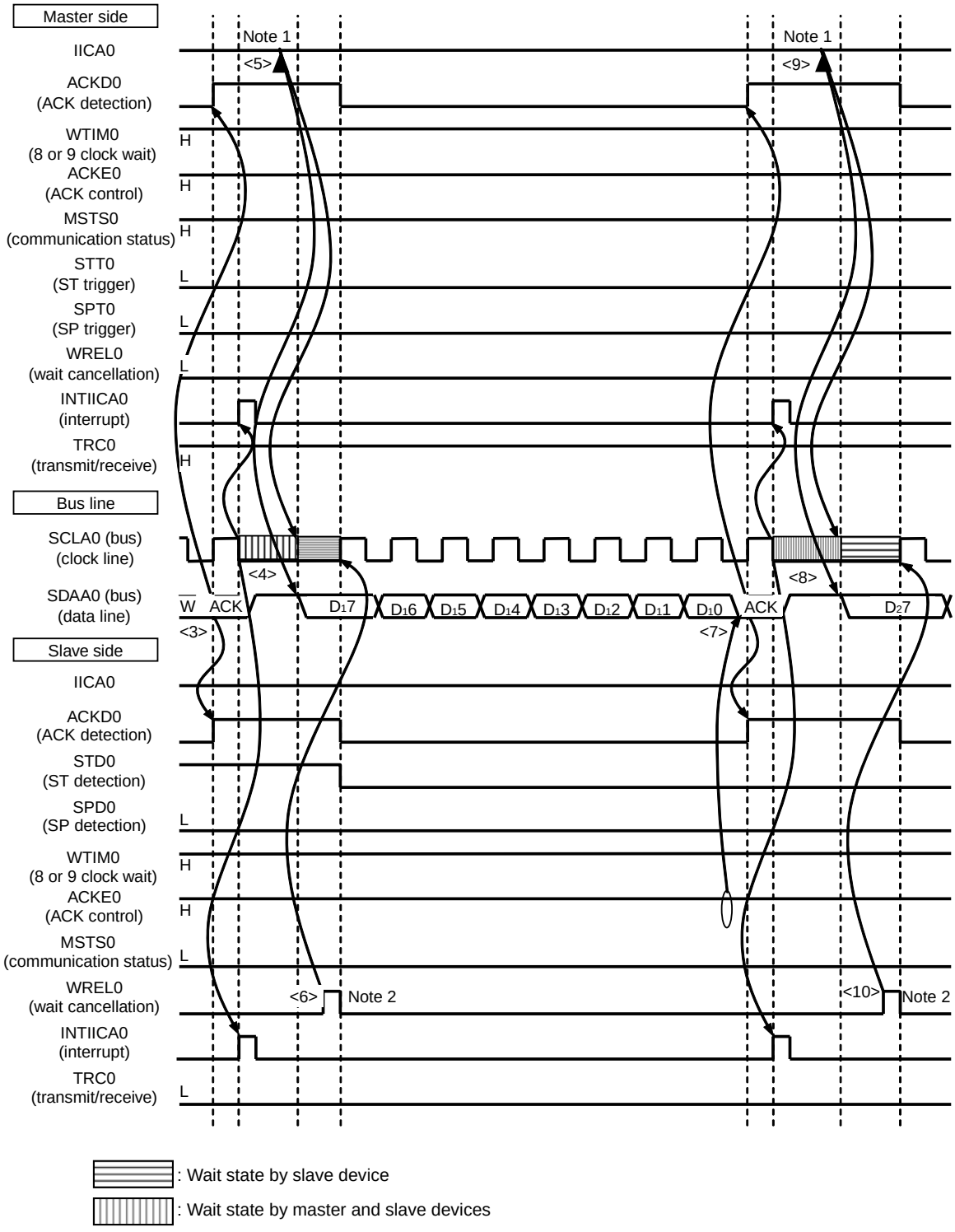
<6> If the slave device releases the wait status (WREL0 = 1), the master device starts transferring data to the slave device.

**Note** If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

**Remark** <1> to <15> in Figure 16-32 represent the entire procedure for communicating data using the $I^2C$ bus. Figure 16-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 16-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 16-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

**Figure 16-32. Example of Master to Slave Communication**
**(When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/4)**

**(2) Address ~ data ~ data**




**Notes 1.** Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission by a master device.

**2.** For releasing wait state during reception of a slave device, write “FFH” to IICA0 or set the WREL0 bit.

The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 16-32 are explained below.

<3> In the slave device if the address received matches the address (SVA0 value) of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match)[Note].

<5> The master device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the wait status that it set by the master device.

<6> If the slave device releases the wait status (WREL0 = 1), the master device starts transferring data to the slave device.

<7> After data transfer is completed, because of ACKE0 = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<9> The master device writes the data to transmit to the IICA0 register and releases the wait status that it set by the master device.

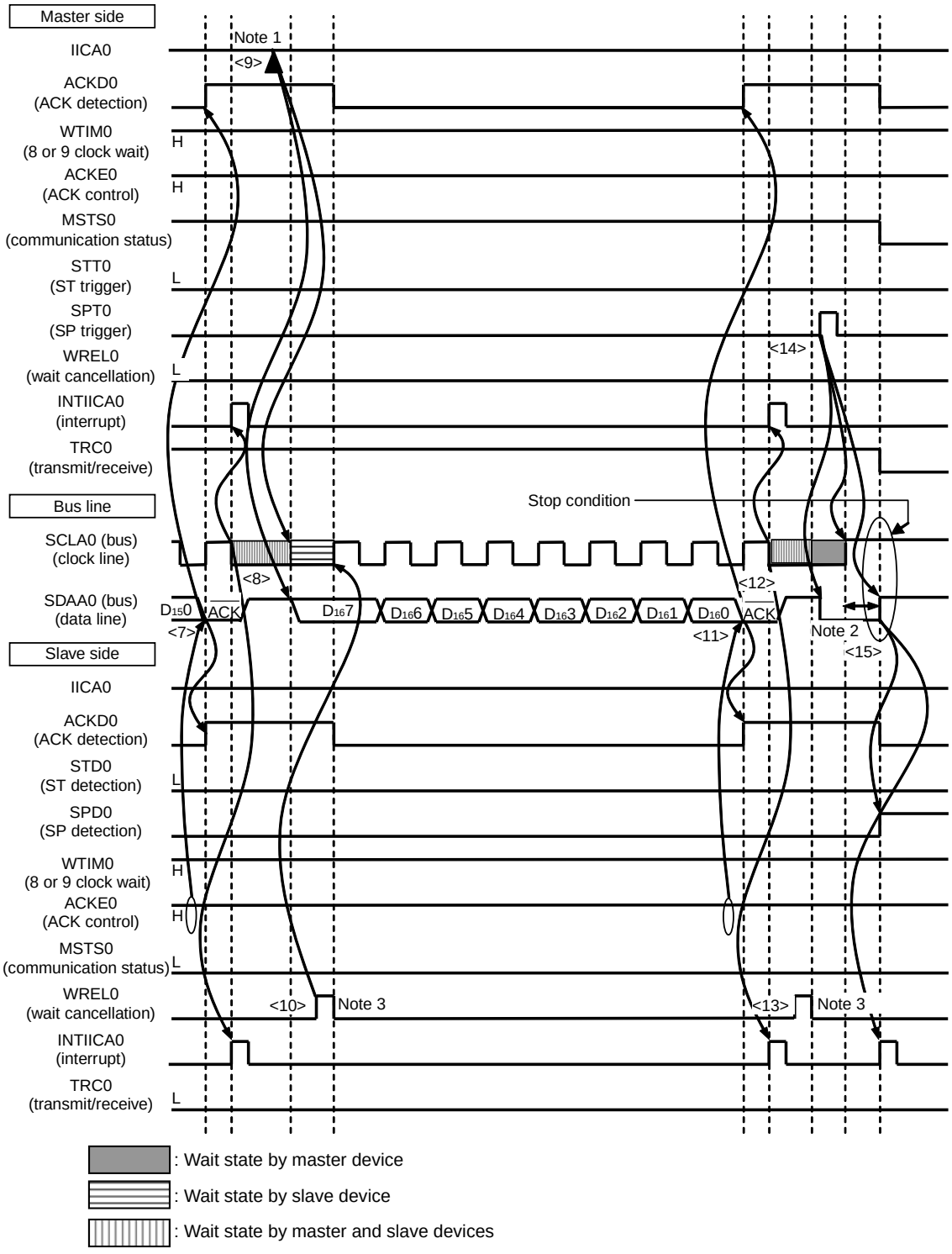
<10> The slave device reads the received data and releases the wait status (WREL0 = 1). The master device then starts transferring data to the slave device.

**Note** If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

**Remark** <1> to <15> in Figure 16-32 represent the entire procedure for communicating data using the $I^2C$ bus. Figure 16-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 16-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 16-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

**Figure 16-32. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/4)**

**(3) Data ~ data ~ Stop condition**




**Notes 1.** Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission by a master device.

**2.** Make sure that the time between the rise of the SCLA0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

**3.** For releasing wait state during reception of a slave device, write “FFH” to IICA0 or set the WREL0 bit.

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 16-32 are explained below.

<7> After data transfer is completed, because of ACKE0 = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the wait status that it set by the master device.

<10> The slave device reads the received data and releases the wait status (WREL0 = 1). The master device then starts transferring data to the slave device.

<11> When data transfer is complete, the slave device (ACKE0 =1) sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<12> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<13> The slave device reads the received data and releases the wait status (WREL0 = 1).

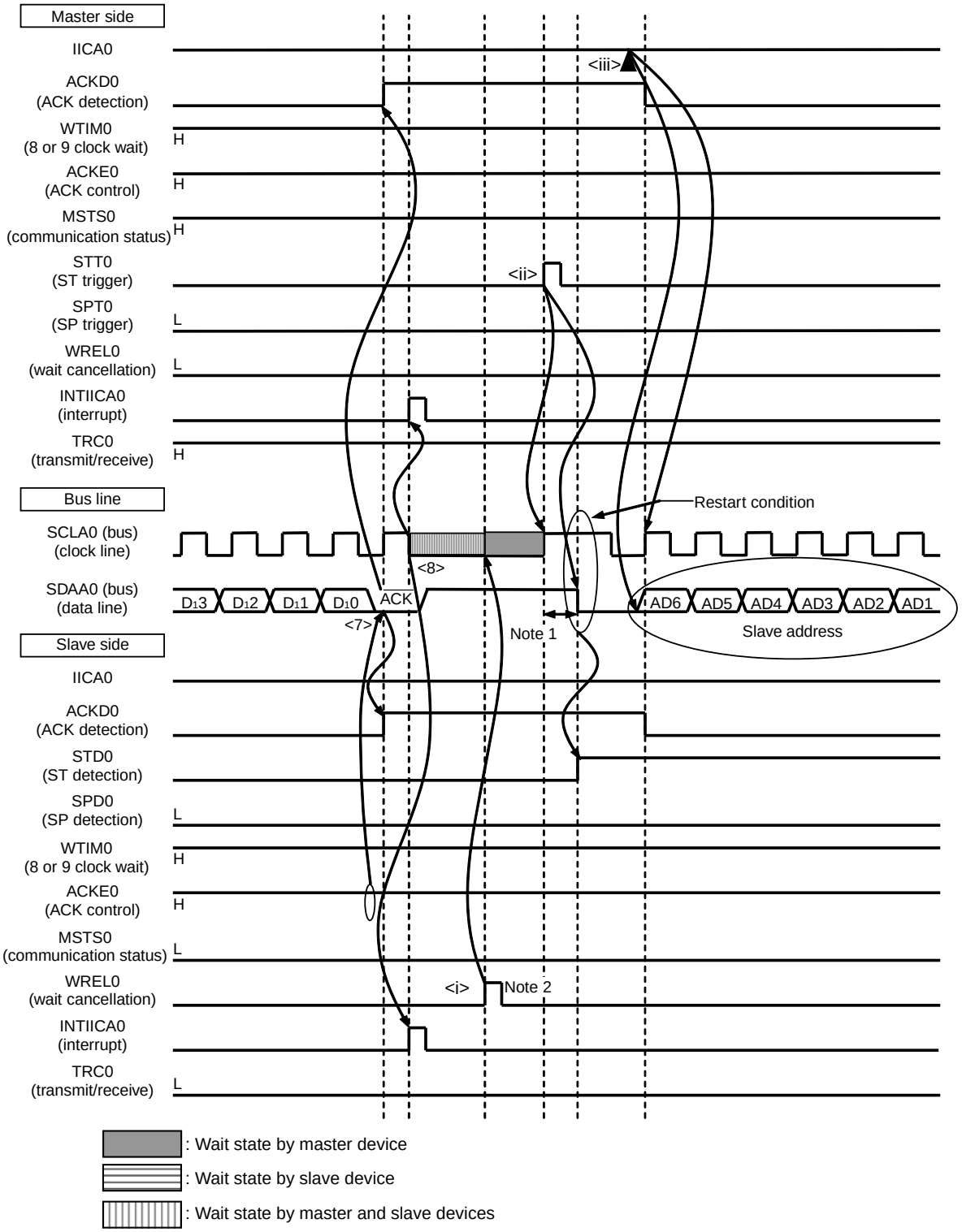
<14> By the master device setting a stop condition trigger (SPT0 = 1), the bus data line is cleared (SDAA0 = 0) and the bus clock line is set (SCLA0 = 1). After the stop condition setup time has elapsed, by setting the bus data line (SDAA0 = 1), the stop condition is then generated (i.e. SCLA0 =1 changes SDAA0 from 0 to 1).

<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICA0: stop condition).

**Remark** <1> to <15> in Figure 16-32 represent the entire procedure for communicating data using the $I^2C$ bus. Figure 16-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 16-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 16-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

**Figure 16-32. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (4/4)**

**(4) Data ~ restart condition ~ address**




**Notes 1.** Make sure that the time between the rise of the SCLA0 pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

**2.** For releasing wait state during reception of a slave device, write “FFH” to IICA0 or set the WREL0 bit.

The following describes the operations in Figure 16-32 (4) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <i> to <iii> are performed. These steps return the processing to step <iii>, the data transmission step.

<7> After data transfer is completed, because of ACKE0 = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

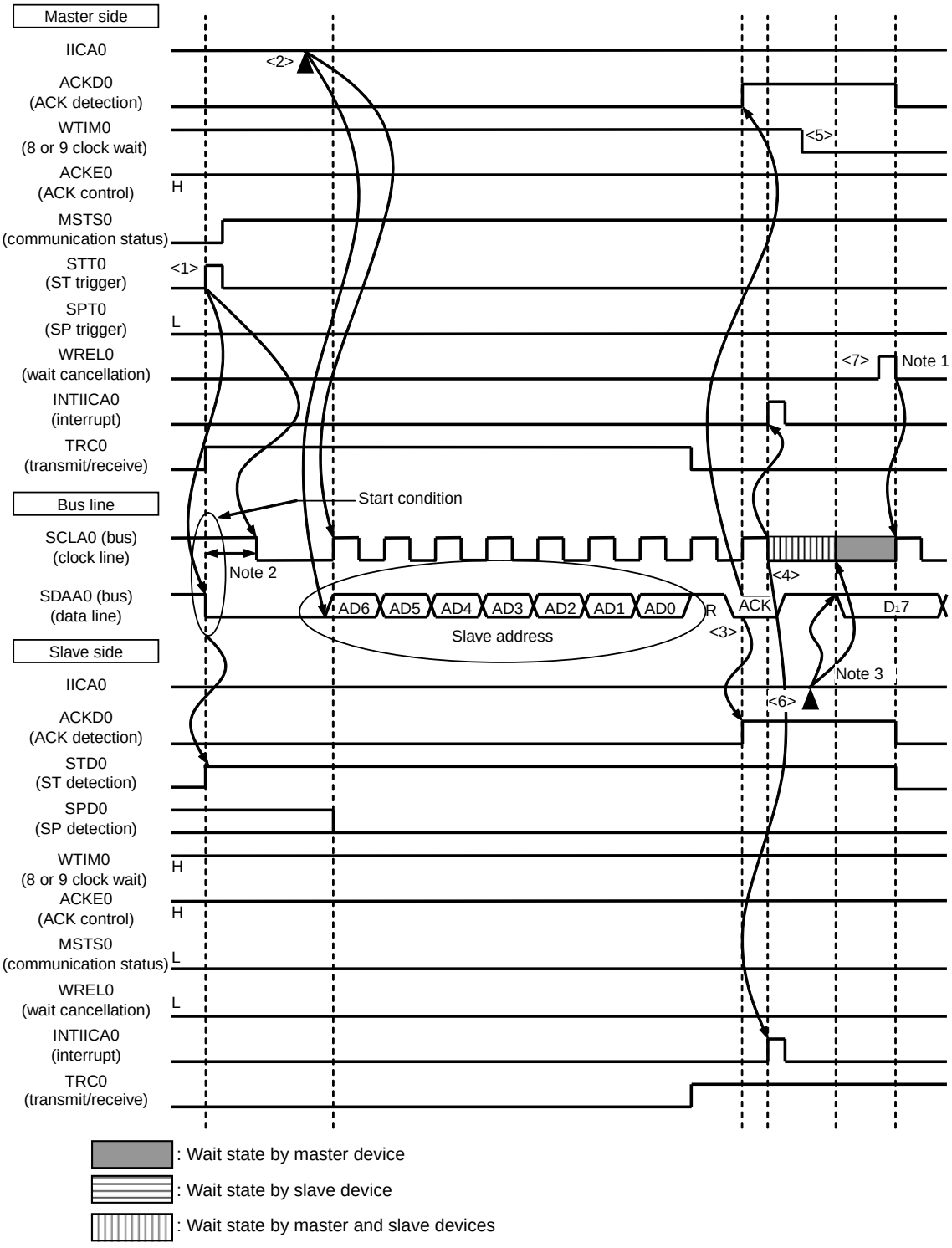
<i> The slave device reads the received data and releases the wait status (WREL0 = 1).

<ii> The start condition trigger is set again by the master device (STT0 = 1) and a start condition (i.e. SCLA0 =1 changes SDAA0 from 1 to 0) is generated once the bus clock line goes high (SCLA0 = 1) and the bus data line goes low (SDAA0 = 0) after the restart condition setup time has elapsed. When the start condition is subsequently detected, the master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<iii> The master device writing the address + R/W (transmission) to the IICA shift register (IICA0) enables the slave address to be transmitted.

**Figure 16-33. Example of Slave to Master Communication (When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)**

**(1) Start condition ~ address ~ data**




**Notes 1.** For releasing wait state during reception of a master device, write “FFH” to IICA0 or set the WREL0 bit.

**2.** Make sure that the time between the fall of the SDAA0 pin signal and the fall of the SCLA0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

**3.** Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission by a slave device.

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 18-33 are explained below.

<1> The start condition trigger is set by the master device (STT0 = 1) and a start condition (i.e. SCLA0 =1 changes SDAA0 from 1 to 0) is generated once the bus data line goes low (SDAA0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS0 = 1). The master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<2> The master device writes the address + R (reception) to the IICA shift register 0 (IICA0) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVA0 value) of a slave device**Note**, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match)**Note**.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIM0 = 0).

<6> The slave device writes the data to transmit to the IICA0 register and releases the wait status that it set by the slave device.

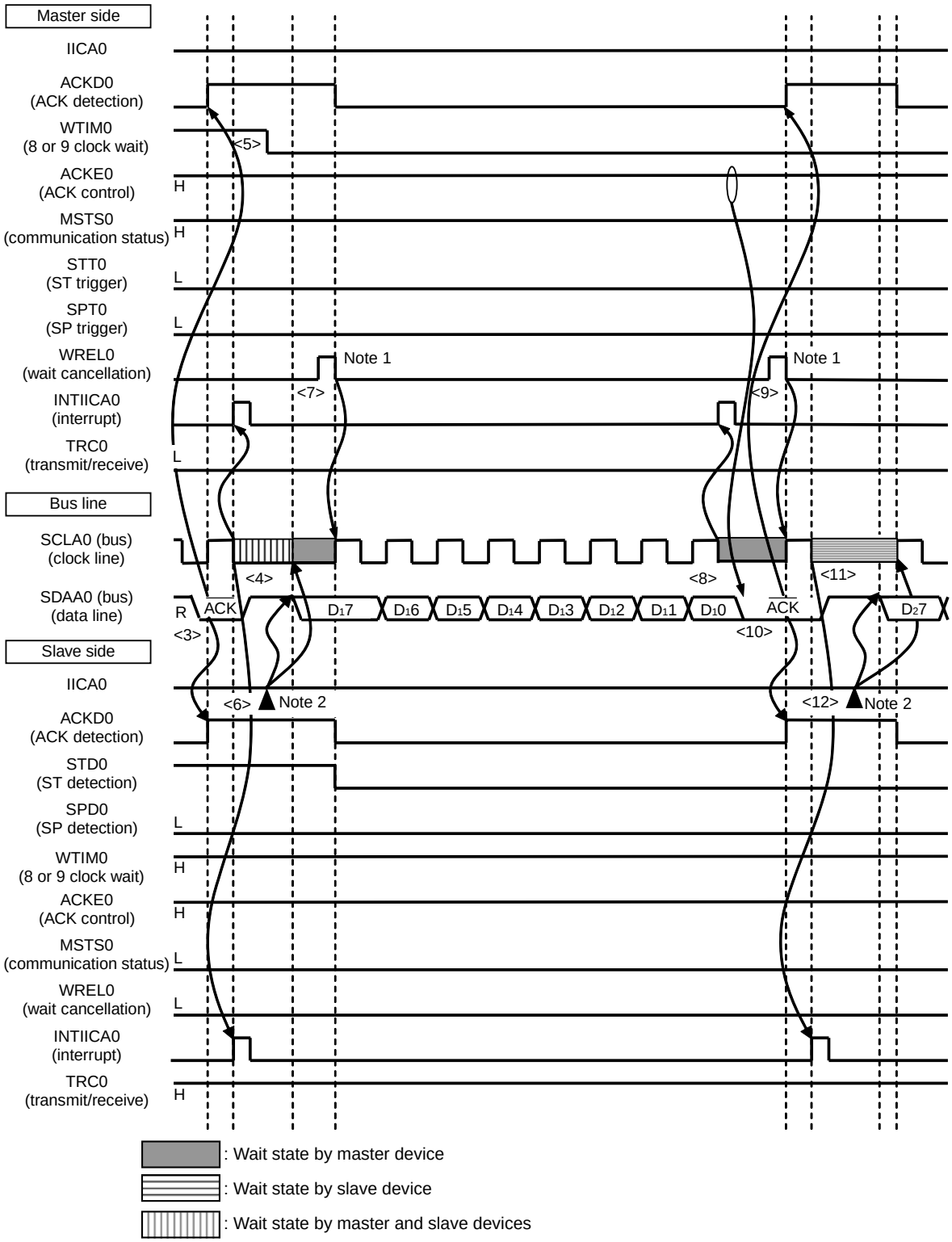
<7> The master device releases the wait status (WREL0 = 1) and starts transferring data from the slave device to the master device.

**Note** If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

**Remark** <1> to <19> in Figure 16-33 represent the entire procedure for communicating data using the $I^2C$ bus. Figure 16-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 16-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 16-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

**Figure 16-33. Example of Slave to Master Communication (When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)**

**(2) Address ~ data ~ data**




**Notes 1.** For releasing wait state during reception of a master device, write “FFH” to IICA0 or set the WREL0 bit.

**2.** Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission by a slave device.

The meanings of <3> to <12> in (2) Address ~ data ~ data in Figure 16-33 are explained below.

<3> In the slave device if the address received matches the address (SVA0 value) of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a wait status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match)[Note].

<5> The master device changes the timing of the wait status to the 8th clock (WTIM0 = 0).

<6> The slave device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the wait status that it set by the slave device.

<7> The master device releases the wait status (WREL0 = 1) and starts transferring data from the slave device to the master device.

<8> The master device sets a wait status (SCLA0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA0: end of transfer). Because of ACKE0 = 1 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the wait status (WREL0 = 1).

<10> The ACK is detected by the slave device (ACKD0 = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA0: end of transfer).

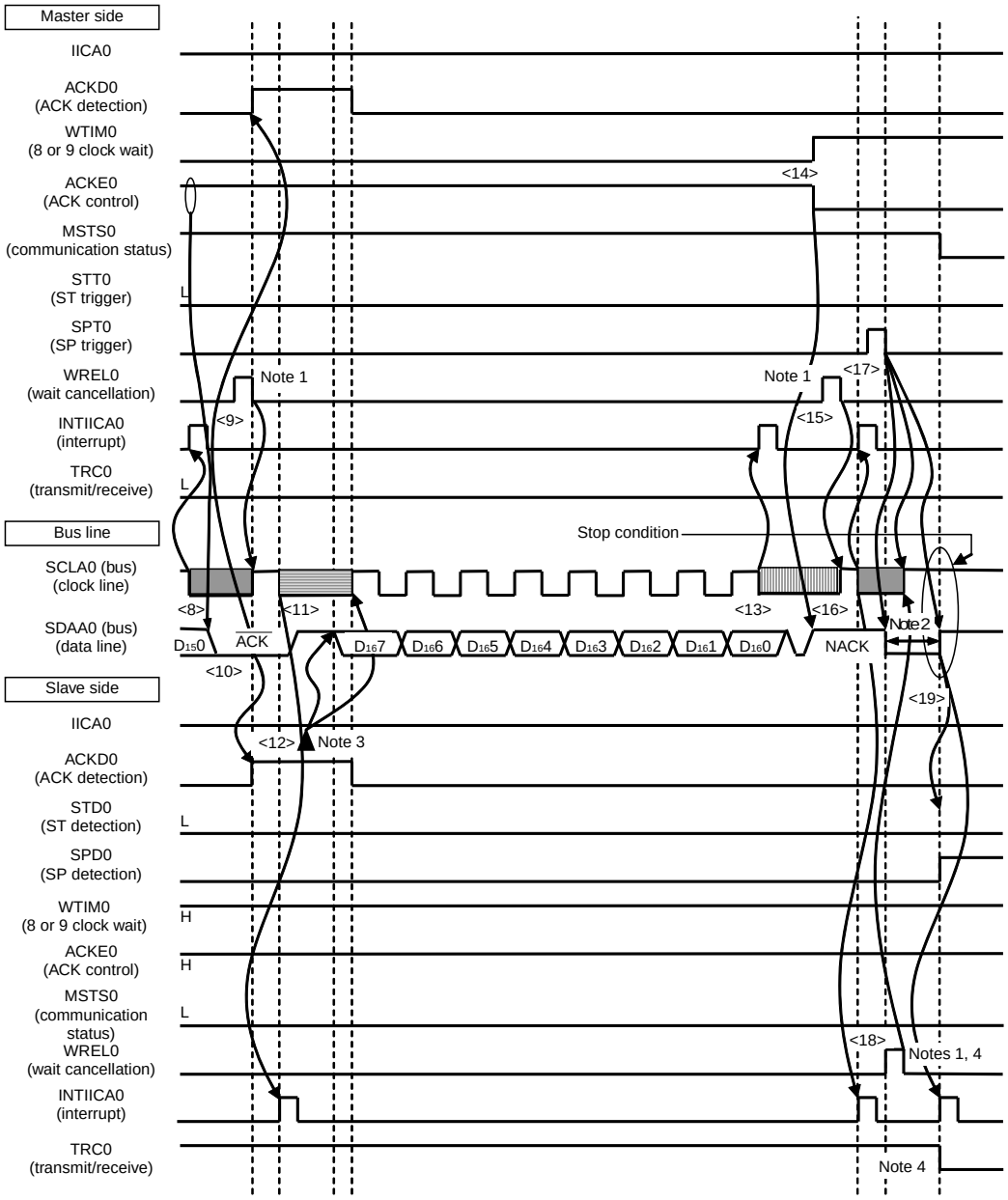
<12> By the slave device writing the data to transmit to the IICA0 register, the wait status set by the slave device is released. The slave device then starts transferring data to the master device.

**Note** If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

**Remark** <1> to <19> in Figure 16-33 represent the entire procedure for communicating data using the $I^2C$ bus. Figure 16-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 16-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 16-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

**Figure 16-33. Example of Slave to Master Communication**
**(When 8-Clock and 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/3)**

**(3) Data ~ data ~ stop condition**




**Notes 1.** To cancel a wait state, write “FFH” to IICA0 or set the WREL0 bit.

**2.** Make sure that the time between the rise of the SCLA0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

**3.** Write data to IICA0, not setting the WREL0 bit, in order to cancel a wait state during transmission by a slave device.

**4.** If a wait state during transmission by a slave device is canceled by setting the WREL0 bit, the TRC0 bit will be cleared.

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 16-33 are explained below.

<8> The master device sets a wait status (SCLA0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA0: end of transfer). Because of ACKE0 = 0 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the wait status (WREL0 = 1).

<10> The ACK is detected by the slave device (ACKD0 = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA0: end of transfer).

<12> By the slave device writing the data to transmit to the IICA register, the wait status set by the slave device is released. The slave device then starts transferring data to the master device.

<13> The master device issues an interrupt (INTIICA0: end of transfer) at the falling edge of the 8th clock, and sets a wait status (SCLA0 = 0). Because ACK control (ACKE0 = 1) is performed, the bus data line is at the low level (SDAA0 = 0) at this stage.

<14> The master device sets NACK as the response (ACKE0 = 0) and changes the timing at which it sets the wait status to the 9th clock (WTIM0 = 1).

<15> If the master device releases the wait status (WREL0 = 1), the slave device detects the NACK (ACKD0 = 0) at the rising edge of the 9th clock.

<16> The master device and slave device set a wait status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<17> When the master device issues a stop condition (SPT0 = 1), the bus data line is cleared (SDAA0 = 0) and the master device releases the wait status. The master device then waits until the bus clock line is set (SCLA0 = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the wait status (WREL0 = 1) to end communication. Once the slave device releases the wait status, the bus clock line is set (SCLA0 = 1).

<19> Once the master device recognizes that the bus clock line is set (SCLA0 = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDAA0 = 1) and issues a stop condition (i.e. SCLA0 =1 changes SDAA0 from 0 to 1). The slave device detects the generated stop condition and slave device issue an interrupt (INTIICA0: stop condition).

**Remark** <1> to <19> in Figure 16-33 represent the entire procedure for communicating data using the $I^2C$ bus. Figure 16-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 16-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 16-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

# CHAPTER 17 LIN/UART MODULE (RLIN3)

## 17.1 Overview

The LIN/UART module is a hardware LIN communication controller that supports LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2 and SAE J2602, and automatically performs frame communication and error determination.
The LIN/UART module is provided with UART mode and can also be used as a UART.
Table 17-1 gives the LIN/UART module specifications and Figures 17-1 and 17-2 show block diagrams of the LIN/UART module.

**Table 17-1. LIN/UART Module Specifications**

| Item | | | Specifications |
|---|---|---|---|
| Channel count | | | RL78/F13: 1 channel<br>RL78/F14: 1 to 2 channels |
| LIN communication function | Protocol | | LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2 and SAE J2602 |
| | Variable frame structure | Master | • Break (low) transmission width: 13 to 28 Tbits<br>• Break delimiter transmission width: 1 to 4 Tbits<br>• Inter-byte space (header): 0 to 7 Tbits (space between Sync field and ID field)Note 1<br>• Response space: 0 to 7 TbitsNote 1<br>• Inter-byte space: 0 to 3 Tbits (space between data bytes in response area)<br>• Wake-up: 1 to 16 Tbits |
| | | Slave | • Break reception width: 9.5 or 10.5 Tbits [for fixed baud rate]<br>: 10 or 11 Tbits [for auto baud rate]<br>• Response space: 0 to 7 Tbits<br>• Inter-byte space: 0 to 3 Tbits (space between data bytes in response area)<br>• Wake-up: 1 to 16 Tbits |
| | Checksum | | • Automatic operation for both transmission and reception<br>• Classic or enhanced selectable (for each frame) |
| | Response field data byte count | | Variable from 0 to 8 bytes<br>Multi-byte (9 or more bytes) response transmission and reception also possible |
| | Frame communication modes | Master | • Mode in which header transmission and response transmission/reception is started with a single transmission start request<br>• Mode in which header transmission and response transmission are started with separate transmission start requests (frame separate mode) |
| | | Slave | • Mode in which header is automatically received with fixed baud rate<br>• Mode in which header is automatically received with the baud rate set according to the sync field measurement result of the sync field and break field detected |
| | Wake-up transmission and reception | | LIN wake-up mode provided<br>• Wake-up transmission (1 to 16 Tbits)<br>• Wake-up reception<br>Low width of input signals measured |
| | Status | Master | • Successful frame/wake-up transmission<br>• Successful header transmission<br>• Successful frame/wake-up receptionNote 2<br>• Successful data 1 reception<br>• Error detection<br>• Operation mode<br>(LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode) |

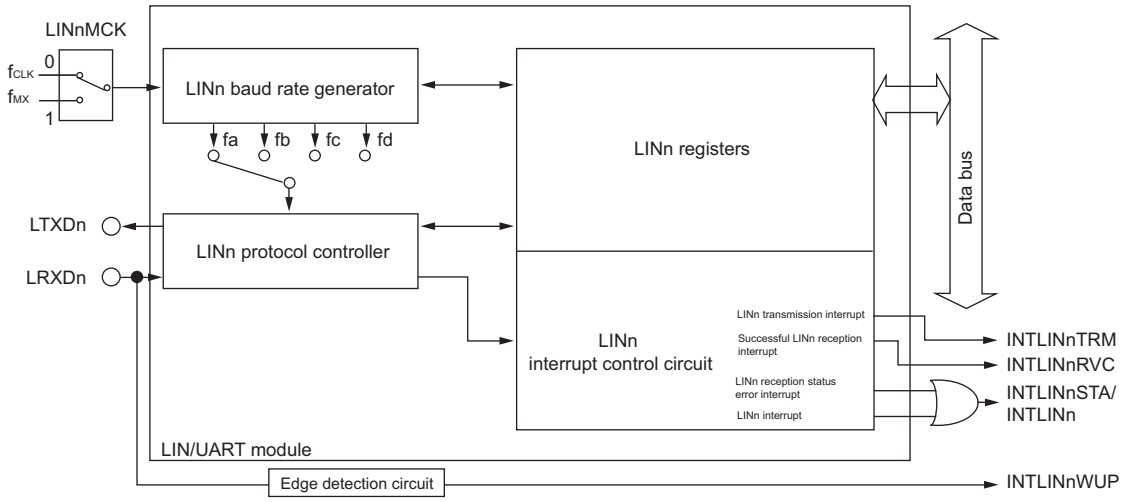
| Item | | Specifications | |
|---|---|---|---|
| LIN communication function | Status | Slave | • Successful frame/wake-up transmission<br>• Successful frame/wake-up reception[Note 2]<br>• Successful header reception<br>• Successful data 1 reception<br>• Error detection<br>• Operation mode<br>(LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode) |
| | Error status | Master | • Bit error<br>• Checksum error<br>• Frame timeout error/response timeout error<br>• Physical bus error<br>• Framing error<br>• Response preparation error |
| | | Slave | • Bit error<br>• Checksum error<br>• Frame timeout error/response timeout error<br>• Sync field error<br>• ID parity error<br>• Framing error<br>• Response preparation error |
| | Baud rate selection | Baud rate conforming to the LIN specifications generated using baud rate generator | |
| | Test mode | Self-test mode for user evaluation | |
| | Interrupt function | Master | • Successful header/frame/wake-up transmission<br>• Successful frame/wake-up reception[Note 2]<br>• Error detection |
| | | Slave | • Successful frame/wake-up transmission<br>• Header/frame/wake-up reception[Note 2]<br>• Error detection |

| Item | | Specifications |
|---|---|---|
| UART communication function | Data buffer | • Transmission data buffer/transmission data buffer for wait (exclusively for transmission; data length of 1, 7, 8, and 9 bits supported)<br>• UART buffer (exclusively for transmission; variable data length from 1 to 9 bits; data length of 7 and 8 bits supported)<br>• Reception data buffer (exclusively for reception; data length of 1, 7, 8, and 9 bits supported) |
| | Data format | • Character length: 7 or 8 bits<br>9 bits including the expansion bit supported.<br>• Transmission stop bit: 1 or 2 bits<br>• Parity function: odd, even, 0, or none<br>• LSB- or MSB-first transfer selectable<br>• Reverse input/output of transmission/reception data |
| | Status | • Transmission status<br>• Reception status<br>• Successful UART buffer transmission<br>• Error SUM<br>• Expansion bit detection<br>• ID match<br>• Reset mode status |
| | Error status | • Bit error<br>• Framing error<br>• Parity error<br>• Overrun error |
| | Baud rate selection | With the baud rate generator incorporated, any baud rate can be set. |
| | If the expected level is detected for any expansion bit, the 8 bits of the received data can be compared to the preset register data.<br>The stop bit received is guaranteed (start of transmission can be delayed when start of transmission is attempted during reception of the stop bit). | |
| | Interrupt function | • Transmission start/successful transmission<br>• Successful reception<br>• Status detection |

**Notes 1.** Since the same register is used for setting, the inter-byte space (header) = response space.

**2.** For wake-up reception, the low level width of the input signal is indicated.

**Figure 17-1. LIN/UART Module Block Diagram (1)**

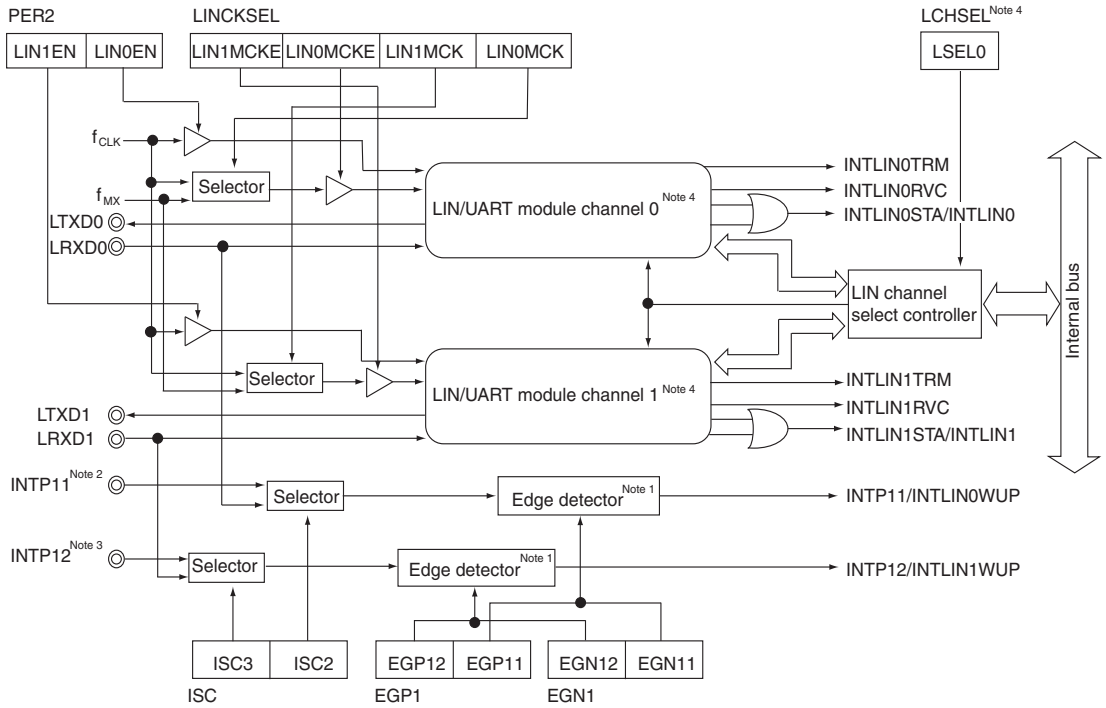



n = 0, 1

- LTXDn, LRXDn: LIN/UART module I/O pins
- LINn baud rate generator: Generates the LIN/UART module communication clock signal.
- LINn registers: LIN/UART module registers
- LINn interrupt controller: Controls interrupt requests generated by the LIN/UART module.

(n = 0, 1)

**Figure 17-2. LIN/UART Module Block Diagram (2)**




**Notes 1.** For details, see **CHAPTER 21 INTERRUPT FUNCTIONS**.
**2.** INTP11 is mounted only on the 80-pin products of RL78/F13 (LIN incorporated), 64- and 80-pin products of RL78/F13 (CAN and LIN incorporated), and 64-, 80-, and 100-pin products of RL78/F14.
**3.** INTP12 is mounted only on 64-, 80-, and 100-pin products of RL78/F14.
**4.** Only the registers of the channel that is selected with the LCHSEL register can be accessed using the CPU instructions and by the DTC. For the product incorporating one channel, set the LSEL0 bit in the LCHSEL register to 0.

Table 17-2 shows the I/O pins used in the LIN/UART module.

**Table 17-2. LIN/UART Module I/O Pins**

| Module Symbol | Pin Name | Input/Output | Function |
|---|---|---|---|
| LINn | LRXDn | Input | LIN communication function<br>Input pin of the UART communication function |
| | LTXDn | Output | LIN communication function<br>Output pin of the UART communication function |

(n = 0, 1)

The appropriate mode should be used for the LIN/UART module according to the application: LIN master, LIN slave, or UART.

LIN master

- LIN reset mode
- LIN mode (LIN master mode)
  - LIN wake-up mode
  - LIN operation mode
- LIN self-test mode

LIN slave

- LIN reset mode
- LIN mode (LIN slave mode [auto baud rate] or LIN slave mode [fixed baud rate])
  - LIN wake-up mode
  - LIN operation mode
- LIN self-test mode

UART

- LIN reset mode
- UART mode

## 17.2 Register Descriptions

Table 17-3 lists the LIN/UART module-related registers.

**Table 17-3. List of LIN/UART Module-Related Registers**

| Register Name | Symbol | LIN Master | LIN Slave | UART |
|---|---|---|---|---|
| Peripheral enable register 2 | PER2 | √ | √ | √ |
| Input switch control register | ISC | √ | √ | √ |
| LIN channel select register | LCHSEL | √ | √ | √ |
| LIN clock select register | LINCKSEL | √ | √ | √ |
| External interrupt rising edge enable registers 0, 1 | EGP0, EGP1 | √ | √ | √ |
| External interrupt falling edge enable registers 0, 1 | EGN0, EGN1 | √ | √ | √ |
| LIN wake-up baud rate select register | LWBR0/LWBR1 | √ | √ | √ |
| LIN/UART baud rate prescaler registers | LBRP0/LBRP1 | — | √ | √ |
| LIN/UART baud rate prescaler 0 register | LBRP00/LBRP10 | √ | √ | √ |
| LIN/UART baud rate prescaler 1 register | LBRP01/LBRP11 | √ | √ | √ |
| LIN self-test control register | LSTC0/LSTC1 | √ | √ | — |
| UART standby control register | LUSC0/LUSC1 | — | — | √ |
| LIN/UART mode register | LMD0/LMD1 | √ | √ | √ |
| LIN break field configuration register/ UART configuration register | LBFC0/LBFC1 | √ | √ | √ |
| LIN/UART space configuration register | LSC0/LSC1 | √ | √ | √ |
| LIN wake-up configuration register | LWUP0/LWUP1 | √ | √ | — |
| LIN interrupt enable register | LIE0/LIE1 | √ | √ | — |
| LIN/UART error detection enable register | LEDE0/LEDE1 | √ | √ | √ |
| LIN/UART control register | LCUC0/LCUC1 | √ | √ | √ |
| LIN/UART transmission control register | LTRC0/LTRC1 | √ | √ | √ |
| LIN/UART mode status register | LMST0/LMST1 | √ | √ | √ |
| LIN/UART status register | LST0/LST1 | √ | √ | √ |
| LIN/UART error status register | LEST0/LEST1 | √ | √ | √ |
| LIN/UART data field configuration register | LDFC0/LDFC1 | √ | √ | √ |
| LIN/UART ID buffer register | LIDB0/LIDB1 | √ | √ | √ |
| LIN checksum buffer register | LCBR0/LCBR1 | √ | √ | — |
| UART data buffer 0 register | LUDB00/LUDB10 | — | — | √ |
| LIN/UART data buffer 1 register | LDB01/LDB11 | √ | √ | √ |
| LIN/UART data buffer 2 register | LDB02/LDB12 | √ | √ | √ |
| LIN/UART data buffer 3 register | LDB03/LDB13 | √ | √ | √ |
| LIN/UART data buffer 4 register | LDB04/LDB14 | √ | √ | √ |
| LIN/UART data buffer 5 register | LDB05/LDB15 | √ | √ | √ |
| LIN/UART data buffer 6 register | LDB06/LDB16 | √ | √ | √ |
| LIN/UART data buffer 7 register | LDB07/LDB17 | √ | √ | √ |
| LIN/UART data buffer 8 register | LDB08/LDB18 | √ | √ | √ |
| UART operation enable register | LUOER0/LUOER1 | — | — | √ |
| UART option register 1 | LUOR01/LUOR11 | — | — | √ |

| Register Name | Symbol | LIN Master | LIN Slave | UART |
|---|---|---|---|---|
| UART transmission data register | LUTDR0/LUTDR1 | — | — | √ |
| | LUTDR0L/ LUTDR1L | — | — | √ |
| | LUTDR0H/ LUTDR1H | — | — | √ |
| UART reception data register | LURDR0/LURDR1 | — | — | √ |
| | LURDR0L/ LURDR1L | — | — | √ |
| | LURDR0H/ LURDR1H | — | — | √ |
| UART wait transmission data register | LUWTDR0/LUWTDR1 | — | — | √ |
| | LUWTDR0L/ LUWTDR1L | — | — | √ |
| | LUWTDR0H/ LUWTDR1H | — | — | √ |

√ : Used

—: Not used

When writing to a register not used, write 00H.

### 17.2.1 LIN Registers for Master Mode

#### (1) Input Switch Control Register (ISC)

The ISC2 and ISC3 bits in the ISC register are used in the LIN/UART module (RLIN3).
Setting bit 2 or bit 3 to 1 selects the input signal of the serial data input pin for the LIN/UART module as the external interrupt input.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Address: F0073H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ISC | 0 | 0 | 0 | 0 | ISC3 | ISC2 | 0 | ISC0 |

| ISC3 | Switching inputs for external interrupt INTP12 |
|---|---|
| 0 | INTP12 pin input signal is set as external interrupt input. |
| 1 | LRXD1 pin input signal is set as external interrupt input. |

| ISC2 | Switching inputs for external interrupt INTP11 |
|---|---|
| 0 | INTP11 pin input signal is set as external interrupt input. |
| 1 | LRXD0 pin input signal is set as external interrupt input. |

| ISC0 | Switching inputs for external interrupt INTP0 |
|---|---|
| 0 | INTP0 pin input signal is set as external interrupt input. (normal operation) |
| 1 | RXD0 pin input signal is set as external interrupt input. (wake-up signal detection) |

**Caution Bits 7 to 4 and 1 should always be set to 0.**

### (2) LIN Channel Select Register (LCHSEL)

Address: F007BH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | LSEL0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | LSEL0 | LIN Channel Select | 0: Selects LIN0. (LIN0 registers can be accessed.)<br>1: Selects LIN1. (LIN1 registers can be accessed.) | R/W |
| 7 to 1 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

**LSEL0 bit (LIN channel select bit)**

Since the LIN/UART module registers are not directly mapped on the CPU memory map, they should be accessed via the register windows. The register windows are mapped on addresses F06C1H to F06E9H.

Setting a value to the LSEL0 bit maps all the registers of the corresponding channel on the register window.

Setting the LSEL0 bit to 0 maps the LIN0 registers.

Setting the LSEL0 bit to 1 maps the LIN1 registers.

With the product incorporating one channel, set the LSEL0 bit to 0.

With the product incorporating two channels, set the LSEL0 bit to the applicable value before accessing a register of the channel to use.

### (3) Peripheral Enable Register 2 (PER2)

The PER2 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use the peripheral functions which are controlled by this register, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Address: F02C1H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | <3> | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PER2 | 0 | 0 | 0 | 0 | LIN1EN Note 1 | LIN0EN | 0 | CAN0EN Note 2 |

| LIN1EN Note 1 | Control of LIN1 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• Disables writing to the SFR used by LIN1.<br>• LIN1 is in the reset state. |
| 1 | Enables input clock supply.<br>• Enables reading from and writing to the SFR used by LIN1. |

| LIN0EN | Control of LIN0 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• Disables writing to the SFR used by LIN0.<br>• LIN0 is in the reset state. |
| 1 | Enables input clock supply.<br>• Enables reading from and writing to the SFR used by LIN0. |

| CAN0EN Note 2 | Control of CAN input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• Disables writing to the SFR used by CAN.<br>• CAN is in the reset state. |
| 1 | Enables input clock supply.<br>• Enables reading from and writing to the SFR used by CAN. |

**Notes 1.** Only in the RL78/F14 products with at least 128 Kbytes of code flash memory and the 100-pin products of the RL78/F14.

**2.** Only in the RL78/F13 (CAN and LIN incorporated) and RL78/F14 products.

**Caution Be sure to clear the following bits to 0.**

**Bits 0, 1, 3, 4, 5, 6, and 7 in the RL78/F13 (LIN incorporated) products**

**Bits 1, 3, 4, 5, 6, and 7 in the RL78/F13 (CAN and LIN incorporated) products and the RL78/F14 products with 30, 32, 48, 64, or 80 pins and up to 96 Kbytes of code flash memory**

**Bits 1, 4, 5, 6, and 7 in the RL78/F14 products with at least 128 Kbytes of code flash memory and the 100-pin products of the RL78/F14**

### (4) LIN Clock Select Register (LINCKSEL)

This register is used to control the communication clock source supplied to the LIN.

Address: F02C3H After reset: 00H R/W

| Symbol | 7 | 6 | <5> | <4> | 3 | 2 | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| LINCKSEL | 0 | 0 | LIN1MCKE Note | LIN0MCKE | 0 | 0 | LIN1MCK Note | LIN0MCK |

| LIN1MCKE Note | Control of supplying or stopping LIN1 communication clock source |
|---|---|
| 0 | Stops LIN communication clock source supply. |
| 1 | Enables LIN communication clock source supply. |

| LIN0MCKE | Control of supplying or stopping LIN0 communication clock source |
|---|---|
| 0 | Stops LIN communication clock source supply. |
| 1 | Enables LIN communication clock source supply. |

| LIN1MCK Note | Control of selecting LIN1 communication clock source |
|---|---|
| 0 | Selects the $f_{CLK}$ clock. |
| 1 | Selects the $f_{MX}$ clock. |

| LIN0MCK | Control of selecting LIN0 communication clock source |
|---|---|
| 0 | Selects the $f_{CLK}$ clock. |
| 1 | Selects the $f_{MX}$ clock. |

**Note** Only in the RL78/F14 products with at least 48 pins and 128 Kbytes or more of code flash memory and the 100-pin products of the RL78/F14.

**Cautions 1. Select the LINn operating clock with the LINnMCK bit before setting the LINnMCKE (n = 0, 1) bit to 1 (operating clock is supplied).**

**2. When operating LINn in SNOOZE mode, set the LINnMCK bit to 0.**

<R> **3. In case of LINnMCK is set to 1, do not used the timeout error detection.**
**In that case, set at least 1.2 times the frequency of the LIN communication clock source to the $f_{CLK}$ clock.**

**(5) External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)**

For details, see **21.3.4 External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)**.

### (6) LIN Wake-up Baud Rate Select Register (LWBRn)

Address: F06C1H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NSPB[3:0] | | | | LPRS[2:0] | | | LWBR0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | LWBR0 | Wake-up Baud Rate Select | 0: When LIN1.3 is used<br>1: When LIN 2.x is used | R/W |
| 3 to 1 | LPRS [2:0] | Prescaler Clock Select | b3 b1<br>0 0 0: 1/1<br>0 0 1: 1/2<br>0 1 0: 1/4<br>0 1 1: 1/8<br>1 0 0: 1/16<br>1 0 1: 1/32<br>1 1 0: 1/64<br>1 1 1: 1/128 | R/W |
| 7 to 4 | NSPB [3:0] | Bit Sampling Count Select | b7 b4<br>0 0 0 0: 16 sampling<br>1 1 1 1: 16 sampling<br>Settings other than the above are prohibited. | R/W |

Set the LWBRn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**LWBR0 bit (wake-up baud rate select bit)**

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the LWBRn register to 0. This allows the 2.5-Tbit or longer low level width of the input signal to be measured. When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. When the LWBR0 bit is set to 1, fa is always selected as the LIN system clock ($f_{LIN}$) in LIN wake-up mode regardless of the setting of LCKS bits in the LMDn register (setting of LCKS bits not affected). This allows the 2.5-Tbit or longer low level width of the input signal to be measured.

Setting the baud rate to 19200 bps with fa selected allows 130 μs or longer low level width of the input signal to be detected in LIN wake-up mode regardless of the setting of LCKS bits in the LMDn register.

**LPRS bits (prescaler clock select bits)**

The LPRS bits select the frequency division ratio for the prescaler.

The LIN communication clock source frequency is divided based on this prescaler.

**NSPB bits (bit sampling count select bits)**

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate).

In LIN master mode (LIN/UART mode select bits in LIN/UART mode register = 00b), set the NSPB bits to 0000b or 1111b (16 sampling).

### (7) LIN/UART Baud Rate Prescaler 0 Register (LBRPn0)

Address: F06C2H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | Setting Range | R/W |
|---|---|---|---|
| 7 to 0 | Assuming that the value set in this register is N (0 to 255), the baud rate prescaler 0 divides the frequency of the prescaler clock by N + 1. | 00H to FFH | R/W |

Set the LBRPn0 register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
The value set in this register is used to control the frequency of baud rate clock sources fa, fb, and fc.
Assuming that the value set in this register is N, baud rate prescaler 0 divides the frequency of the clock that is selected by the LPRS bits by N + 1.

### (8) LIN/UART Baud Rate Prescaler 1 Register (LBRPn1)

Address: F06C3H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | Setting Range | R/W |
|---|---|---|---|
| 7 to 0 | Assuming that the value set in this register is M (0 to 255), the baud rate prescaler 1 divides the frequency of the prescaler clock by M+1. | 00H to FFH | R/W |

Set the LBRPn1 register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
The value set in this register is used to control the frequency of baud rate clock source fd.
Assuming that the value set in this register is M, baud rate prescaler 1 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by M+1.

### (9) LIN Self-Test Control Register (LSTCn)

Address: F06C4H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | LSTM |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 7 to 0 | Writing A7H, 58H, and 01H successively to these bits places the LIN/UART module into LIN self-test mode. | | | R/W |
| 0 | LSTM | LIN Self-Test Mode | 0: The module is not in LIN self-test mode<br>1: The module is in LIN self-test mode. | R/W |

The LSTCn register cancels protection of LIN self-test mode.
Set the LSTCn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
Writing A7H, 58H, and 01H successively to the LSTCn register places the module into LIN self-test mode.
When successive writing is completed thus placing LIN self-test mode to be entered, the LSTM bit is set to 1.
Do not write any other value during successive writing.
For making transition to LIN self-test mode, refer to **17.6 LIN Self-Test Mode**.
Reading bits 6 to 1 returns 000000b, and reading bit 7 returns the undefined value.

**LSTM bit (LIN self-test mode bit)**
When transition to LIN self-test mode is completed, the LSTM bit is set to 1.
For leaving LIN self-test mode, refer to **17.6 LIN Self-Test Mode**.
Writing 1 to this bit does not affect the value of the LSTCn register if it is not a part of successive writing of A7H, 58H, and 01H.

### (10) LIN/UART Mode Register (LMDn)

Address: F06C8H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | LRDNFS | LIOS | LCKS[1:0] | | LMD[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 1, 0 | LMD[1:0] | LIN/UART Mode Select | b1 b0<br>0 0: LIN master mode | R/W |
| 3, 2 | LCKS[1:0] | LIN System Clock Select | b3 b2<br>0 0: fa (Clock generated by baud rate prescaler 0)<br>0 1: fb (1/2 clock generated by baud rate prescaler 0)<br>1 0: fc (1/8 clock generated by baud rate prescaler 0)<br>1 1: fd (1/2 clock generated by baud rate prescaler 1) | R/W |
| 4 | LIOS | LIN Interrupt Output Select | 0: LIN interrupt is used.<br>1: Transmission interrupt, successful reception interrupt, and reception status interrupt are used. | R/W |
| 5 | LRDNFS | LIN Reception Data Noise Filtering Disable | 0: The noise filter is enabled.<br>1: The noise filter is disabled. | R/W |
| 6 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0 | R/W |

Set the LMDn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**LMD[1:0] bits (LIN/UART mode select bits)**
The LMD bits select the LIN/UART module mode.
To use the LIN/UART module as a LIN master, set these bits to 00b.
With 00b set, the LIN/UART module operates in LIN master mode.

**LCKS[1:0] bits (LIN system clock select bits)**
The LCKS bits select the clock to be input to the protocol controller.
With 00b set, the protocol controller is provided with fa (clock generated by baud rate prescaler 0).
With 01b set, the protocol controller is provided with fb (1/2 clock generated by baud rate prescaler 0).
With 10b set, the protocol controller is provided with fc (1/8 clock generated by baud rate prescaler 0).
With 11b set, the protocol controller is provided with fd (1/2 clock generated by baud rate prescaler 1).
When the LWBR0 bit in the LWBRn register is 1 (LIN 2.x is used) and the LMSTn register is 01h (LIN wake-up mode), fa is always input to the protocol controller regardless of the setting of LCKS bits (setting of LCKS bits not affected).

**LIOS bit (LIN interrupt output select bit)**
The LIOS bit selects the number of interrupt outputs from the LIN/UART module.
With 0 set, the LIN interrupt is generated from the LIN/UART module.
With 1 set, the transmission interrupt, successful reception interrupt, and reception status interrupt are generated from the LIN/UART module.
For each interrupt source, refer to **17.9 Interrupts**.

**LRDNFS bit (LIN reception data noise filtering disable bit)**
The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

### (11) LIN Break Field Configuration Register/UART Configuration Register (LBFCn)

Address: F06C9H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | BDT[1:0] | | BLT[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 3 to 0 | BLT[3:0] | Transmission Break (Low) Width Select | b3 b0<br>0 0 0 0: 13 Tbits<br>0 0 0 1: 14 Tbits<br>0 0 1 0: 15 Tbits<br>:<br>1 1 1 0: 27 Tbits<br>1 1 1 1: 28 Tbits | R/W |
| 5, 4 | BDT[1:0] | Transmission Break Delimiter (High) Width Select | b5 b4<br>0 0: 1 Tbit<br>0 1: 2 Tbits<br>1 0: 3 Tbits<br>1 1: 4 Tbits | R/W |
| 6 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |

Set the LBFCn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
Some combinations of the set values result in the length of a frame exceeding the frame timeout time. Set the appropriate values in this register.

**BLT[3:0] bits (transmission break (low) width select bits)**
The BLT bits set the break (low) width of the transmission frame header.
13 Tbits to 28 Tbits can be set.

**BDT bits (transmission break delimiter (high) width select bits)**
The BDT bits set the break delimiter (high) width of the transmission frame header field.
1 Tbit to 4 Tbits can be set.

### (12) LIN/UART Space Configuration Register (LSCn)

Address: F06CAH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | IBS[1:0] | | — | IBSH[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 2 to 0 | IBSH[2:0] | Inter-Byte Space (Header)/ Response Space Select | b2 b0<br>0 0 0: 0 Tbit<br>0 0 1: 1 Tbit<br>0 1 0: 2 Tbits<br>0 1 1: 3 Tbits<br>1 0 0: 4 Tbits<br>1 0 1: 5 Tbits<br>1 1 0: 6 Tbits<br>1 1 1: 7 Tbits | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 5, 4 | IBS[1:0] | Inter-Byte Space Select | b5 b4<br>0 0: 0 Tbit<br>0 1: 1 Tbit<br>1 0: 2 Tbits<br>1 1: 3 Tbits | R/W |
| 7, 6 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

Set the LSCn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

**IBSH[2:0] bits (inter-byte space (header)/response space select bits)**
The IBSH bits set the width of the inter-byte space (header) of the transmission frame header field and the response space.
0 Tbit to 7 Tbits can be set.
The response space setting is enabled only during response transmission; setting is disabled during response reception.
The inter-byte space (header) is equal to the response space.

**IBS[1:0] bits (inter-byte space select bits)**
The IBS bits set the width of the inter-byte space of the transmission frame response field.
0 Tbit to 3 Tbits can be set.
These bits are enabled only during response transmission; these are disabled during response reception.

### (13) LIN Wake-up Configuration Register (LWUPn)

Address: F06CBH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WUTL[3:0] | | | | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 3 to 0 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |
| 7 to 4 | WUTL[3:0] | Wake-up Transmission Low Width Select | b7 b4<br>0 0 0 0: 1 Tbit<br>0 0 0 1: 2 Tbits<br>0 0 1 0: 3 Tbits<br>0 0 1 1: 4 Tbits<br>:<br>1 1 0 0: 13 Tbits<br>1 1 0 1: 14 Tbits<br>1 1 1 0: 15 Tbits<br>1 1 1 1: 16 Tbits | R/W |

Set the LWUPn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**WUTL[3:0] bits (wake-up transmission low width select bits)**

The WUTL bits set the low width of the wake-up signal transmission.

1 Tbit to 16 Tbits can be set.

When the LWBR0 bit in the LWBRn register is 1 (LIN 2.x is used), fa is always selectd as the LIN system clock ($f_{LIN}$) regardless of the setting of LCKS bits in the LMDn register (setting of LCKS bits not affected).

### (14) LIN Interrupt Enable Register (LIEn)

Address: F06CCH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | SHIE | ERRIE | FRCIE | FTCIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | FTCIE | Successful Frame/Wake-up Transmission Interrupt Enable | 0: Disables successful frame/wake-up transmission interrupt.<br>1: Enables successful frame/wake-up transmission interrupt. | R/W |
| 1 | FRCIE | Successful Frame/Wake-up Reception Interrupt Enable | 0: Disables successful frame/wake-up reception interrupt.<br>1: Enables successful frame/wake-up reception interrupt. | R/W |
| 2 | ERRIE | Error Detection Interrupt Enable | 0: Disables error detection interrupt.<br>1: Enables error detection interrupt. | R/W |
| 3 | SHIE | Successful Header Transmission Interrupt Enable | 0: Disables successful header transmission interrupt.<br>1: Enables successful header transmission interrupt. | R/W |
| 7 to 4 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

Set the LIEn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**FTCIE bit (successful frame/wake-up transmission interrupt enable bit)**
The FTCIE bit enables or disables interrupt generation upon successful transmission of a frame or a wake-up signal.
With 0 set, the interrupt is not generated when the FTC flag in the LSTn register is set to 1.
With 1 set, the interrupt is generated when the FTC flag in the LSTn register is set to 1.

**FRCIE bit (successful frame/wake-up reception interrupt enable bit)**
The FRCIE bit enables or disables interrupt generation upon successful reception of a frame or a wake-up signal (counting of low width of the input signal).
With 0 set, the interrupt is not generated when the FRC flag in the LSTn register is set to 1.
With 1 set, the interrupt is generated when the FRC flag in the LSTn register is set to 1.

**ERRIE bit (error detection interrupt enable bit)**
The ERRIE bit enables or disables interrupt generation upon detection of an error.
With 0 set, the interrupt is not generated when the ERR flag in the LSTn register is set to 1.
With 1 set, the interrupt is generated when the ERR flag in the LSTn register is set to 1.
Interrupt sources can be the bit error, physical bus error, frame/response timeout error, framing error, checksum error, and response preparation error.
Detection of the bit error, physical bus error, frame/response timeout error, and framing error can be enabled or disabled using the LEDEn register.

**SHIE bit (successful header transmission interrupt enable bit)**
The SHIE bit enables or disables interrupt generation upon successful transmission of a header.
With 0 set, the interrupt is not generated when the HTRC flag in the LSTn register is set to 1.
With 1 set, the interrupt is generated when the HTRC flag in the LSTn register is set to 1.

### (15) LIN/UART Error Detection Enable Register (LEDEn)

Address: F06CDH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LTES | — | — | — | FERE | FTERE | PBERE | BERE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | BERE | Bit Error Detection Enable | 0: Disables bit error detection.<br>1: Enables bit error detection. | R/W |
| 1 | PBERE | Physical Bus Error Detection Enable | 0: Disables physical bus error detection.<br>1: Enables physical bus error detection. | R/W |
| 2 | FTERE | Timeout Error Detection Enable | 0: Disables frame/response timeout error detection.<br>1: Enables frame/response timeout error detection. | R/W |
| 3 | FERE | Framing Error Detection Enable | 0: Disables framing error detection.<br>1: Enables framing error detection. | R/W |
| 4 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 5 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 6 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 7 | LTES | Timeout Error Select | 0: Frame timeout error<br>1: Response timeout error | R/W |

Set the LEDEn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**BERE bit (bit error detection enable bit)**

The BERE bit enables or disables detection of the bit error.

<R> Set 1(the bit error detection enables) to this bit.

The bit error detection result of the bit error is indicated in the BER flag in the LESTn register.

For details of the bit error, refer to **17.4.6 Error Status.**

**PBERE bit (physical bus error detection enable bit)**

The PBERE bit enables or disables detection of the physical bus error.
With 0 set, the physical bus error is not detected.
With 1 set, the physical bus error is detected.
When this bit is set to 1, the detection result is indicated in the PBER flag in the LESTn register.
For details of the physical bus error, refer to **17.4.6 Error Status**.

**FTERE bit (timeout error detection enable bit)**

The FTERE bit enables or disables detection of the frame timeout error or the response timeout error.
With 0 set, the frame timeout error or response timeout error is not detected.
With 1 set, the frame timeout error or response timeout error is detected.
When this bit is set to 1, the detection result is indicated in the FTER flag in the LESTn register.
With the LTES bit, either the frame timeout error or response timeout error can be selected.
Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.
For details of the timeout error, refer to **17.4.6 Error Status**.

**FERE bit (framing error detection enable bit)**

The FERE bit enables or disables detection of the framing error.

<R> Set 1(the framing error detection enables) to this bit.

The framing error detection result is indicated in the FER flag in the LESTn register.

For details of the framing error, refer to **17.4.6 Error Status**.

**LTES bit (timeout error select bit)**

The LTES bit selects the specific timeout function to be used.

With 0 set, the timeout function applies to frame timeout.

With 1 set, the timeout function applies to response timeout.

For details of the timeout error, refer to **17.4.6 Error Status**.

### (16) LIN/UART Control Register (LCUCn)

Address: F06CEH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | OM1 | OM0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | OM0 | LIN Reset | 0: LIN reset mode is caused.<br>1: LIN reset mode is canceled. | R/W |
| 1 | OM1 | LIN Mode Select | 0: LIN wake-up mode is caused.<br>1: LIN operation mode is caused. | R/W |
| 7 to 2 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

Set the LCUCn register to 01H to cause a transition to LIN wake-up mode after canceling LIN reset mode, and set the LCUCn register to 03H to cause a transition to LIN operation mode.
In LIN self-test mode, set the LCUCn register to 03H after a transition to LIN self-test mode is completed.
After a value is written to this register, confirm that the value written is actually indicated in the LMSTn register before writing another value.

**OM0 bit (LIN reset bit)**
The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.
With 0 set, LIN reset mode is caused.
With 1 set, LIN reset mode is canceled.

**OM1 bit (LIN mode select bit)**
The OM1 bit selects the specific operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.
With 0 set, LIN wake-up mode is caused.
With 1 set, LIN operation mode is caused.
This register is valid only when the OMM0 bit in the LMSTn register is 1.
Writing a value to this bit is disabled while the FTS bit in the LTRCn register is 1.

### (17) LIN/UART Transmission Control Register (LTRCn)

Address: F06D0H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | RTS | FTS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | FTS | Frame Transmission or Wake-up Transmission/Reception Start | 0: Frame transmission or wake-up transmission/reception is stopped.<br>1: Frame transmission or wake-up transmission/reception is started. | R/W |
| 1 | RTS | Response Transmission/Reception Start | 0: Response transmission/reception is stopped in frame separate mode.<br>1: Response transmission/reception is started in frame separate mode. | R/W |
| 2 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 7 to 3 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

**FTS bit (frame transmission or wake-up transmission/reception start bit)**

Set the FTS bit to 1 to start frame or wake-up transmission.
Also set this bit to 1 to allow wake-up reception (counting of the low width of the input signal).
Only 1 can be written to this bit; 0 cannot be written.
Writing a value to this bit is disabled when the OMM0 bit in the LMSTn register is 0 (LIN reset mode). This bit is set to 0 upon completion of frame or wake-up communication and transition to LIN reset mode.

**RTS bit (response transmission/reception start bit)**

Set the RTS bit to 1 in frame separate mode after header transmission is started (FTS bit is 1) and response transmission data is ready. Once set, this bit is automatically cleared to 0 upon completion of frame communication or transition to LIN reset mode.
Only 1 can be written to this bit; 0 cannot be written.
To write 1 to this bit, write 02H to the LTRCn register by using an 8-bit data transfer instruction.
Writing a value to this bit is disabled when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 upon completion of data group communication or transition to LIN reset mode.

### (18) LIN/UART Mode Status Register (LMSTn)

Address: F06D1H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | OMM1 | OMM0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | OMM0 | LIN Reset Status Monitor | 0: The module is in LIN reset mode.<br>1: The module is not in LIN reset mode. | R |
| 1 | OMM1 | LIN Mode Status Monitor | 0: The module is in LIN wake-up mode.<br>1: The module is in LIN operation mode. | R |
| 7 to 2 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

**OMM0 bit (LIN reset status monitor)**

**OMM1 bit (LIN mode status monitor)**

The OMM0 and OMM1 bits indicate the current operation mode.

### (19) LIN/UART Status Register (LSTn)

Address: F06D2H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | HTRC | D1RC | — | — | ERR | — | FRC | FTC |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | FTC | Successful Frame/Wake-up Transmission Flag | 0: Frame or wake-up transmission has not been completed.<br>1: Frame or wake-up transmission has been completed. | R/W |
| 1 | FRC | Successful Frame/Wake-up Reception Flag | 0: Frame or wake-up reception has not been completed.<br>1: Frame or wake-up reception has been completed. | R/W |
| 2 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 3 | ERR | Error Detection Flag | 0: No error has been detected.<br>1: Error has been detected. | R |
| 4, 5 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |
| 6 | D1RC | Successful Data 1 Reception Flag | 0: Data 1 reception has not been completed.<br>1: Data 1 reception has been completed. | R/W |
| 7 | HTRC | Successful Header Transmission Flag | 0: Header transmission has not been completed.<br>1: Header transmission has been completed. | R/W |

The LSTn register is automatically cleared to 00H upon transition to LIN reset mode and start of the next communication (the FTS bit in the LTRCn register is 1).
In LIN reset mode, writing to this register is disabled. In LIN reset mode, the register retains 00H.
To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits by using an 8-bit data transfer instruction.

**FTC flag (successful frame/wake-up transmission flag)**
Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FTC flag is set to 1 upon completion of response or wake-up transmission. Here, an interrupt is generated if the FTCIE bit in the LIEn register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (the FTS bit in the LTRCn register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

**FRC flag (successful frame/wake-up reception flag)**
Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FRC flag is set to 1 upon completion of response or wake-up reception. Here, an interrupt is generated if the FRCIE bit in the LIEn register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (the FTS bit in the LTRCn register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

**ERR flag (error detection flag)**

The ERR flag is set to 1 upon detection of an error (any of the LESTn register flags is 1). Here, an interrupt is generated if the ERRIE bit in the LIEn register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (the FTS bit in the LTRCn register is 1), write 0 to the RPER, CSER, FER, FTER, PBER, and BER flags in the LESTn register in LIN operation mode or LIN wake-up mode. This clears the ERR flag to 0.

**D1RC flag (successful data 1 reception flag)**

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value that has been retained before 1 is written. The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt is not generated. To clear the bit to 0 before the next communication (the FTS bit in the LTRCn register is 1), write 0 to the bit in LIN operation mode.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

**HTRC flag (successful header transmission flag)**

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written. The HTRC flag is set to 1 upon completion of header transmission. Here, an interrupt is generated if the SHIE bit in the LIEn register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (the FTS bit in the LTRCn register is 1), write 0 to the bit in LIN operation mode.

### (20) LIN/UART Error Status Register (LESTn)

Address: F06D3H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RPER | — | CSER | — | FER | FTER | PBER | BER |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | BER | Bit Error Flag | 0: Bit error has not been detected.<br>1: Bit error has been detected. | R/W |
| 1 | PBER | Physical Bus Error Flag | 0: Physical bus error has not been detected.<br>1: Physical bus error has been detected. | R/W |
| 2 | FTER | Timeout Error Flag | 0: Frame/response timeout error has not been detected.<br>1: Frame/response timeout error has been detected. | R/W |
| 3 | FER | Framing Error Flag | 0: Framing error has not been detected.<br>1: Framing error has been detected. | R/W |
| 4 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 5 | CSER | Checksum Error Flag | 0: Checksum error has not been detected.<br>1: Checksum error has been detected. | R/W |
| 6 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 7 | RPER | Response Preparation Error Flag | 0: Response preparation error has not been detected.<br>1: Response preparation error has been detected. | R/W |

The LESTn register is automatically cleared to 00H upon transition to LIN reset mode and start of the next communication (the FTS bit in the LTRCn register is 1).
In LIN reset mode, writing to this register is disabled. In LIN reset mode, the register retains 00H.
When the FTS bit in the LTRCn register is 1 (frame transmission or wake-up transmission/reception is started), do not write a value to this register.
To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits by using an 8-bit data transfer instruction.

**BER flag (bit error flag)**
Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The BER flag is set to 1 upon bit error detection if the BERE bit in the LEDEn register is 1 (bit error detection is enabled). To clear the bit to 0 before the next communication (the FTS bit in the LTRCn register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

**PBER flag (physical bus error flag)**
Only 0 can be written to the PBER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The PBER flag is set to 1 upon physical bus error detection if the PBERE bit in the LEDEn register is 1 (physical bus error detection is enabled). To clear the bit to 0 before the next communication (the FTS bit in the LTRCn register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

**FTER flag (timeout error flag)**
Only 0 can be written to the FTER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FTER flag is set to 1 upon frame timeout error or response timeout error detection if the FTERE bit in the LEDEn register is 1 (frame/response timeout error detection is enabled). To clear the bit to 0 before the next communication (the FTS bit in the LTRCn register is 1), write 0 to the bit in LIN operation mode.

**FER flag (framing error flag)**

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value that has been retained before 1 is written. The FER flag is set to 1 upon framing error detection if the FERE bit in the LEDEn register is 1 (framing error detection is enabled). To clear the bit to 0 before the next communication (the FTS bit in the LTRCn register is 1), write 0 to the bit in LIN operation mode.

**CSER flag (checksum error flag)**

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value that has been retained before 1 is written. The CSER flag is set to 1 upon checksum error detection. To clear the bit to 0 before the next communication (the FTS bit in the LTRCn register is 1), write 0 to the bit in LIN operation mode.

**RPER flag (response preparation error flag)**

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value that has been retained before 1 is written. The RPER flag is set to 1 upon response preparation error detection. To clear the bit to 0 before the next communication (the FTS bit in the LTRCn register is 1), write 0 to the bit in LIN operation mode.

### (21) LIN/UART Data Field Configuration Register (LDFCn)

Address: F06D4H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LSS | FSM | CSM | RFT | RFDL[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 3 to 0 | RFDL[3:0] | Response Field Length Select | b3 b0<br>0 0 0 0: 0 byte (+ checksum)<br>0 0 0 1: 1 byte (+ checksum)<br>0 0 1 0: 2 bytes (+ checksum)<br>:<br>0 1 1 1: 7 bytes (+ checksum)<br>1 0 0 0: 8 bytes (+ checksum)<br>Settings other than the above are prohibited. | R/W |
| 4 | RFT | Response Field Communication Direction Select | 0: Reception<br>1: Transmission | R/W |
| 5 | CSM | Checksum Select | 0: Classic checksum mode<br>1: Enhanced checksum mode | R/W |
| 6 | FSM | Frame Separate Mode Select | 0: Frame separate mode is not set.<br>1: Frame separate mode is set. | R/W |
| 7 | LSS | Transmission/Reception Continuation Select | 0: The data group to be transmitted/received next is the last one.<br>1: The data group to be transmitted/received next is not the last one. (Checksum is not included.) | R/W |

**RFDL[3:0] bits (response field length select bits)**

The RFDL bits set the length of the response field data.
The data length can be 0 to 8 bytes excluding the checksum size.
To transmit response data with the FSM bit set to 0 (not frame separate mode), set the RFDL bits before header transmission (the FTS bit in the LTRCn register is 0).
To transmit response data with the FSM bit set to 1 (frame separate mode), set the RFDL bits before response transmission (the RTS bit in the LTRCn register is 0).
To receive response data, set the RFDL bits before header transmission (the FTS bit in the LTRCn register is 0).

When response data of 9 bytes or more is to be transmitted or received, set the RFDL bits before data group transmission/reception (RTS bit in the LTRCn register is 0).
During communication of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

**RFT bit (response field communication direction select bit)**
The RFT bits set the direction of the response field/wake-up signal communication.
With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (low width of the input signal is counted).
With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.
Set this bit when the FTS bit in the LTRCn register is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, do not change the RFT bit setting after the first data group through the last data group.

**CSM bit (checksum select bit)**
The CSM bit sets the checksum mode.
With 0 set, classic checksum mode is selected.
With 1 set, enhanced checksum mode is selected.
When the timeout error is used (the FTERE bit in the LEDEn register is 1), the specific timeout time depends on the setting of this bit. For details of the bit error, refer to **17.4.6 Error Status**.
Set this bit when the FTS bit in the LTRCn register is 0 (frame transmission or wake-up transmission/reception is halted).
When response of 9 bytes or more is to be transmitted or received, do not change the CSM bit setting after the first data group through the last data group.
During communication of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

**FSM bit (frame separate mode select bit)**
The FSM bit sets the response communication mode.
With 0 set, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the LTRCn register is 1), response is transmitted/received without the RTS bit in the LTRCn register being set.
With 1 set, frame separate mode is selected. If the RTS bit in the LTRCn register is set to 1 during header transmission, response is transmitted after successful header transmission.
For response reception (the RFT bit is 0), set the FSM bit to 0.
When causing a transition to LIN self-test mode, set this bit to 0 before transition.
For details of frame separate mode, refer to **17.4.3 (1) (a) Frame Separate Mode**.
Set this bit when the FTS bit in the LTRCn register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, set the FSM bit to 1.

**LSS bit (transmission/reception continuation select bit)**
The LSS bit shows that the next data group to be transmitted or received is not the last one when response data of 9 bytes or more is to be transmitted or received.
With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one.
With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.
Set the LSS bit only when the FSM bit is 1 (frame separate mode) and response data of 9 bytes or more is to be transmitted or received.
Set this bit while the RTS bit in the LTRCn register is 0 (response transmission/reception stopped).

### (22) LIN/UART ID Buffer Register (LIDBn)

Address: F06D5H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IDP[1:0] | | ID[5:0] | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 5 to 0 | ID[5:0] | ID Setting | Sets the 6-bit ID value to be transmitted in the ID field. | R/W |
| 7, 6 | IDP[1:0] | Parity Setting | Sets the parity bits (P) to be transmitted in the ID field. | R/W |

Set the LIDBn register when the FTS bit in the LTRCn register is 0 (frame transmission or wake-up transmission/reception is halted).

In LIN self-test mode, this register operates as follows:
Write the value to be transmitted before communication. The reversed value of the value received can be read from the register after frame transmission/reception is completed (after loopback).
For details of LIN self-test mode, refer to **17.6 LIN Self-Test Mode**.

**ID bits (ID setting bits)**
The ID bit sets the 6-bit ID value to be transmitted in the ID field of the LIN frame.

**IDP bits (parity setting bits)**
The IDP bits set the parity bits (P0 and P1) to be transmitted in the ID field of the LIN frame.
The IDP0 bit is P0 and the IDP1 bit is P1.
Since parity is not automatically calculated, set the calculation result. Note that if the erroneous result is set, it is transmitted as is.

### (23) LIN Checksum Buffer Register (LCBRn)

Address: F06D6H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | R/W |
|---|---|---|
| 7 to 0 | Holds the checksum value transmitted or received. | R/W |

In LIN mode, this register operates as follows:

- When the RFT bit in the LDFCn register is 1 (transmission):

  The value transmitted can be read from the register. Read the value after transmission is completed.

  Writing to this register is invalid.

- When the RFT bit in the LDFCn register is 0 (reception):

  The value received can be read from the register. Read the value after reception is completed.

  Writing to this register is invalid.

In LIN self-test mode, this register operates as follows:

- When the RFT bit in the LDFCn register is 1 (transmission):

  The reversed value of the value transmitted can be read from the register after frame transmission is completed (after loopback).

- When the RFT bit in the LDFCn register is 0 (reception):

  Write the value to be received before communication.
  The reversed value of the value received can be read from the register after frame transmission/reception is completed (after loopback).

For details of LIN self-test mode, refer to **17.6 LIN Self-Test Mode**.

Set the LCBRn register when the FTS bit in the LTRCn register is 0 (frame transmission or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for the other data groups.

### (24) LIN/UART Data Buffer m Register (LDBnm) (m = 1 to 8)

Address: LDBn1 F06D8H, LDBn2 F06D9H, LDBn3 F06DAH, LDBn4 F06DBH, LDBn5 F06DCH, LDBn6 F06DDH, LDBn7 F06DEH, LDBn8 F06DFH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | Setting Range | R/W |
|---|---|---|---|
| 7 to 0 | Sets the data to be transmitted or allows the received data to be read. | 00H to FFH | R/W |

For response transmission:
These registers set the data to be transmitted in the response field.
Use these registers with the following settings.

- RFT in LDFCn register is 1 (transmission)
- FSM in LDFCn register is 0 (not frame separate mode)
- FTS bit in LTRCn register is 0 (frame transmission or wake-up transmission/reception is halted)

or

- RFT in LDFCn register is 1 (transmission)
- FSM in LDFCn register is 1 (frame separate mode)
- RTS in LTRCn register is 0 (response transmission/reception is halted)

For response reception:
These registers hold the data received in the response field.
The received data is overwritten. If an error is detected, the data prior to reception interruption is stored in the register.
Do not read these registers when the FTS bit is 1 (frame transmission or wake-up transmission/reception is started)

For transmission of response data of 9 bytes or more:
Use these registers with the following settings.

- RFT in LDFCn register is 1 (transmission)
- FSM in LDFCn register is 1 (frame separate mode)
- RTS in LTRCn register is 0 (response transmission/reception is halted)

For reception of response data of 9 bytes or more:
Do not read these registers when the RTS bit is 1 (response transmission/reception is started).

In LIN self-test mode, these registers operate as follows:
Write the value to be transmitted before communication.
The reversed value of the value received can be read from the register after frame transmission/reception is completed (after loopback).

For details of LIN self-test mode, refer to **17.6 LIN Self-Test Mode**.

### 17.2.2 LIN Registers for Slave Mode

#### (1) Input Switch Control Register (ISC)

The ISC2 and ISC3 bits in the ISC register are used in the LIN/UART module (RLIN3).
Setting bit 2 or bit 3 to 1 selects the input signal of the serial data input pin for the LIN/UART module as the external interrupt input.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Address: F0073H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ISC | 0 | 0 | 0 | 0 | ISC3 | ISC2 | 0 | ISC0 |

| ISC3 | Switching inputs for external interrupt INTP12 |
|---|---|
| 0 | INTP12 pin input signal is set as external interrupt input. |
| 1 | LRXD1 pin input signal is set as external interrupt input. |

| ISC2 | Switching inputs for external interrupt INTP11 |
|---|---|
| 0 | INTP11 pin input signal is set as external interrupt input. |
| 1 | LRXD0 pin input signal is set as external interrupt input. |

| ISC0 | Switching inputs for external interrupt INTP0 |
|---|---|
| 0 | INTP0 pin input signal is set as external interrupt input. (normal operation) |
| 1 | RXD0 pin input signal is set as external interrupt input. (wake-up signal detection) |

**Caution Bits 7 to 4 and 1 should always be set to 0.**

### (2) LIN Channel Select Register (LCHSEL)

Address: F007BH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | LSEL0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | LSEL0 | LIN Channel Select | 0: Selects LIN0. (LIN0 registers can be accessed.)<br>1: Selects LIN1. (LIN1 registers can be accessed.) | R/W |
| 7 to 1 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

**LSEL0 bit (LIN channel select bit)**

Since the LIN/UART module registers are not directly mapped on the CPU memory map, they should be accessed via the register windows. The register windows are mapped on addresses F06C1H to F06E9H.

Setting a value to the LSEL0 bit maps all the registers of the corresponding channel on the register window.

Setting the LSEL0 bit to 0 maps the LIN0 registers.

Setting the LSEL0 bit to 1 maps the LIN1 registers.

With the product incorporating one channel, set the LSEL0 bit to 0.

With the product incorporating two channels, set the LSEL0 bit to the applicable value before accessing a register of the channel to use.

### (3) Peripheral Enable Register 2 (PER2)

The PER2 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use the peripheral functions which are controlled by this register, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Address: F02C1H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | <3> | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PER2 | 0 | 0 | 0 | 0 | LIN1EN Note 1 | LIN0EN | 0 | CAN0EN Note 2 |

| LIN1EN Note 1 | Control of LIN1 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• Disables writing to the SFR used by LIN1.<br>• LIN1 is in the reset state. |
| 1 | Enables input clock supply.<br>• Enables reading from and writing to the SFR used by LIN1. |

| LIN0EN | Control of LIN0 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• Disables writing to the SFR used by LIN0.<br>• LIN0 is in the reset state. |
| 1 | Enables input clock supply.<br>• Enables reading from and writing to the SFR used by LIN0. |

| CAN0EN Note 2 | Control of CAN input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• Disables writing to the SFR used by CAN.<br>• CAN is in the reset state. |
| 1 | Enables input clock supply.<br>• Enables reading from and writing to the SFR used by CAN. |

**Notes 1.** Only in the RL78/F14 products with at least 128 Kbytes of code flash memory and the 100-pin products of the RL78/F14.

**2.** Only in the RL78/F13 (CAN and LIN incorporated) and RL78/F14 products.

**Caution Be sure to clear the following bits to 0.**

**Bits 0, 1, 3, 4, 5, 6, and 7 in the RL78/F13 (LIN incorporated) products**

**Bits 1, 3, 4, 5, 6, and 7 in the RL78/F13 (CAN and LIN incorporated) products and the RL78/F14 products with 30, 32, 48, 64, or 80 pins and up to 96 Kbytes of code flash memory**

**Bits 1, 4, 5, 6, and 7 in the RL78/F14 products with at least 128 Kbytes of code flash memory and the 100-pin products of the RL78/F14**

### (4) LIN Clock Select Register (LINCKSEL)

This register is used to control the communication clock source supplied to the LIN.

Address: F02C3H After reset: 00H R/W

| Symbol | 7 | 6 | <5> | <4> | 3 | 2 | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| LINCKSEL | 0 | 0 | LIN1MCKE Note | LIN0MCKE | 0 | 0 | LIN1MCK Note | LIN0MCK |

| LIN1MCKE Note | Control of supplying or stopping LIN1 communication clock source |
|---|---|
| 0 | Stops LIN communication clock source supply. |
| 1 | Enables LIN communication clock source supply. |

| LIN0MCKE | Control of supplying or stopping LIN0 communication clock source |
|---|---|
| 0 | Stops LIN communication clock source supply. |
| 1 | Enables LIN communication clock source supply. |

| LIN1MCK Note | Control of selecting LIN1 communication clock supply |
|---|---|
| 0 | Selects the $f_{CLK}$ clock. |
| 1 | Selects the $f_{MX}$ clock. |

| LIN0MCK | Control of selecting LIN0 communication clock source |
|---|---|
| 0 | Selects the $f_{CLK}$ clock. |
| 1 | Selects the $f_{MX}$ clock. |

**Note** Only in the RL78/F14 products with at least 48 pins and 128 Kbytes or more of code flash memory and the 100-pin products of the RL78/F14.

**Cautions 1. Select the LINn operating clock with the LINnMCK bit before setting the LINnMCKE (n = 0, 1) bit to 1 (operating clock is supplied).**

**2. When operating LINn in SNOOZE mode, set the LINnMCK bit to 0.**

<R> **3. In case of LINnMCK is set to 1, do not used the timeout error detection.**
**In that case, set at least 1.2 times the frequency of the LIN communication clock source to the $f_{CLK}$ clock.**

**(5) External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)**

For details, see **21.3.4 External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1).**

### (6) LIN Wake-up Baud Rate Select Register (LWBRn)

Address: F06C1H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NSPB[3:0] | | | | LPRS[2:0] | | | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 3 to 1 | LPRS [2:0] | Prescaler Clock Select | b3 b1<br>0 0 0: 1/1<br>0 0 1: 1/2<br>0 1 0: 1/4<br>0 1 1: 1/8<br>1 0 0: 1/16<br>1 0 1: 1/32<br>1 1 0: 1/64<br>1 1 1: 1/128 | R/W |
| 7 to 4 | NSPB [3:0] | Bit Sampling Count Select | b7 b4<br>0 0 0 0: 16 sampling<br>0 0 1 1: 4 sampling<br>0 1 1 1: 8 sampling<br>1 1 1 1: 16 sampling<br>Settings other than the above are prohibited. | R/W |

Set the LWBRn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**LPRS[2:0] bits (prescaler clock select bits)**

The LPRS bits select the frequency division ratio for the prescaler.

The LIN communication clock source frequency is divided based on this prescaler.

In LIN slave mode with auto baud rate (LIN/UART mode select bits in LIN/UART mode register = 10b), set these bits according to the target baud rate so that the frequency of the prescaler clock is the corresponding value from the list.

| [Target baud rate] | [Frequency of prescaler clock] |
|---|---|
| 1 kbps to 20 kbps: | 4 MHz **Note** |
| 1 kbps to less than 2.4 kbps: | 4 MHz |
| 2.4 kbps to 20 kbps: | 8 MHz to 12 MHz |

**Note** Set the NSPB bits to 0011b (4 sampling).

**NSPB[3:0] bits (bit sampling count select bits)**

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate).

In LIN slave mode with auto baud rate (LIN/UART mode select bits in LIN/UART mode register = 10b), set the NSPB bits to 0011b (4 sampling) or 0111b (8 sampling).

In LIN slave mode with fixed baud rate (LIN/UART mode select bits in LIN/UART mode register = 11b), set the NSPB bits to 0000b or 1111b (16 sampling).

### (7) LIN/UART Baud Rate Prescaler Register (LBRPn)

Address: F06C3H, F06C2H

| | LBRPn1 | | | | | | | | LBRPn0 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | Setting Range | R/W |
|---|---|---|---|
| 15 to 0 | Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. | 0000H to FFFFH | R/W |

Set the LBRPn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) in the LWBRn register by L + 1.
The LBRPn register can be accessed in 8-bit units using the following registers.
- Lower 8 bits: LIN/UART baud rate prescaler 0 register (LBRPn0); address F06C2H
- Upper 8 bits: LIN/UART baud rate prescaler 1 register (LBRPn1); address F06C3H

**Remark** When a sync field reception succeeded in LIN slave mode [auto baud rate], baud rate correction result is set to LBRPn register automatically.

### (8) LIN Self-Test Control Register (LSTCn)

Address: F06C4H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | LSTM |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 7 to 0 | Writing A7H, 58H, and 01H successively to these bits places the module into LIN self-test mode. | | | R/W |
| 0 | LSTM | LIN Self-Test Mode | 0: The module is not in LIN self-test mode<br>1: The module is in LIN self-test mode. | R/W |

The LSTCn register cancels protection of LIN self-test mode.
Set the LSTCn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
Writing A7H, 58H, and 01H successively to the LSTCn register places the module into LIN self-test mode.
When successive writing is completed thus placing LIN self-test mode to be entered, the LSTM bit is set to 1.
Do not write any other value during successive writing.
For making transition to LIN self-test mode, refer to **17.6 LIN Self-Test Mode**.
Reading bits 6 to 1 returns 000000b, and reading bit 7 returns the undefined value.

**LSTM bit (LIN self-test mode bit)**
When transition to LIN self-test mode is completed, the LSTM bit is set to 1.
For leaving LIN self-test mode, refer to **17.6 LIN Self-Test Mode**.
Writing 1 to this bit does not affect the value of the LSTCn register if it is not a part of successive writing of A7H, 58H, and 01H.

### (9) LIN/UART Mode Register (LMDn)

Address: F06C8H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | LRDNFS | LIOS | — | — | LMD[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 1, 0 | LMD[1:0] | LIN/UART Mode Select | b1 b0<br>1 0: LIN slave mode (auto baud rate)<br>1 1: LIN slave mode (fixed baud rate) | R/W |
| 3, 2 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |
| 4 | LIOS | LIN Interrupt Output Select | 0: LIN interrupt is used.<br>1: Transmission interrupt, successful reception interrupt, and reception status interrupt are used. | R/W |
| 5 | LRDNFS | LIN Reception Data Noise Filtering Disable | 0: The noise filter is enabled.<br>1: The noise filter is disabled. | R/W |
| 6 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |

Set the LMDn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**LMD[1:0] bits (LIN/UART mode select bits)**
The LMD bits select the LIN/UART module mode.
To use the LIN/UART module as a LIN slave, set these bits to 10b or 11b.
With 10b set, the LIN/UART module operates in LIN slave mode with auto baud rate.
With 11b set, the LIN/UART module operates in LIN slave mode with fixed baud rate.

**LIOS bit (LIN interrupt output select bit)**
The LIOS bit selects the number of interrupt outputs from the LIN/UART module.
With 0 set, the LIN interrupt is generated from the LIN/UART module.
With 1 set, the transmission interrupt, successful reception interrupt, and reception status interrupt are generated from the LIN/UART module.
For each interrupt source, refer to **17.9 Interrupts**.

**LRDNFS bit (LIN reception data noise filtering disable bit)**
The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

### (10) LIN Break Field Configuration Register/UART Configuration Register (LBFCn)

Address: F06C9H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | BLT |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | BLT | Reception Break (Low) Width Select | 0: Reception break (low width) of 9.5/10 or more Tbits is detected.<br>1: Reception break (low width) of 10.5/11 or more Tbits is detected. | R/W |
| 6 to 1 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |

Set the LBFCn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**BLT bit (reception break (low) width select bit)**
The BLT bit sets the critical low width of the received data to be determined as break.

In LIN slave mode with auto baud rate (the LMD bits in the LMDn register are 10b):
With 0 set, the low width of 10 or more Tbits is detected.
With 1 set, the low width of 11 or more Tbits is detected.

In LIN slave mode with fixed baud rate (the LMD bits in the LMDn register are 11b):
With 0 set, the low width of 9.5 or more Tbits is detected.
With 1 set, the low width of 10.5 or more Tbits is detected.

### (11) LIN/UART Space Configuration Register (LSCn)

Address: F06CAH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | IBS[1:0] | | — | RS[2:0] | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 2 to 0 | RS[2:0] | Response Space Select | b2 b0<br>0 0 0: 0 Tbit<br>0 0 1: 1 Tbit<br>0 1 0: 2 Tbits<br>0 1 1: 3 Tbits<br>1 0 0: 4 Tbits<br>1 0 1: 5 Tbits<br>1 1 0: 6 Tbits<br>1 1 1: 7 Tbits | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 5, 4 | IBS[1:0] | Inter-Byte Space Select | b5 b4<br>0 0: 0 Tbit<br>0 1: 1 Tbit<br>1 0: 2 Tbits<br>1 1: 3 Tbits | R/W |
| 7, 6 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

Set the LSCn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
Setting is enabled only during response transmission; setting is disabled during response reception.
Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

**RS[2:0] bits (response space select bits)**
The RS bits set the width of the response space of the response transmission.
0 Tbit to 7 Tbits can be set.

**IBS[1:0] bits (inter-byte space select bits)**
The IBS bits set the width of the inter-byte space of the response transmission.
0 Tbit to 3 Tbits can be set.

### (12) LIN Wake-up Configuration Register (LWUPn)

Address: F06CBH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WUTL[3:0] | | | | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 3 to 0 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |
| 7 to 4 | WUTL[3:0] | Wake-up Transmission Low Width Select | b7 b4<br>0 0 0 0: 1 Tbit<br>0 0 0 1: 2 Tbits<br>0 0 1 0: 3 Tbits<br>0 0 1 1: 4 Tbits<br>:<br>1 1 0 0: 13 Tbits<br>1 1 0 1: 14 Tbits<br>1 1 1 0: 15 Tbits<br>1 1 1 1: 16 Tbits | R/W |

Set the LWUPn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**WUTL[3:0] bits (wake-up transmission low width select bits)**
The WUTL bits set the low width of the wake-up frame transmission.
1 Tbit to 16 Tbits can be set.

### (13) LIN Interrupt Enable Register (LIEn)

Address: F06CCH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | SHIE | ERRIE | FRCIE | FTCIE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | FTCIE | Successful Frame/Wake-up Transmission Interrupt Enable | 0: Disables successful response/wake-up transmission interrupt.<br>1: Enables successful response/wake-up transmission interrupt. | R/W |
| 1 | FRCIE | Successful Frame/Wake-up Reception Interrupt Enable | 0: Disables successful response/wake-up reception interrupt.<br>1: Enables successful response/wake-up reception interrupt. | R/W |
| 2 | ERRIE | Error Detection Interrupt Enable | 0: Disables error detection interrupt.<br>1: Enables error detection interrupt. | R/W |
| 3 | SHIE | Successful Header Reception Interrupt Enable | 0: Disables successful header reception interrupt.<br>1: Enables successful header reception interrupt. | R/W |
| 7 to 4 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

Set the LIEn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**FTCIE bit (successful frame/wake-up transmission interrupt enable bit)**
The FTCIE bit enables or disables interrupt generation upon successful transmission of a response or a wake-up signal.
With 0 set, the interrupt is not generated when the FTC flag in the LSTn register is set to 1.
With 1 set, the interrupt is generated when the FTC flag in the LSTn register is set to 1.

**FRCIE bit (successful frame/wake-up reception interrupt enable bit)**
The FRCIE bit enables or disables interrupt generation upon successful reception of a response or a wake-up signal (counting of low width of the input signal).
With 0 set, the interrupt is not generated when the FRC flag in the LSTn register is set to 1.
With 1 set, the interrupt is generated when the FRC flag in the LSTn register is set to 1.

**ERRIE bit (error detection interrupt enable bit)**
The ERRIE bit enables or disables interrupt generation upon detection of an error.
With 0 set, the interrupt is not generated when the ERR flag in the LSTn register is set to 1.
With 1 set, the interrupt is generated when the ERR flag in the LSTn register is set to 1.
Interrupt sources can be the bit error, frame/response timeout error, framing error, sync field error, checksum error, ID parity error, and response preparation error.
Detection of the bit error, frame/response timeout error, framing error, sync field error, and ID parity error can be enabled or disabled using the LEDEn register.

**SHIE bit (successful header reception interrupt enable bit)**
The SHIE bit enables or disables interrupt generation upon successful transmission of a header.
With 0 set, the interrupt is not generated when the HTRC flag in the LSTn register is set to 1.
With 1 set, the interrupt is generated when the HTRC flag in the LSTn register is set to 1.

### (14) LIN/UART Error Detection Enable Register (LEDEn)

Address: F06CDH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LTES | IPERE | — | SFERE | FERE | TERE | — | BERE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | BERE | Bit Error Detection Enable | 0: Disables bit error detection.<br>1: Enables bit error detection. | R/W |
| 1 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 2 | TERE | Timeout Error Detection Enable | 0: Disables frame/response timeout error detection.<br>1: Enables frame/response timeout error detection. | R/W |
| 3 | FERE | Framing Error Detection Enable | 0: Disables framing error detection.<br>1: Enables framing error detection. | R/W |
| 4 | SFERE | Sync Field Error Detection Enable | 0: Disables sync field error detection.<br>1: Enables sync field error detection. | R/W |
| 5 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 6 | IPERE | ID Parity Error Detection Enable | 0: Disables ID parity error detection.<br>1: Enables ID parity error detection. | R/W |
| 7 | LTES | Timeout Error Select | 0: Frame timeout error<br>1: Response timeout error | R/W |

Set the LEDEn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**BERE bit (bit error detection enable bit)**

The BERE bit enables or disables detection of the bit error.

<R> Set 1(the bit error detection enables) to this bit.

The bit error detection result of the bit error is indicated in the BER flag in the LESTn register.

For details of the bit error, refer to **17.4.6 Error Status**.

**TERE bit (timeout error detection enable bit)**

The TER bit enables or disables detection of the frame timeout error or the response timeout error.

With 0 set, the frame timeout error or response timeout error is not detected.

With 1 set, the frame timeout error or response timeout error is detected.

When this bit is set to 1, the detection result is indicated in the TER flag in the LESTn register.

With the LTES bit, either the frame timeout error or response timeout error can be selected.

Do not use the timeout error in LIN slave mode with auto baud rate (LIN/UART mode select bits in LIN/UART mode register = 10b).

Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.

For details of the timeout error, refer to **17.4.6 Error Status**.

**FERE bit (framing error detection enable bit)**

The FERE bit enables or disables detection of the framing error.

<R> Set 1(the framing error detection enables) to this bit.

The framing error detection result is indicated in the FER flag in the LESTn register.

For details of the framing error, refer to **17.4.6 Error Status**.

**SFERE bit (sync field error detection enable bit)**

The SFERE bit enables or disables detection of the sync field error.
With 0 set, the sync field error is not detected.
With 1 set, the sync field error is detected.

Upon detection of the sync field error, the system is placed in the next header wait state, irrespective of the setting of this bit.

When this bit is set to 1, the detection result is indicated in the SFER flag in the LESTn register.
For details of the sync field error, refer to **17.4.6 Error Status**.

**IPERE bit (ID parity error detection enable bit)**

The IPERE bit enables or disables detection of the ID parity error.
With 0 set, the ID parity error is not detected.
With 1 set, the ID parity error is detected.
When this bit is set to 1, the detection result is indicated in the IPER flag in the LESTn register.
For details of the ID parity error, refer to **17.4.6 Error Status**.

**LTES bit (timeout error select bit)**

The LTES bit selects the specific timeout function to be used.
With 0 set, the timeout function applies to frame timeout.
With 1 set, the timeout function applies to response timeout.
For details of the timeout error, refer to **17.4.6 Error Status**.

### (15) LIN/UART Control Register (LCUCn)

Address: F06CEH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | OM1 | OM0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | OM0 | LIN Reset | 0: LIN reset mode is caused.<br>1: LIN reset mode is canceled. | R/W |
| 1 | OM1 | LIN Mode Select | 0: LIN wake-up mode is caused.<br>1: LIN operation mode is caused. | R/W |
| 7 to 2 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

Set the LCUCn register to 01H to cause a transition to LIN wake-up mode after canceling LIN reset mode, and set the LCUCn register to 03H to cause a transition to LIN operation mode.
In LIN self-test mode, set the LCUCn register to 03H after a transition to LIN self-test mode is completed.
When the LIN/UART module makes a transition from the LIN operating mode to the LIN reset mode while operating as a LIN slave (at a fixed baud rate), write 1 to the LIN0EN bit (or the LIN1EN bit) in the PER2 register after having cleared the given bit to 0.
After a value is written to this register, confirm that the value written is actually indicated in the LMSTn register before writing another value.

**OM0 bit (LIN reset bit)**
The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.
With 0 set, LIN reset mode is caused.
With 1 set, LIN reset mode is canceled.

**OM1 bit (LIN mode select bit)**
The OM1 bit selects the specific operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.
With 0 set, LIN wake-up mode is caused.
With 1 set, LIN operation mode is caused.
This register is valid only when the OMM0 bit in the LMSTn register is 1.
Writing a value to this bit is disabled while the FTS bit in the LTRCn register is 1.

### (16) LIN/UART Transmission Control Register (LTRCn)

Address: F06D0H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | LNRR | RTS | FTS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | FTS | LIN Communication Start | 0: Header reception or wake-up transmission/reception is stopped.<br>1: Header reception or wake-up transmission/reception is started. | R/W |
| 1 | RTS | Response Transmission/Reception Start | 0: Response transmission/reception is stopped.<br>1: Response transmission/reception is started. | R/W |
| 2 | LNRR | No-Response Request | 0: Response to the reception ID is received/transmitted.<br>1: Response to the reception ID is not received/transmitted. | R/W |
| 7 to 3 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

<R>

**FTS bit (LIN communication start bit)**

Set the FTS bit to 1 to start header or wake-up reception (counting of the low width of the input signal).
Also set this bit to 1 to allow wake-up transmission.
Only 1 can be written to this bit; 0 cannot be written.
Writing a value to this bit is disabled when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
This bit is set to 0 upon completion of wake-up transmission/reception and transition to LIN reset mode.

**RTS bit (response transmission/reception start bit)**

Set the RTS bit to 1 when response transmission/reception is started after the header is received and the received ID is checked. Once set, this bit is automatically cleared to 0 upon completion of response communication or transition to LIN reset mode.
Only 1 can be written to this bit; 0 cannot be written.
To write 1 to this bit, write 02H to the LTRCn register by using an 8-bit data transfer instruction.
Do not set this bit and the LNRR bit to 1 simultaneously.
Writing a value to this bit is disabled when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
Writing a value to this bit is disabled when the FTS bit is 0 (header reception or wake-up transmission/reception is stopped).

When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 upon completion of data group transmission/reception or transition to LIN reset mode.

**LNRR bit (no-response request bit)**

Set the LNRR bit to 1 when neither response transmission nor reception is started after the header is received and the received ID is checked.

Once set, this bit is automatically cleared to 0 upon detection of the new sync field or transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 04H to the LTRCn register by using an 8-bit data transfer instruction.

Do not set this bit and the RTS bit to 1 simultaneously.

Writing a value to this bit is disabled when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (header reception or wake-up transmission/reception is stopped).

When a 9-byte or longer response is to be transmitted or received, do not use this bit other than on completion of header reception (do not use this bit on completion of the second and subsequent data groups).

### (17) LIN/UART Mode Status Register (LMSTn)

Address: F06D1H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | OMM1 | OMM0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | OMM0 | LIN Reset Status Monitor | 0: The module is in LIN reset mode.<br>1: The module is not in LIN reset mode. | R |
| 1 | OMM1 | LIN Mode Status Monitor | 0: The module is in LIN wake-up mode.<br>1: The module is in LIN operation mode. | R |
| 7 to 2 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

**OMM0 bit (LIN reset status monitor)**

**OMM1 bit (LIN mode status monitor)**

The OMM0 and OMM1 bits indicate the current operation mode.

### (18) LIN/UART Status Register (LSTn)

Address: F06D2H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | HTRC | D1RC | — | — | ERR | — | FRC | FTC |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | FTC | Successful Frame/Wake-up Transmission Flag | 0: Response or wake-up transmission has not been completed.<br>1: Response or wake-up transmission has been completed. | R/W |
| 1 | FRC | Successful Frame/Wake-up Reception Flag | 0: Response or wake-up reception has not been completed.<br>1: Response or wake-up reception has been completed. | R/W |
| 2 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 3 | ERR | Error Detection Flag | 0: No error has been detected.<br>1: Error has been detected. | R |
| 4, 5 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |
| 6 | D1RC | Successful Data 1 Reception Flag | 0: Data 1 reception has not been completed.<br>1: Data 1 reception has been completed. | R/W |
| 7 | HTRC | Successful Header Reception Flag | 0: Header reception has not been completed.<br>1: Header reception has been completed. | R/W |

The LSTn register is automatically cleared to 00H upon transition to LIN reset mode.
In LIN reset mode, writing to this register is disabled. In LIN reset mode, the register retains 00H.
To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits by using an 8-bit data transfer instruction.

**FTC flag (successful frame/wake-up transmission flag)**
Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FTC flag is set to 1 upon completion of response or wake-up transmission. Here, an interrupt is generated if the FTCIE bit in the LIEn register is 1 (interrupt is enabled). Note that when the response or wake-up transmission is completed with the FTC flag set to 1, an interrupt is not generated. To clear the bit to 0, write 0 to the bit.

When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

**FRC flag (successful frame/wake-up reception flag)**
Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FRC flag is set to 1 upon completion of response or wake-up reception. Here, an interrupt is generated if the FRCIE bit in the LIEn register is 1 (interrupt is enabled). Note that when the response or wake-up reception is completed with the FRC flag set to 1, an interrupt is not generated. To clear the bit to 0, write 0 to the bit.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

**ERR flag (error detection flag)**

The ERR flag is set to 1 upon detection of an error (any of the LESTn register flags is 1). Here, an interrupt is generated if the ERRIE bit in the LIEn register is 1 (interrupt is enabled). Note that when an error is detected with the ERR flag set to 1, an interrupt is not generated. To clear the bit to 0, write 0 to the RPER, IPER ,CSER, SFER, FER, TER, and BER flags in the LESTn register. This clears the ERR flag to 0.

**D1RC flag (successful data 1 reception flag)**

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value that has been retained before 1 is written. The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt is not generated. To clear the bit to 0, write 0 to the bit.

When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

**HTRC flag (successful header transmission flag)**

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written. The HTRC flag is set to 1 upon completion of header transmission. Here, an interrupt is generated if the SHIE bit in the LIEn register is 1 (interrupt is enabled). Note that when header reception is completed with the HTRC flag set to 1, an interrupt is not generated. To clear the bit to 0, write 0 to the bit.
After the reception of a header, clear this bit after reading it as 1 so that a new header will be detectable.

### (19) LIN/UART Error Status Register (LESTn)

Address: F06D3H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RPER | IPER | CSER | SFER | FER | TER | — | BER |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | BER | Bit Error Flag | 0: Bit error has not been detected.<br>1: Bit error has been detected. | R/W |
| 1 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 2 | TER | Timeout Error Flag | 0: Frame/response timeout error has not been detected.<br>1: Frame/response timeout error has been detected. | R/W |
| 3 | FER | Framing Error Flag | 0: Framing error has not been detected.<br>1: Framing error has been detected. | R/W |
| 4 | SFER | Sync Field Error Flag | 0: Sync field error has not been detected.<br>1: Sync field error has been detected. | R/W |
| 5 | CSER | Checksum Error Flag | 0: Checksum error has not been detected.<br>1: Checksum error has been detected. | R/W |
| 6 | IPER | ID Parity Error Flag | 0: ID parity error has not been detected.<br>1: ID parity error has been detected. | R/W |
| 7 | RPER | Response Preparation Error Flag | 0: Response preparation error has not been detected.<br>1: Response preparation error has been detected. | R/W |

The LESTn register is automatically cleared to 00H upon transition to LIN reset mode.
In LIN reset mode, writing to this register is disabled. In LIN reset mode, the register retains 00H.
To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits by using an 8-bit data transfer instruction.

**BER flag (bit error flag)**
Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The BER flag is set to 1 upon bit error detection if the BERE bit in the LEDEn register is 1 (bit error detection is enabled). To clear the bit to 0, write 0 to the bit.

**TER flag (timeout error flag)**
Only 0 can be written to the TER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The TER flag is set to 1 upon frame timeout error or response timeout error detection if the TERE bit in the LEDEn register is 1 (frame/response timeout error detection is enabled). To clear the bit to 0, write 0 to the bit.

**FER flag (framing error flag)**
Only 0 can be written to the FER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FER flag is set to 1 upon framing error detection if the FERE bit in the LEDEn register is 1 (framing error detection is enabled). To clear the bit to 0, write 0 to the bit.

**SFER flag (sync field error flag)**
Only 0 can be written to the SFER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The SFER flag is set to 1 upon sync field error detection if the SFERE bit in the LEDEn register is 1 (sync field error detection is enabled). To clear the bit to 0, write 0 to the bit.

**CSER flag (checksum error flag)**

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value that has been retained before 1 is written. The CSER flag is set to 1 upon checksum error detection. To clear the bit to 0, write 0 to the bit.

**IPER flag (ID parity error flag)**

Only 0 can be written to the IPER flag; when 1 is written, the bit retains the value that has been retained before 1 is written. The IPER flag is set to 1 upon ID parity error detection if the IPERE bit in the LEDEn register is 1 (ID parity error detection is enabled). To clear the bit to 0, write 0 to the bit.

**RPER flag (response preparation error flag)**

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value that has been retained before 1 is written. The RPER flag is set to 1 upon response preparation error detection. To clear the bit to 0, write 0 to the bit.

### (20) LIN/UART Data Field Configuration Register (LDFCn)

Address: F06D4H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LSS | — | LCS | RCDS | RFDL[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 3 to 0 | RFDL[3:0] | Response Field Length Select | b3 b0<br>0 0 0 0: 0 byte (+ checksum)<br>0 0 0 1: 1 byte (+ checksum)<br>0 0 1 0: 2 bytes (+ checksum)<br>:<br>0 1 1 1: 7 bytes (+ checksum)<br>1 0 0 0: 8 bytes (+ checksum)<br>Settings other than the above are prohibited. | R/W |
| 4 | RCDS | Response Field Communication Direction Select | 0: Reception<br>1: Transmission | R/W |
| 5 | LCS | Checksum Select | 0: Classic checksum mode<br>1: Enhanced checksum mode | R/W |
| 6 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 7 | LSS | Transmission/Reception Continuation Select | 0: The data group to be transmitted/received next is the last one.<br>1: The data group to be transmitted/received next is not the last one.<br>(Data transmission/reception is continued without waiting for the next header reception.) | R/W |

**RFDL[3:0] bits (response field length select bits)**
The RFDL bits set the length of the response field data.
The data length can be 0 to 8 bytes excluding the checksum size.
Set these bits when the RTS bit is 0 (response transmission/reception stopped).

<R> When response data of 9 bytes or more is to be transmitted and received, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

**RCDS bit (response field communication direction select bit)**
The RCDS bit sets the direction of the response field/wake-up signal communication.
With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (low width of the input signal is counted).
With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.
When the module is in the LIN operating mode, set this bit while the RTS bit in the LTRCn register is 0 (response transmission/reception stopped). When the module is in the LIN wakeup mode, set this bit while the FTS bit in the LTRCn register is 0 (header reception or wake-up transmission/reception stopped).

When response data of 9 bytes or more is to be transmitted or received, do not change the RCDS bit setting after the first data group through the last data group.

**LCS bit (checksum select bit)**
The LCS bit sets the checksum mode.
With 0 set, classic checksum mode is selected.
With 1 set, enhanced checksum mode is selected.
When the timeout error is used (the TERE bit in the LEDEn register is 1), the specific timeout time depends on the setting of this bit. For details, refer to **17.4.6 Error Status**.
Do not set this bit to 1 (enhanced mode) when the response field is 0 bytes long (the RFDL bit is 0).

When response of 9 bytes or more is to be transmitted or received, do not change the LCS bit setting after the first data group through the last data group.
During communication of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.
Set this bit while the RTS bit in the LTRCn register is 0 (response transmission/reception stopped).

**LSS bit (transmission/reception continuation select bit)**
The LSS bit shows that the next data group to be transmitted or received is not the last one.
With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one.
With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.

During LIN communication, do not set this bit to 1.
Set this bit while the RTS bit in the LTRCn register is 0 (response transmission/reception stopped).

### (21) LIN/UART ID Buffer Register (LIDBn)

Address: F06D5H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IDP[1:0] | | ID[5:0] | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 5 to 0 | ID[5:0] | ID | Holds the 6-bit ID value received in the ID field. | R/W |
| 7, 6 | IDP[1:0] | Parity | Holds the parity bits (P) received in the ID field. | R/W |

Writing to the LIDBn register is enabled upon completion of header reception.
In LIN mode (LIN operation mode or LIN wake-up mode), writing is disabled.

In LIN self-test mode, this register operates as follows:
Write the value to be transmitted before communication. The reversed value of the value received can be read from the register after frame transmission/reception is completed (after loopback).
For details of LIN self-test mode, refer to **17.6 LIN Self-Test Mode**.

**ID[5:0] bits (ID bits)**
The ID bits hold the 6-bit ID value received in the ID field of the LIN frame.

**IDP[1:0] bits (parity bits)**
The IDP bits hold the parity bits (P0 and P1) received in the ID field of the LIN frame.
The IDP0 bit is P0 and the IDP1 bit is P1.
When the IPERE bit in the LEDEn register is 1 (ID parity detection is enabled), the received value is checked against the internally pre-calculated value, and if they do not agree, the IPER bit (ID parity error flag) is set.

### (22) LIN Checksum Buffer Register (LCBRn)

Address: F06D6H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | R/W |
|---|---|---|
| 7 to 0 | Holds the checksum value transmitted or received. | R/W |

In LIN mode, this register operates as follows:

- When the RCDS bit in the LDFCn register is 1 (transmission):

  The value transmitted can be read from the register. Writing to this register is invalid.
- When the RCDS bit in the LDFCn register is 0 (reception):

  The value received can be read from the register. Writing to this register is invalid.

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for the other data groups.

In LIN self-test mode, this register operates as follows:

- When the RCDS bit in the LDFCn register is 1 (transmission):

  The reversed value of the value received can be read from the register after frame transmission is completed (after loopback).
- When the RCDS bit in the LDFCn register is 0 (reception):

  Write the value to be received before communication. The reversed value of the value received can be read from the register after frame transmission/reception is completed (after loopback).

For details of LIN self-test mode, refer to **17.6 LIN Self-Test Mode**.

Set the LCBRn register when the FTS bit in the LTRCn register is 0 (frame transmission or wake-up transmission/reception is halted).

### (23) LIN/UART Data Buffer m Register (LDBnm) (m = 1 to 8)

Address: LDBn1 F06D8H, LDBn2 F06D9H, LDBn3 F06DAH, LDBn4 F06DBH, LDBn5 F06DCH, LDBn6 F06DDH, LDBn7 F06DEH, LDBn8 F06DFH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | Setting Range | R/W |
|---|---|---|---|
| 7 to 0 | Sets the data to be transmitted or allows the received data to be read. | 00H to FFH | R/W |

For response transmission:
These registers set the data to be transmitted in the response field.
Set these registers when the RTS bit is 0 (response reception/transmission is halted).

For response reception:
These registers hold the data received in the response field.
The received data is overwritten. If an error is detected, the data prior to reception interruption is stored in the register.
Do not read these registers when the RTS bit is 1 (response transmission/reception is started)

In LIN self-test mode, this register operates as follows:
Write the value to be transmitted before communication. The reversed value of the value received can be read from the register after frame transmission/reception is completed (after loopback).
For details of LIN self-test mode, refer to **17.6 LIN Self-Test Mode**.

### 17.2.3 Registers for UART

#### (1) Input Switch Control Register (ISC)

The ISC2 and ISC3 bits in the ISC register are used in the LIN/UART module (RLIN3).
Setting bit 2 or bit 3 to 1 selects the input signal of the serial data input pin for the LIN/UART module as the external interrupt input.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Address: F0073H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ISC | 0 | 0 | 0 | 0 | ISC3 | ISC2 | 0 | ISC0 |

| ISC3 | Switching inputs for external interrupt INTP12 |
|---|---|
| 0 | INTP12 pin input signal is set as external interrupt input. |
| 1 | LRXD1 pin input signal is set as external interrupt input. |

| ISC2 | Switching inputs for external interrupt INTP11 |
|---|---|
| 0 | INTP11 pin input signal is set as external interrupt input. |
| 1 | LRXD0 pin input signal is set as external interrupt input. |

| ISC0 | Switching inputs for external interrupt INTP0 |
|---|---|
| 0 | INTP0 pin input signal is set as external interrupt input. (normal operation) |
| 1 | RXD0 pin input signal is set as external interrupt input. (wake-up signal detection) |

**Caution Bits 7 to 4 and 1 should always be set to 0.**

### (2) LIN Channel Select Register (LCHSEL)

Address: F007BH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | LSEL0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | LSEL0 | LIN Channel Select | 0: Selects LIN0. (LIN0 registers can be accessed.)<br>1: Selects LIN1. (LIN1 registers can be accessed.) | R/W |
| 7 to 1 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

**LSEL0 bit (LIN channel select bit)**

Since the LIN/UART module registers are not directly mapped on the CPU memory map, they should be accessed via the register windows. The register windows are mapped on addresses F06C1H to F06E9H.

Setting a value to the LSEL0 bit maps all the registers of the corresponding channel on the register window.

Setting the LSEL0 bit to 0 maps the LIN0 registers.

Setting the LSEL0 bit to 1 maps the LIN1 registers.

With the product incorporating one channel, set the LSEL0 bit to 0.

With the product incorporating two channels, set the LSEL0 bit to the applicable value before accessing a register of the channel to use.

### (3) Peripheral Enable Register 2 (PER2)

The PER2 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use the peripheral functions which are controlled by this register, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Address: F02C1H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | <3> | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PER2 | 0 | 0 | 0 | 0 | LIN1EN Note 1 | LIN0EN | 0 | CAN0EN Note 2 |

| LIN1EN Note 1 | Control of LIN1 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• Disables writing to the SFR used by LIN1.<br>• LIN1 is in the reset state. |
| 1 | Enables input clock supply.<br>• Enables reading from and writing to the SFR used by LIN1. |

| LIN0EN | Control of LIN0 input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• Disables writing to the SFR used by LIN0.<br>• LIN0 is in the reset state. |
| 1 | Enables input clock supply.<br>• Enables reading from and writing to the SFR used by LIN0. |

| CAN0EN Note 2 | Control of CAN input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• Disables writing to the SFR used by CAN.<br>• CAN is in the reset state. |
| 1 | Enables input clock supply.<br>• Enables reading from and writing to the SFR used by CAN. |

**Notes 1.** Only in the RL78/F14 products with at least 128 Kbytes of code flash memory and the 100-pin products of the RL78/F14.

**2.** Only in the RL78/F13 (CAN and LIN incorporated) and RL78/F14 products.

**Caution Be sure to clear the following bits to 0.**

**Bits 0, 1, 3, 4, 5, 6, and 7 in the RL78/F13 (LIN incorporated) products**

**Bits 1, 3, 4, 5, 6, and 7 in the RL78/F13 (CAN and LIN incorporated) products and the RL78/F14 products with 30, 32, 48, 64, or 80 pins and up to 96 Kbytes of code flash memory**

**Bits 1, 4, 5, 6, and 7 in the RL78/F14 products with at least 128 Kbytes of code flash memory and the 100-pin products of the RL78/F14**

### (4) LIN Clock Select Register (LINCKSEL)

This register is used to control the communication clock source supplied to the LIN.

Address: F02C3H After reset: 00H R/W

| Symbol | 7 | 6 | <5> | <4> | 3 | 2 | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| LINCKSEL | 0 | 0 | LIN1MCKE Note | LIN0MCKE | 0 | 0 | LIN1MCK Note | LIN0MCK |

| LIN1MCKE Note | Control of supplying or stopping LIN1 communication clock source |
|---|---|
| 0 | Stops LIN communication clock source supply. |
| 1 | Enables LIN communication clock source supply. |

| LIN0MCKE | Control of supplying or stopping LIN0 communication clock source |
|---|---|
| 0 | Stops LIN communication clock source supply. |
| 1 | Enables LIN engine clock supply. |

| LIN1MCK Note | Control of selecting LIN1 communication clock source |
|---|---|
| 0 | Selects the $f_{CLK}$ clock. |
| 1 | Selects the $f_{MX}$ clock. |

| LIN0MCK | Control of selecting LIN0 communication clock source |
|---|---|
| 0 | Selects the $f_{CLK}$ clock. |
| 1 | Selects the $f_{MX}$ clock. |

**Note** Only in the RL78/F14 products with at least 48 pins and 128 Kbytes or more of code flash memory and the 100-pin products of the RL78/F14.

**Cautions**
1. **Select the LINn operating clock with the LINnMCK bit before setting the LINnMCKE (n = 0, 1) bit to 1 (operating clock is supplied).**
2. **When operating LINn in SNOOZE mode, set the LINnMCK bit to 0.**
3. <R> **In case of LINnMCK is set to 1, set at least 1.2 times the frequency of the LIN communication clock source to the $f_{CLK}$ clock.**

**(5) External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)**

For details, see **21.3.4 External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)**.

### (6) LIN Wake-up Baud Rate Select Register (LWBRn)

Address: F06C1H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NSPB[3:0] | | | | LPRS[2:0] | | | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 3 to 1 | LPRS [2:0] | Prescaler Clock Select | b3 b1<br>0 0 0: 1/1<br>0 0 1: 1/2<br>0 1 0: 1/4<br>0 1 1: 1/8<br>1 0 0: 1/16<br>1 0 1: 1/32<br>1 1 0: 1/64<br>1 1 1: 1/128 | R/W |
| 7 to 4 | NSPB [3:0] | Bit Sampling Count Select | b7 b4<br>0 0 0 0: 16 sampling<br>0 1 0 1: 6 sampling<br>0 1 1 0: 7 sampling<br>0 1 1 1: 8 sampling<br>1 0 0 0: 9 sampling<br>1 0 0 1: 10 sampling<br>1 0 1 0: 11 sampling<br>1 0 1 1: 12 sampling<br>1 1 0 0: 13 sampling<br>1 1 0 1: 14 sampling<br>1 1 1 0: 15 sampling<br>1 1 1 1: 16 sampling<br>Settings other than the above are prohibited. | R/W |

Set the LWBRn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**LPRS[2:0] bits (prescaler clock select bits)**
The LPRS bits select the frequency division ratio for the prescaler.
The LIN communication clock source frequency is divided based on this prescaler.

**NSPB[3:0] bits (bit sampling count select bits)**
The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate).
In UART mode (LIN/UART mode select bits in LIN/UART mode register = 01b), the NSPB bits can be set for 6 to 16 sampling.

### (7) LIN/UART Baud Rate Prescaler Register (LBRPn)

Address: F06C3H, F06C2H

| | LBRPn1 | | | | | | | | LBRPn0 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | Setting Range | R/W |
|---|---|---|---|
| 15 to 0 | Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. | 0000H to FFFFH | R/W |

Set the LBRPn register when the OMM0 bit in the LMSTn register is 0 ( LIN reset mode).
Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits in the LWBRn register (prescaler clock select bits) by L + 1.
The LBRPn register can be accessed in 8-bit units using the following registers.

- Lower 8 bits: LIN/UART baud rate prescaler 0 register (LBRPn0); address F06C2H
- Upper 8 bits: LIN/UART baud rate prescaler 1 register (LBRPn1); address F06C3H

### (8) UART Standby Control Register (LUSCn)

Address: F06C5H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | URDCC | USEC | UWC |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | UWC | UART Standby Wake-up Control | 0: Disables start of reception from STOP mode.<br>1: Enables start of reception from STOP mode. | R/W |
| 1 | USEC | UART Standby Error Control | 0: Enables error detection interrupt generation.<br>1: Disables error detection interrupt generation. | R/W |
| 2 | URDCC | UART Standby Received data Comparison Control | 0: Disables comparison of the received data and the LIDBn register value in SNOOZE mode.<br>1: Enables comparison of the received data and the LIDBn register value in SNOOZE mode. | R/W |
| 7 to 3 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

Set the LUSCn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**UWC bit (UART standby wake-up control bit)**

The UWC bit enables or disables transition to SNOOZE mode upon detection of the falling edge on the reception pin in STOP mode.

With 0 set, detection of the falling edge on the reception pin in STOP mode does not cause a transition to SNOOZE mode thus not initiating reception.

With 1 set, detection of the falling edge on the reception pin in STOP mode causes a transition to SNOOZE mode thus initiating reception.

**USEC bit (UART standby error control bit)**

The USEC bit enables or disables interrupt generation upon detection of an error or change in status in SNOOZE mode.

With 0 set, if an error (framing error or parity error) or change in status (detection of the expansion bit) is detected in SNOOZE mode, the corresponding flag is set to 1 thus generating the error detection interrupt.

With 1 set, if an error (framing error or parity error) or change in status (detection of the expansion bit) is detected in SNOOZE mode, the corresponding flag is not set to 1 thus generating no error detection interrupt and the module makes a transition to STOP mode.

Do not set this bit to 1 (error detection interrupt generation is disabled) when the UWC bit is 0 (start of reception from STOP mode is disabled).

This bit is enabled when the UWC bit is set to 1 (start of reception from STOP mode is enabled).

**URDCC bit (UART standby received data comparison control bit)**

The URDCC bit enables or disables comparison of the data received in SNOOZE mode and the LIDBn register value.

With 0 set, the data received in SNOOZE mode is not compared with the LIDBn register value and the appropriate interrupt is generated.

With 1 set, the data received in SNOOZE mode is compared with the LIDBn register value, and if they agree, the successful reception interrupt is generated. If they do not agree, an interrupt is not generated but the module makes a transition to STOP mode.

Do not set this bit to 1 (comparison of the received data and the LIDBn register value in SNOOZE mode is enabled) when the UWC bit is 0 (start of reception from STOP mode is disabled).

When this bit should be set to 1 (comparison of the received data and the LIDBn register value in SNOOZE mode is enabled), be sure to set the bit length to 8 bits (the UBLS bit in the LBFCn register is 0; 8-bit UART communication) and set the UEBE bit in the LUORn1 register to 0; expansion bit operation is disabled).

This bit is enabled when the UWC bit is set to 1 (start of reception from STOP mode is enabled).

### (9) LIN/UART Mode Register (LMDn)

Address: F06C8H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | LRDNFS | — | — | — | LMD[1:0] | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 1, 0 | LMD[1:0] | LIN/UART Mode Select | b1 b0<br>0 1: UART mode | R/W |
| 4 to 2 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |
| 5 | LRDNFS | LIN Reception Data Noise Filtering Disable | 0: The noise filter is enabled.<br>1: The noise filter is disabled. | R/W |
| 6 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |

Set the LMDn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**LMD[1:0] bits (LIN/UART mode select bits)**
The LMD bits select the LIN/UART module mode.
To use the LIN/UART module as UART, set these bits to 01b.
With 01b set, the LIN/UART module operates as UART.

**LRDNFS bit (LIN reception data noise filtering disable bit)**
The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

### (10) LIN Break Field Configuration Register/UART Configuration Register (LBFCn)

Address: F06C9H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | UTPS | URPS | UPS[1:0] | | USBLS | UBOS | UBLS |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | UBLS | UART Character Length Select | 0: 8-bit UART communication<br>1: 7-bit UART communication | R/W |
| 1 | UBOS | UART Transfer Format Select | 0: LSB first<br>1: MSB first | R/W |
| 2 | USBLS | UART Stop Bit Length Select | 0: One stop bit<br>1: Two stop bits | R/W |
| 4, 3 | UPS[1:0] | UART Parity Select | b4 b3<br>0 0: No parity<br>0 1: Even parity<br>1 0: 0-parity<br>1 1: Odd parity | R/W |
| 5 | URPS | UART Input Polarity Select | 0: Reception data is input as is.<br>1: Reception data is reversed before being input. | R/W |
| 6 | UTPS | UART Output Polarity Select | 0: Transmission data is output as is.<br>1: Transmission data is reversed before being output. | R/W |
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |

Set the LBFCn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**UBLS bit (UART character length select bit)**
The UBLS bit sets the length of a character of a UART communication frame.
With 0 set, a character of a frame is 8 bits long in communication.
With 1 set, a character of a frame is 7 bits long in communication.
Setting this bit is invalid when a character of a UART frame for communication is 9 bits long (the UEBE bit in the LUORn1 register is 1).

**UBOS bit (UART transfer format select bit)**
The UBOS bit sets the bit order of UART communication data.
With 0 set, data is transferred with the LSB first.
With 1 set, data is transferred with the MSB first.

**USBLS bit (UART stop bit length select bit)**
The USBLS bit sets the stop bit length in UART communication.
With 0 set, transmission is performed with 1 stop bit.
With 1 set, transmission is performed with 2 stop bits.

**UPS[1:0] bit (UART parity select bit)**

The UPS bits set the parity for UART communication.
With 00b set, no parity is used in communication.

- Transmission
  A parity bit is not appended to transmission data.
- Reception
  A parity bit is not appended to reception data, thus causing no parity error.

With 01b set, an even parity is used in communication.

- Transmission
  When the number of 1s in transmission data is odd, 1 is appended as the parity bit, whereas when the number of 1s in the transmission data is even, 0 is appended as the parity bit.
- Reception
  When the number of 1s in the received data including the parity bit is odd, the parity error occurs.

With 10b set, 0-parity is used in communication.

- Transmission
  A 0 is appended as the parity bit irrespective of the number of 1s in the transmission data.
- Reception
  No parity error is caused since the value of the parity bit is not checked.

With 11b set, an odd parity is used in communication.

- Transmission
  When the number of 1s in transmission data is odd, 0 is appended as the parity bit, whereas when the number of 1s in the transmission data is even, 1 is appended as the parity bit.
- Reception
  When the number of 1s in the received data including the parity bit is even, the parity error occurs.

**URPS bit (UART input polarity select bit)**

The URPS bit sets the input polarity in UART communication.
With 0 set, the reception data is input as is.
With 1 set, the reception data is reversed before being input.
The setting of this bit applies to all the bits for the UART frames.
In half-duplex communication, set this bit and the UTPS bit to the same value.

**UTPS bit (UART output polarity select bit)**

The UTPS bit sets the output polarity in UART communication.
With 0 set, the transmission data is output as is.
With 1 set, the transmission data is reversed before being output.
The setting of this bit applies to all the bits for the UART frames.
In half-duplex communication, set this bit and the URPS bit to the same value.

### (11) LIN/UART Space Configuration Register (LSCn)

Address: F06CAH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | IBS[1:0] | | — | — | — | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 2 to 0 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |
| 3 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 5, 4 | IBS[1:0] | Inter-Byte Space Select | b5 b4<br>0 0: 0 Tbit<br>0 1: 1 Tbit<br>1 0: 2 Tbits<br>1 1: 3 Tbits | R/W |
| 7, 6 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

Set the LSCn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**IBS[1:0] bits (Inter-byte space select bits)**

The IBS bits set the width of the space between UART frames in transmission using UART buffer.

0 Tbit to 3 Tbits can be set.

<R> When transmitting from the transmission buffer(LUTDRn register) and the wait transmission buffer(LUWTDRn register),set 00b to the IBS[1:0] bit.

### (12) LIN/UART Error Detection Enable Register (LEDEn)

Address: F06CDH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | FERE | OERE | — | BERE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | BERE | Bit Error Detection Enable | 0: Disables bit error detection.<br>1: Enables bit error detection. | R/W |
| 1 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 2 | OERE | Overrun Error Detection Enable | 0: Disables overrun error detection.<br>1: Enables overrun error detection. | R/W |
| 3 | FERE | Framing Error Detection Enable | 0: Disables framing error detection.<br>1: Enables framing error detection. | R/W |
| 4 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 5 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 6 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |

Set the LEDEn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

**BERE bit (bit error detection enable bit)**
The BERE bit enables or disables detection of the bit error.
With 0 set, the bit error is not detected.
With 1 set, the bit error is detected.
When this bit is set to 1, the detection result is indicated in the BER flag in the LESTn register.
Do not set this bit to 1 when the LIN/UART module is used in full-duplex mode.
For details of the bit error, refer to **17.5.5 Error Status**.
Do not set this bit when the NSPB bits in the LWBRn register are 0101b (6 sampling) and the LRDNFS bit in the LMDn register is 0 (the noise filter is in use).

**OERE bit (overrun error detection enable bit)**
The OERE bit enables or disables detection of the overrun error.
With 0 set, the overrun error is not detected.
With 1 set, the overrun error is detected.
When this bit is set to 1, the detection result is indicated in the OER flag in the LESTn register.
For details of the overrun error, refer to **17.5.5 Error Status**.

**FERE bit (framing error detection enable bit)**
The FERE bit enables or disables detection of the framing error.
With 0 set, the framing error is not detected.
With 1 set, the framing error is detected.
When this bit is set to 1, the detection result is indicated in the FER flag in the LESTn register.
For details of the framing error, refer to **17.5.5 Error Status**.

### (13) LIN/UART Control Register (LCUCn)

Address: F06CEH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | OM0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | OM0 | LIN Reset | 0: LIN reset mode is caused.<br>1: LIN reset mode is canceled. | R/W |
| 1 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 7 to 2 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

After a value is written to this register, confirm that the value written is actually indicated in the LMSTn register before writing another value.

**OM0 bit (LIN reset bit)**
The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.
With 0 set, LIN reset mode is caused.
With 1 set, LIN reset mode is canceled.

### (14) LIN/UART Transmission Control Register (LTRCn)

Address: F06D0H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | RTS | — |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 1 | RTS | UART Buffer Transmission Start | 0: UART buffer transmission is disabled.<br>1: UART buffer transmission is enabled. | R/W |
| 2 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 7 to 3 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

**RTS bit (UART buffer transmission start bit)**

Set the RTS bit to 1 to transmit data from UART buffer.

Only 1 can be written to this bit; 0 cannot be written.

Write a value to this bit when the UTOE bit in the LUOERn register is 1 (transmission is enabled) and the UTS bit in the LSTn register is 0 (transmission not in progress).

Once set, whether or not an error has occurred, this bit is automatically cleared to 0 upon completion of transmission of the amount of data corresponding to the setting for the number of data units (by the MDL bits in the LDFCn register).

This bit is automatically cleared to 0 upon transition to LIN reset mode.

Writing a value to this bit is disabled when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

When 1 is to be written to this bit while the setting of the UTSW bit in the LDFCn register is 1 (requesting transmission from the UART buffer after the completion of waiting for stop bit reception), only do so after the reception of a stop bit.

### (15) LIN/UART Mode Status Register (LMSTn)

Address: F06D1H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | OMM0 |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | OMM0 | LIN Reset Status Monitor | 0: The module is in LIN reset mode.<br>1: The module is not in LIN reset mode. | R |
| 1 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R |
| 7 to 2 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

**OMM0 bit (LIN reset status monitor bit)**

The OMM0 bit indicates whether or not the LIN reset mode is currently set.
With 0 set, the LIN/UART module is in LIN reset mode.
With 1 set, the LIN/UART module is not in LIN reset mode.

### (16) LIN/UART Status Register (LSTn)

Address: F06D2H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | URS | UTS | ERR | — | — | FTC |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | FTC | Successful Buffer Transmission Flag | 0: Data transmission from the UART buffer has not been completed.<br>1: Data transmission from the UART buffer has been completed. | R/W |
| 1 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 2 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 3 | ERR | Error Detection Flag | 0: No error has been detected.<br>1: Error has been detected. | R |
| 4 | UTS | Transmission Status Flag | 0: The LIN/UART module is not transmitting data.<br>1: The LIN/UART module is transmitting data. | R |
| 5 | URS | Reception Status Flag | 0: The LIN/UART module is not receiving data.<br>1: The LIN/UART module is receiving data. | R |
| 7, 6 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

The LSTn register is automatically cleared to 00H upon transition to LIN reset mode.
In LIN reset mode, writing to this register is disabled. In LIN reset mode, the register retains 00H.
To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits by using an 8-bit data transfer instruction.

<R> **FTC flag (successful buffer transmission flag)**

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
Whether or not an error has occurred, this bit is set to 1 upon completion of transmission of data, which is equal to the number of data units set with the MDL bits in the LDFCn register, from UART buffer. Here, an interrupt is generated.
To clear the bit to 0, write 0 to the bit.

**ERR flag (error detection flag)**

The ERR flag is set to 1 upon detection of an error (any of the LESTn register flags is 1). Here, an interrupt is generated.
Note that when an error, the expansion bit, or ID match is detected with the ERR flag set to 1, an interrupt is not generated.
To clear the bit to 0, write 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the LESTn register. This clears the ERR flag to 0.

**UTS flag (transmission status flag)**

The UTS flag is set to 1 upon start of transmission. During transmission, the flag retains 1.
Transmission is started when:

- transmission data is set in the LUTDRn or LUWTDRn register.
- 1 is set in the RTS bit in the LTRCn register.

The UTS flag is cleared to 0 upon end of transmission. While transmission is halted, the flag retains 0.
Transmission is ended when:

- transmission of the data set in the LUTDRn or LUWTDRn register is completed and the next transmission data is not set.
- data transmission from the UART buffer is completed (the RTS bit in the LTRCn register is set to 0 ).

**URS flag (reception status flag)**

The URS flag is set to 1 upon start of reception. During reception, the flag retains 1.
Reception is started when:

- the start bit is detected.

The URS flag is cleared to 0 upon end of reception. While reception is halted, the flag retains 0.
Reception is ended when:

- the first stop bit is sampled.

### (17) LIN/UART Error Status Register (LESTn)

Address: F06D3H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | UPER | IDMT | EXBT | FER | OER | — | BER |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | BER | Bit Error Flag | 0: Bit error has not been detected.<br>1: Bit error has been detected. | R/W |
| 1 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 2 | OER | Overrun Error Flag | 0: Overrun error has not been detected.<br>1: Overrun error has been detected. | R/W |
| 3 | FER | Framing Error Flag | 0: Framing error has not been detected.<br>1: Framing error has been detected. | R/W |
| 4 | EXBT | Expansion Bit Detection Flag | 0: Expansion bit has not been detected.<br>1: Expansion bit has been detected. | R/W |
| 5 | IDMT | ID match flag | 0: Received data does not agree with the ID value.<br>1: Received data agrees with the ID value. | R/W |
| 6 | UPER | Parity Error Flag | 0: Parity error has not been detected.<br>1: Parity error has been detected. | R/W |
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |

The LESTn register is automatically cleared to 00H upon transition to LIN reset mode.
In LIN reset mode, writing to this register is disabled. In LIN reset mode, the register retains 00H.
To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits by using an 8-bit data transfer instruction.

**BER flag (bit error flag)**
Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The BER flag is set to 1 upon bit error detection if the BERE bit in the LEDEn register is 1 (bit error detection is enabled). To clear the bit to 0, write 0 to the bit.

**OER flag (overrun error flag)**
Only 0 can be written to the OER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The OER flag is set to 1 upon overrun error detection if the OERE bit in the LEDEn register is 1 (overrun error detection is enabled). To clear the bit to 0, write 0 to the bit.

**FER flag (framing error flag)**
Only 0 can be written to the FER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FER flag is set to 1 upon framing error detection if the FERE bit in the LEDEn register is 1 (framing error detection is enabled).
In SNOOZE mode, the following conditions should also be satisfied to set this flag to 1.

- The UWC bit in the LUSCn register is 1 (start of reception from STOP mode is enabled).
- The USEC bit in the LUSCn register is 0 (error detection interrupt generation is enabled).

To clear the bit to 0, write 0 to the bit.

**EXBT flag (expansion bit detection flag)**

Only 0 can be written to the EXBT flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
This flag is set to 1 when the UEBE bit in the LUORn1 register is 1 (expansion bit operation is enabled) and the received expansion bit agrees with the UEBDL bit value in the LUORn1 register.
In SNOOZE mode, the following conditions should also be satisfied to set this flag to 1.

- The UWC bit in the LUSCn register is 1 (start of reception from STOP mode is enabled).
- The USEC bit in the LUSCn register is 0 (error detection interrupt generation is enabled).
- The UECD bit in the LUORn1 register is 0 (comparison of the expansion bit is enabled).

To clear the bit to 0, write 0 to the bit.

**IDMT flag (ID match flag)**

Only 0 can be written to the IDMT flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The IDMT flag is set to 1 when all the following conditions are satisfied.

- The UEBE bit in the LUORn1 register is 1 (expansion bit operation is enabled).
- The UECD bit in the LUORn1 register is 0 (expansion bit comparison is enabled).
- The UEBDCE bit in the LUORn1 register is 1 (data comparison after expansion bit detection is enabled).
- The received expansion bit agrees with the value of the UEBDL bit in the LUORn1 register.
- The 8-bit received data excluding the expansion bits agrees with the LIDBn register value.

To clear the bit to 0, write 0 to the bit.

**UPER flag (parity error flag)**

Only 0 can be written to the IDMT flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The UPER flag is set to 1 upon parity error detection.
In SNOOZE mode, the following conditions should also be satisfied to set this flag to 1.

- The UWC bit in the LUSCn register is 1 (start of reception from STOP mode is enabled).
- The USEC bit in the LUSCn register is 0 (error detection interrupt generation is enabled).

To clear the bit to 0, write 0 to the bit.

### (18) LIN/UART Data Field Configuration Register (LDFCn)

Address: F06D4H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | UTSW | — | MDL[3:0] | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 3 to 0 | MDL[3:0] | UART Buffer Data Length Select | b3 b0<br>0 0 0 0: 9 data<br>0 0 0 1: 1 data<br>0 0 1 0: 2 data<br>0 0 1 1: 3 data<br>0 1 0 0: 4 data<br>0 1 0 1: 5 data<br>0 1 1 0: 6 data<br>0 1 1 1: 7 data<br>1 0 0 0: 8 data<br>1 0 0 1: 9 data<br>Settings other than the above are prohibited. | R/W |
| 4 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R/W |
| 5 | UTSW | Transmission Start Wait | 0: When UART buffer transmission is requested, transmission is started immediately.<br>1: When UART buffer transmission is requested, transmission is started after waiting for stop bit reception to be completed. | R/W |
| 7, 6 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

**MDL[3:0] bits (UART buffer data length select bits)**

The MDL bits set the data length of the UART buffer.

Writing a value to these bits is disabled when the RTS bit is 1 (UART buffer transmission is enabled).

**UTSW bit (transmission start wait bit)**

The UTSW bit controls the start timing of UART buffer transmission.

With 0 set, when UART buffer transmission is requested, transmission is started immediately.

With 1 set, when UART buffer transmission is requested, transmission is started after waiting for reception of the stop bit to be completed.

Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the LBFCn register.

This bit is enabled when 1 is set to the RTS bit in the LTRCn register.

Writing a value to this bit is disabled when the RTS bit is 1 (UART buffer transmission is enabled).

Do not set this bit to 1 for the purpose other than switching from reception to transmission in half-duplex communication.

### (19) LIN/UART ID Buffer Register (LIDBn)

Address: F06D5H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | R/W |
|---|---|---|
| 7 to 0 | Sets the ID value to be referenced for expansion bit/data comparison or data comparison in SNOOZE mode. | R/W |

When the UEBE bit is 1 (expansion bit operation is enabled) and the UEBDCE bit is 1 (data comparison after expansion bit detection is enabled) in the LUORn1 register, set the value to be compared with the received data to the LIDBn register. When the UWC bit is 1 (start of reception from STOP mode is enabled) and the URDCC bit is 1 (comparison of the received data and the LIDBn register value is enabled in SNOOZE mode) in the LUSCn register, set the value to be compared with the received data to the LIDBn register.

Set the LIDBn register when the URS bit in the LSTn register is 0 (reception not in progress).

### (20) UART Data Buffer 0 Register (LUDBn0)

Address: F06D7H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | Setting Range | R/W |
|---|---|---|---|
| 7 to 0 | Sets the data to be transmitted from the UART buffer. | 00H to FFH | R/W |

The LUDBn0 register sets the data to be first transmitted from the UART buffer for nine-data-long transmission (the MDL bits in the LDFCn register is 0H or 9H).

Write to the LUDBn0 register while the RTS bit is 0 (UART buffer transmission is disabled).

For details of the UART buffer, refer to **17.5,3 Buffer Processing of Transmission Data**.

### (21) LIN/UART Data Buffer m Register (LDBnm) (m = 1 to 8)

Address: LDBn1 F06D8H, LDBn2 F06D9H, LDBn3 F06DAH, LDBn4 F06DBH, LDBn5 F06DCH, LDBn6 F06DDH, LDBn7 F06DEH, LDBn8 F06DFH

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | Setting Range | R/W |
|---|---|---|---|
| 7 to 0 | Sets the data to be transmitted from the UART buffer. | 00H to FFH | R/W |

These registers set the data to be transmitted from the UART buffer.
Write to these registers while the RTS bit is 0 (UART buffer transmission is disabled).

For details of the UART buffer, refer to **17.5,3 Buffer Processing of Transmission Data**.

### (22) UART Operation Enable Register (LUOERn)

Address: F06E0H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | UROE | UTOE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | UTOE | Transmission Enable | 0: Disables transmission.<br>1: Enables transmission. | R/W |
| 1 | UROE | Reception Enable | 0: Disables reception.<br>1: Enables reception. | R/W |
| 7 to 2 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

The LUOERn register is automatically cleared to 00H upon transition to LIN reset mode.
In LIN reset mode, writing to this register is disabled. In LIN reset mode, the register retains 00H.

**UTOE bit (transmission enable bit)**
The UTOE bit enables or disables transmission.
With 0 set, transmission is disabled.
With 1 set, transmission is enabled.
Do not clear this bit during transmission. To cancel transfer while transmission is in progress, place the module in the LIN reset mode by setting the OM0 bit in the LCUCn register to 0 (LIN reset mode). Note that this operation also cancels reception.

**UROE bit (reception enable bit)**
The UROE bit enables or disables reception.
With 0 set, reception is disabled.
With 1 set, reception is enabled.
Do not clear this bit during reception. To cancel transfer while reception is in progress, place the module in the LIN reset mode by setting the OM0 bit in the LCUCn register to 0 (LIN reset mode). Note that this operation also cancels transmission.
<R> This bit must be 0 during transmitting data from UART buffer.

### (23) UART Option Register 1 (LUORn1)

Address: F06E1H

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | UECD | UTIGTS | UEBDCE | UEBDL | UEBE |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Function | R/W |
|---|---|---|---|---|
| 0 | UEBE | Expansion Bit Enable | 0: Disables expansion bit operation.<br>1: Enables expansion bit operation. | R/W |
| 1 | UEBDL | Expansion Bit Detection Level Select | 0: Selects expansion bit value 0 as the expansion bit detection level.<br>1: Selects expansion bit value 1 as the expansion bit detection level. | R/W |
| 2 | UEBDCE | Expansion Bit/Data Comparison Enable | 0: Disables data comparison after detection of the expansion bit.<br>1: Enables data comparison after detection of the expansion bit. | R/W |
| 3 | UTIGTS | Transmission Interrupt Generation Timing Select | 0: Transmission interrupt is generated at the start of transmission.<br>1: Transmission interrupt is generated at the completion of transmission. | R/W |
| 4 | UECD | Expansion Bit Comparison Disable | 0: Enables comparison between the received expansion bit and the UEBDL bit value.<br>1: Disables comparison between the received expansion bit and the UEBDL bit value. | R/W |
| 7 to 5 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R/W |

**UEBE bit (expansion bit enable bit)**

The UEBE bit enables or disables expansion bit operation.
With 0 set, expansion bit operation is disabled.
With 1 set, expansion bit operation is enabled.
Set this bit when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
Do not set this bit to 1 when the UART buffer is in use.

**UEBDL bit (expansion bit detection level select bit)**

The UEBDL bit sets the level to be detected as the expansion bit when the UEBE bit is 1 (expansion bit operation is enabled) and the UECD bit is 0 (comparison of the expansion bit is enabled).
With 0 set, expansion bit value 0 is the level to be detected as the expansion bit.
With 1 set, expansion bit value 1 is the level to be detected as the expansion bit.
Set this bit when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
Do not set this bit to 1 when the UART buffer is in use.

**UEBDCE bit (expansion bit/data comparison enable bit)**

The UEBDCE bit enables or disables comparison between the 8-bit received data excluding the expansion bits and the value of LIDBn register after detection of the expansion bit.

With 0 set, comparison between the received data in the LURDRn register and the LIDBn register value is disabled after detection of the expansion bit value selected by the UEBDL bit as the expansion bit.

With 1 set, comparison between the received data in the LURDRn register and the LIDBn register value is enabled after detection of the expansion bit value selected by the UEBDL bit as the expansion bit.

Set this bit when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

Do not set this bit to 1 with the UEBE bit set to 0 (expansion bit operation disabled).

Do not set this bit to 1 with the UECD bit set to 1 (expansion bit comparison disabled).

Do not set this bit to 1 when the UART buffer is in use.

Do not set this bit to 1 with the UWC bit in the LUSCn register set to 1 (start of reception from STOP mode enabled).

**UTIGTS bit (transmission interrupt generation timing select bit)**

The UTIGTS bit sets the generation timing of the transmission interrupt.

With 0 set, the transmission interrupt is generated at the start of transmission.

With 1 set, the transmission interrupt is generated at the completion of transmission.

When transmission from the UART buffer is performed with 0 set, the transmission interrupt is generated only at the start of the transmission of the last data of the data length set with the MDL bits in the LDFCn register.

When transmission from the UART buffer is performed with 1 set, the transmission interrupt is generated only at the completion of the transmission of the last data of the data length set with the MDL bits in the LDFCn register.

**UECD bit (expansion bit comparison disable bit)**

The UECD bit enables or disables comparison between the received expansion bit and the UEBDL bit value when the UEBE bit is 1 (expansion bit operation is enabled).

With 0 set, comparison between the received expansion bit and the UEBDL bit value is enabled when the expansion bit is received.

With 1 set, comparison between the received expansion bit and the UEBDL bit value is disabled when the expansion bit is received.

Set this bit when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

Do not set this bit to 1 when the UART buffer is in use.

Do not set this bit to 1 with the UEBDCE bit set to 1 (data comparison after expansion bit detection is enabled).

### (24) UART Transmission Data Register (LUTDRn)

Address: F06E5H, F06E4H

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | [8:0] | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | Setting Range | R/W |
|---|---|---|---|
| 8 to 0 | Sets the data to be transmitted from the transmission buffer. | 000H to 1FFH | R/W |
| 15 to 9 | Reserved.<br>These bits are always read as 0. The write value should always be 0. | — | R/W |

The LUTDRn register sets the data to be transmitted from the transmission data register.
Writing data to this register with the UTOE bit in the LUOERn register set to 1 starts transmission.
This register can be accessed in 8 bits.
In 9-bit communication mode, do not attempt 8-bit access.
Do not write data to this register when data transmission from the UART buffer is in progress.
Also, do not write data to this register when a transmission request is being generated due to write access to the LUWTDRn register.
When transmitting multiple sets of data continuously, do not write another data item to this register before a transmission interrupt is generated.

The table below shows the bit arrangement according to the set transfer format.

| | LUTDRn | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Item | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 7-bit; LSB first | — | — | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 7-bit; MSB first | — | — | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 |
| 8-bit; LSB first | — | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 8-bit; MSB first | — | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 |
| 9-bit; LSB first | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 9-bit; MSB first | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 | Bit 8 |

### (25) UART Reception Data Register (LURDRn)

Address: F06E7H, F06E6H

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | [8:0] | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | Setting Range | R/W |
|---|---|---|---|
| 8 to 0 | Allows the reception data to be read from the reception buffer. | 000H to 1FFH | R |
| 15 to 9 | Reserved.<br>These bits are always read as 0. The write value should always be 0. | — | R/W |

The LURDRn register allows the reception data to be read from the reception data register.
When the UROE bit in the LUOERn register is 1, the reception data is stored in this register and can be read out.
This register is updated at the stop bit in the reception data.
This register is also updated when an error is caused by the parity or stop bit.
This register is not updated upon occurrence of an overrun error if the OERE bit in the LEDEn register is 1 (overrun error detection is enabled). This register is updated upon occurrence of an overrun error if the OERE bit is 0 (overrun error detection is disabled).
Read this register upon occurrence of a reception error (overrun error, framing error, or parity error) if the OERE bit in the LEDEn register is 1 (overrun error detection is enabled). Reading the next data without reading this register causes an overrun error.
This register can be accessed in 8 bits. However, do not access this register in 8-bit units when the expansion bits are used (the UEBE bit in the LUORn1 register is 1 (expansion bit operation enabled)).

The table below shows the bit arrangement according to the set transfer format.

| | LURDRn | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Item | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 7-bit; LSB first | — | — | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 7-bit; MSB first | — | — | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 |
| 8-bit; LSB first | — | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 8-bit; MSB first | — | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 |
| 9-bit; LSB first | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 9-bit; MSB first | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 | Bit 8 |

### (26) UART Wait Transmission Data Register (LUWTDRn)

Address: F06E9H, F06E8H

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | [8:0] | | | | | | | | |
| Value after reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Function | Setting Range | R/W |
|---|---|---|---|
| 8 to 0 | Sets the data to be transmitted from the UART wait transmission data register after waiting for the stop bit reception to be completed. | 000H to 1FFH | R/W |
| 15 to 9 | Reserved.<br>These bits are always read as 0. The write value should always be 0. | — | R/W |

The LUWTDRn register sets the data to be transmitted from the UART wait transmission data register.
Writing data to this register with the UTOE bit in the LUOERn register set to 1 starts transmission.
Use this register only to switch from reception to transmission in half-duplex communication.
The user should write to this register only while the stop bit is received.
Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the LBFCn register.
When this register is read, the LUTDRn register value is actually read.
In 9-bit communication mode, do not attempt 8-bit access.
Do not write data to this register when data transmission from the UART buffer is in progress.

The table below shows the bit arrangement according to the set transfer format.

| | LUWTDRn | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Item | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 7-bit; LSB first | — | — | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 7-bit; MSB first | — | — | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 |
| 8-bit; LSB first | — | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 8-bit; MSB first | — | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 |
| 9-bit; LSB first | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 9-bit; MSB first | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 | Bit 8 |

## 17.3 Modes

The LIN/UART module provides the following four modes, depending upon the specific function to be performed:

- LIN reset mode
- LIN mode

  (LIN master mode/LIN slave mode [auto baud rate]/LIN slave mode [fixed baud rate])
- UART mode
- LIN self-test mode

The supply of clocks to the LIN/UART module is stopped in LIN reset mode, which reduces power consumption.
Figure 17-3 shows mode transitions. Table 17-4 describes mode transition conditions. Table 17-5 lists operations available in each mode.

**Figure 17-3. Mode Transitions**

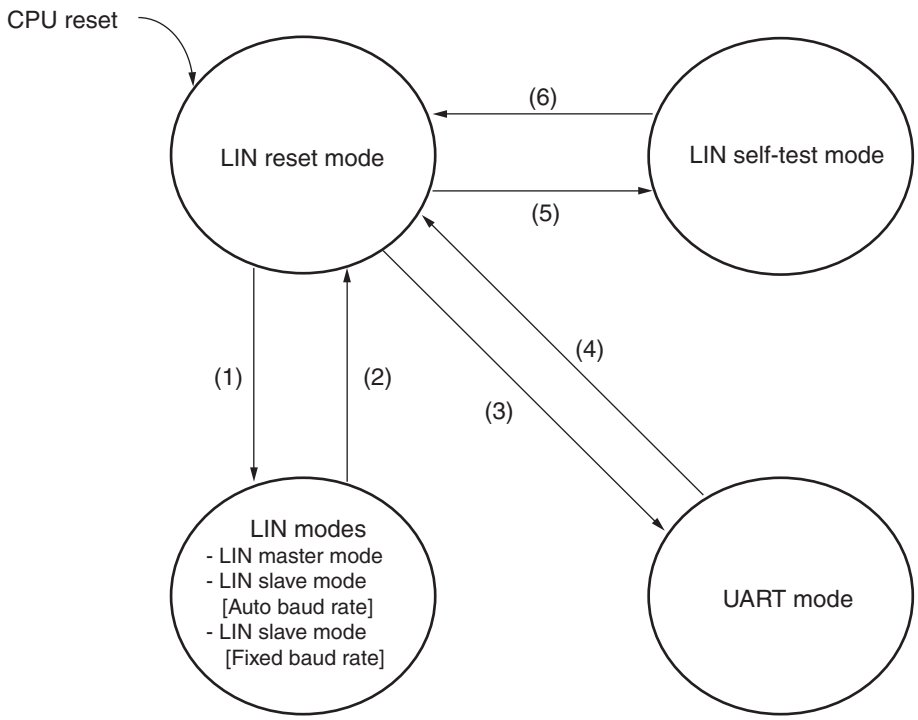


**Table 17-4. Mode Transition Conditions**

| Step | Mode transition | | | Transition condition |
|---|---|---|---|---|
| (1) | LIN reset mode | → | LIN mode<br>(LIN master mode) | LMD bits in LMDn register = 00b<br>and<br>OM1 and OM0 bits in LCUCn register = 01b or 11b |
| | LIN reset mode | → | LIN mode<br>(LIN slave mode[auto baud rate]) | LMD bits in LMDn register = 10b<br>and<br>OM1 and OM0 bits in LCUCn register = 01b or 11b |
| | LIN reset mode | → | LIN mode<br>(LIN slave mode [fixed baud rate]) | LMD bits in LMDn register = 11b<br>and<br>OM1 and OM0 bits in LCUCn register = 01b or 11b |
| (2) | LIN mode | → | LIN reset mode | OM0 bit in LCUCn register = 0b |
| (3) | LIN reset mode | → | UART mode | LMD bits in LMDn register = 01b<br>and<br>OM0 bit in LCUCn register = 1b |
| (4) | UART mode | → | LIN reset mode | OM0 bit in LCUCn register = 0b |
| (5) | LIN reset mode | → | LIN self-test mode | See **17.6 LIN Self-Test Mode**. |
| (6) | LIN self-test mode | → | LIN reset mode | See **17.6 LIN Self-Test Mode**. |

**Table 17-5. Operations Available in Each Mode**

| LIN mode | | UART mode | LIN self-test mode |
|---|---|---|---|
| LIN master mode | LIN slave mode [auto baud rate]/<br>LIN slave mode [fixed baud rate] | | |
| Header transmission<br>Response transmission<br>Response reception<br>Wake-up transmission<br>Wake-up reception<br>Error detection | Header reception<br>Response transmission<br>Response reception<br>Wake-up transmission<br>Wake-up reception<br>Error detection | UART transmission<br>UART reception<br>Error detection | Self test |

Whether a transition has been caused to the LIN reset mode, the LIN mode, or the UART mode can be verified by reading the LMD bits in the LMDn register or the OMM0 bit in the LMSTn register.
The maximum mode transition time (maximum time from when the value is set to the LSUCn register to when the value is indicated in the LMSTn register) is the sum of three CPU clock ($f_{CLK}$) cycles and four cycles of the LIN communication clock source (input clock to the LIN/UART module selected by LINnMCK).
For a description of the LIN self-test mode, see **17.6 LIN Self-Test Mode**.

### 17.3.1 LIN Reset Mode

Setting the OM0 bit in the LCUCn register to 0b (LIN reset mode) causes a transition to LIN reset mode. The change to LIN reset mode can be verified by determining that the OMM0 bit in the LMSTn register has been set to 0b (LIN reset mode). In this mode, the LIN communication and the UART communication functions are all halted, and fLIN also stops.
From LIN reset mode, transitions to LIN mode, UART mode, and LIN self-test mode can be made.

When the mode changes to LIN reset mode, the following registers are initialized to their reset values, and as long as LIN reset mode is in effect, they retain their initial values.

- LTRCn register
- LSTn register
- LESTn register
- LUOERn register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- LCHSEL register
- LWBRn register
- LBRPn0 register
- LBRPn1 register
- LUSCn register
- LMDn register
- LBFCn register
- LSCn register
- LWUPn register
- LIEn register
- LEDEn register
- LDFCn register
- LIDBn register
- LCBRn register
- LUDBn0 register
- LDBnm register (m = 1 to 8)
- LUORn1 register
- LUTDRn register
- LURDRn register
- LUWTDRn register

### 17.3.2 LIN Mode

LIN mode can operate in the following submodes: LIN master mode, LIN slave mode [auto baud rate], and LIN slave mode [fixed baud rate].

In LIN master mode, the following operations can be performed: header transmission, response transmission, response reception, wake-up transmission, wake-up reception, and error detection. In LIN reset mode, setting the LMD bits in the LMDn register to 00b (LIN master mode) and the OM1 and OM0 bits in the LCUCn register to either 01b or 11b sets LIN master mode, turning the OMM1 and OMM0 bits in the LMSTn register to either 01b to 11b.
In LIN slave mode [auto baud rate] and LIN slave mode [fixed baud rate], header reception, response transmission, response reception, wake-up transmission, wake-up reception, and error detection can be performed.
The LIN slave mode [auto baud rate] allows automatic detection of the break field and the sync field, and sets a baud rate based on the results of measurement of a sync field. Operation is possible with baud rates from 1 kbps to 20 kbps. Set the LPRS[2:0] bits in the LWBRn register according to the target baud rate so that the frequency of the clock (prescaler clock) obtained by dividing the LIN communication clock source frequency by the prescaler is the corresponding value from the list.

| [Target baud rate] | [Frequency of prescaler clock] |
|---|---|
| 1 kbps to 20 kbps: | 4 MHz Note |
| 1 kbps to less than 2.4 kbps: | 4 MHz |
| 2.4 kbps to 20 kbps: | 8 MHz to 12 MHz |

**Note** Set the NSPB[3:0] bits in the LWBRn register to 0011b (4 sampling).

In LIN slave mode [fixed baud rate] allows automatic detection of the break field, the sync field, and the ID filed at a baud rate that is set in advance by the baud rate generator.
In LIN reset mode, setting the LMD bits in the LMDn register to 10b (LIN slave mode [auto baud rate] and setting the OM1 and OM0 bit in the LCUCn register to 01b or 11b sets LIN slave mode [auto baud rate]; and setting the LMD bits in the LMDn register to 11b (LIN slave mode [fixed baud rate]), and setting the OM1 and OM0 bits in the LCUCn register to 01b or 11b sets LIN slave mode [fixed baud rate], turning the OMM1 and OMM0 bits in the LMSTn register to 01b or 11b.

When changing a submode to another submode within LIN mode, a transition to LIN reset mode should first be made and change the LMD bits in the LMDn register.

The LIN mode provides the following two operation modes:

- LIN operation mode
- LIN wake-up mode

Figure 17-4 shows the transition of operation modes. Table 17-6 describes the transition conditions of operation modes.

**Figure 17-4. Transition of Operation Modes**

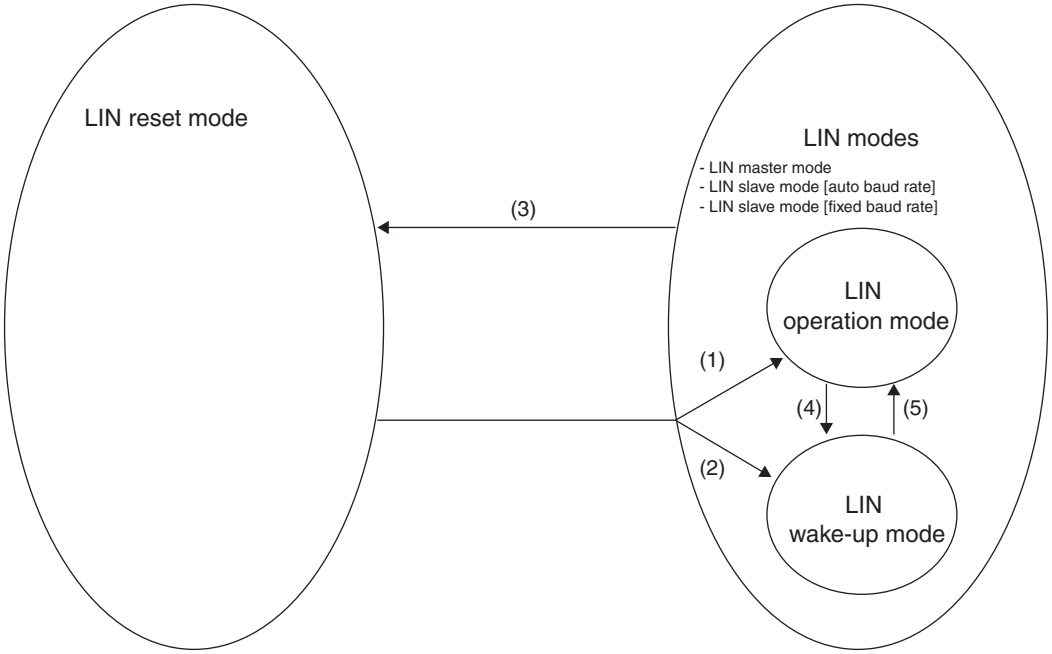



**Table 17-6. Transition Conditions of Operation Modes**

| Step | Operation mode transition | | | Transition condition |
|---|---|---|---|---|
| (1) | LIN reset mode | → | LIN mode<br>-LIN operation mode | LMD bits in LMDn register = 00b or 10b or 11b<br>and<br>OM1 and OM0 bits in LCUCn register = 11b |
| (2) | LIN reset mode | → | LIN mode<br>-LIN wake-up mode | LMD bits in LMDn register = 00b or 10b or 11b<br>and<br>OM1 and OM0 bits in LCUCn register = 01b |
| (3)Note 1 | LIN mode<br>-LIN operation mode<br>-LIN wake-up mode | → | LIN reset mode | OM0 bit in LCUCn register = 0b |
| (4)Note 2 | LIN mode<br>-LIN operation mode | → | LIN mode<br>-LIN wake-up mode | OM1 and OM0 bits in LCUCn register = 01b |
| (5)Note 2 | LIN mode<br>-LIN wake-up mode | → | LIN mode<br>-LIN operation mode | OM1 and OM0 bits in LCUCn register = 11b |

**Notes 1.** When the LIN/UART module makes a transition from the LIN operating mode to the LIN reset mode while operating as a LIN slave (at a fixed baud rate), write 1 to the LIN0EN bit (or the LIN1EN bit) in the PER2 register after having cleared the given bit to 0.

**2.** Transition between LIN operation mode and LIN wake-up mode cannot be made when communication is going on (when the FTS bit in the LTRCn register is 1).

#### (1) LIN Operation Mode

In LIN operation mode, frame processing (header transmission, header reception, response transmission, response reception, and error detection) can be performed.
During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the LCUCn register to 11b changes the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the LMSTn register to 11b. Communication settings should be performed after the LMSTn register has become 11b.

#### (2) LIN Wake-up Mode

In LIN wake-up mode, wake-up signal processing (wake-up transmission, wake-up reception, and error detection) can be performed.
During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the LCUCn register to 01b changes the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the LMSTn register to 01b. Communication settings should be performed after the LMSTn register has become 01b.

### 17.3.3 UART Mode

In LIN reset mode, setting the LMD bits in the LMDn register to 01b (UART mode) and the OM0 bit in the LCUCn register to 1b changes the mode to UART mode, turning the OMM0 bit in the LMSTn register to 1b. Communication settings should be performed after the LMSTn register has become 01b.

### 17.3.4 LIN Self-Test Mode

Writing to the LSTCn register changes the mode to LIN self-test mode. The LSTM bit in the LSTCn register being 1 indicates that the mode has transitioned to the LIN self-test mode.
For further details of operations, see **17.6 LIN Self-Test Mode**.

## 17.4 LIN Mode

### 17.4.1 Operation Overview

#### (1) LIN Master Mode

##### (a) Header Transmission

Figure 17-5 shows the operation of the LIN/UART module (LIN master mode) in header transmission. Table 17-7 provides processing in header transmission.

**Figure 17-5. Operation in Header Transmission**

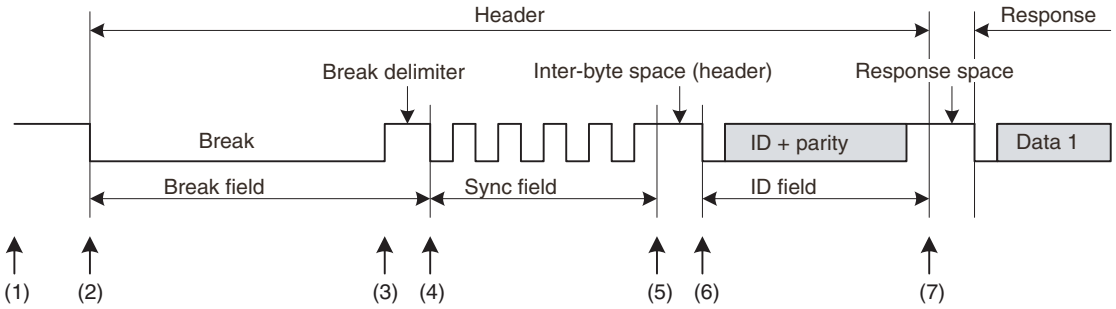



**Table 17-7. Processing in Header Transmission**

| Step | Software processing | LIN/UART module processing |
|---|---|---|
| (1) | • Sets a baud rate<br>• Sets noise filter ON/OFF<br>• Enables interrupt<br>• Enables error detection<br>• Sets frame configuration parameters<br>• Changes the LIN/UART module to the LIN master mode: LIN operation mode<br>• Sets information on the frame to be transmitted (ID, parity, data length, response direction, Checksum method, and transmission data) | Waits for the setting of the FTS bit in the LTRCn register by software (idle). |
| (2) | Sets the FTS bit in the LTRCn register to 1 (start a frame transmission or wake-up transmission/reception). | Transmits a break. |
| (3) | Waits for an interrupt request. | Transmits a break delimiter. |
| (4) | | Transmits a sync field (55H). |
| (5) | | Transmits an inter-byte space (header). |
| (6) | | Transmits an ID field. |
| (7) | | Sets a successful header transmission flag. |

For information about error detection, refer to **17.4.6 Error Status**.

**(b) Response Transmission**

Figure 17-6 shows the operation of the LIN/UART module (LIN master mode) in response transmission. Table 17-8 provides processing in response transmission.

**Figure 17-6. Operation in Response Transmission**

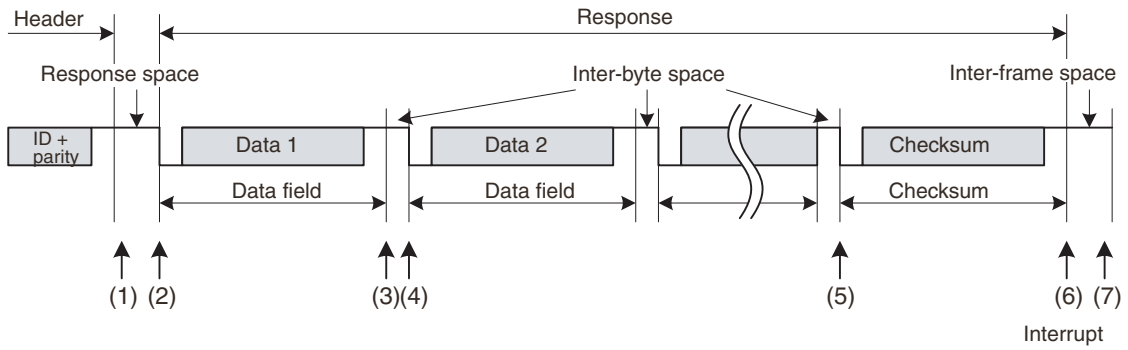



**Table 17-8. Processing in Response Transmission**

| Step | Software processing | LIN/UART module processing |
|---|---|---|
| (1) | (When in frame separate mode)<br>• Sets the RTS bit in the LTRCn register to 1 (response transmission/reception started).<br>(When not in frame separate mode)<br>• Waits for an interrupt request . | (When in frame separate mode)<br>• Waits for the setting of the RTS bit in the LTRCn register to 1 by software.<br>• When the bit is set to 1, sends a response space.<br>(When not in frame separate mode)<br>• Sends a response space. |
| (2) | Waits for an interrupt request. | Transmits the data 1. |
| (3) | | Transmits an inter-byte space. |
| (4) | | • Transmits the data 2.<br>• Transmits an inter-byte space<br>• Transmits the data 3.<br>• Transmits an inter-byte space<br>(Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the LDFCn register, and stops the transmission when the BER flag in the LESTn register is 1 (bit error detected). If an error occurs, does not perform the Checksum transmission in item (5)).<br>: |
| (5) | | Transmits the checksum. |
| (6) | | • Sets a successful frame/wake-up transmission flag.<br>• Sets the FTS bit in the LTRCn register to 0 (frame transmission or wake-up transmission/reception stopped).<br>(When in frame separate mode)<br>• Sets the RTS bit in the LTRCn register to 0 (response transmission/reception stopped). |
| (7) | • Processing after communication Checks the LSTn register and clears flags. | Idle |

For information about error detection, refer to **17.4.6 Error Status**.

### (c) Response Reception

Figure 17-7 shows the operation of the LIN/UART module (LIN master mode) on response reception. Table 17-9 provides processing in response reception.

**Figure 17-7. Operation in Response Reception**

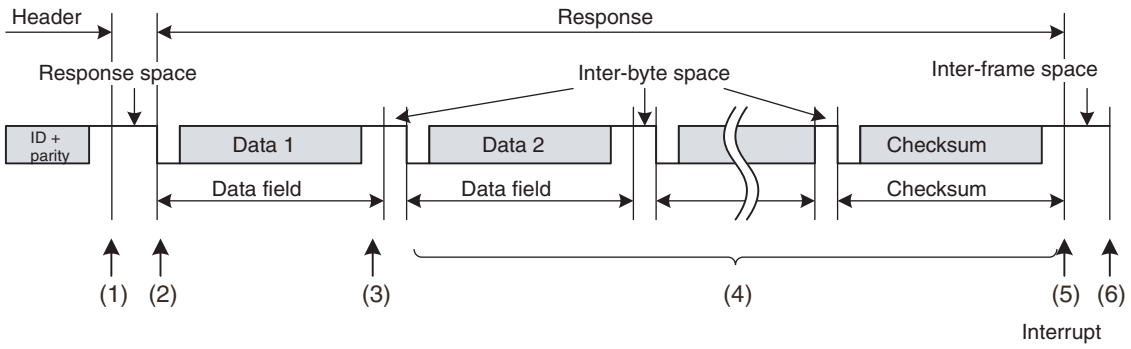



**Table 17-9. Processing in Response Reception**

| Step | Software processing | LIN/UART module processing |
|---|---|---|
| (1) | Waits for an interrupt request (no processing). | Waits for detection of a start bit. |
| (2) | | Receives the data 1 when the start bit is detected. |
| (3) | | Sets the successful data 1 reception flag. |
| (4) | | • Receives the data 2 when the start bit is detected.<br>• Receives the data 3 when the start bit is detected.<br>Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the LDFCn register, and stops the transmission when any bit in the LESTn register is 1 (bit error detected). If an error occurs, the checksum determination in item (5) is not performed).<br>:<br>• Receives the checksum when the start bit is detected. |
| (5) | | • Determines the checksum.<br>• Sets the successful frame/wake-up reception flag.<br>• Sets the FTS bit in the LTRCn register to 0 (frame transmission or wake-up transmission/reception stopped). |
| (6) | • Processing after communication<br>Reads the received data.<br>Checks the LSTn register and clears flags. | Idle |

For information about error detection, refer to **17.4.6 Error Status**.

### (2) LIN Slave Mode

#### (a) Header Reception

Figure 17-8 shows the operation of the LIN/UART module (LIN slave mode) in header reception. Table 17-10 provides processing in header reception.

**Figure 17-8. Operation in Header Reception**

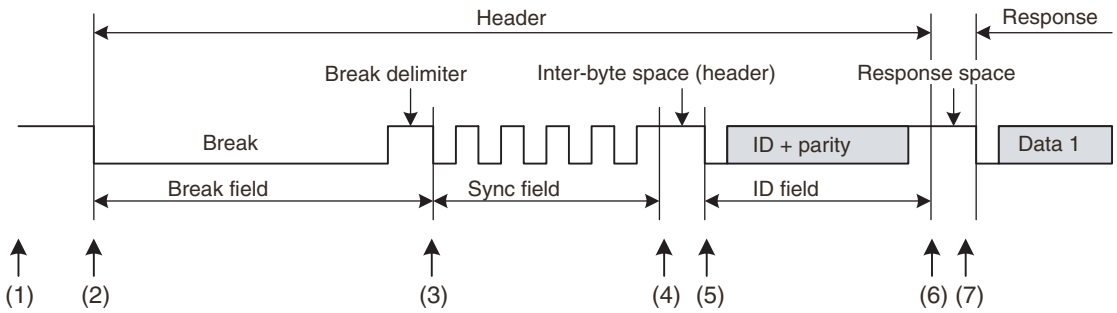



**Table 17-10. Processing in Header Reception**

| Step | Software processing | LIN/UART module processing |
|---|---|---|
| (1) | • Sets a baud rate.<br>• Sets noise filter ON/OFF.<br>• Enables interrupt.<br>• Enables error detection.<br>• Sets frame configuration parameters.<br>• Changes the LIN/UART module to the LIN slave mode: LIN operation mode.<br>• Sets the FTS bit in the LTRCn register to 1 (header reception or wake-up transmission/reception started). | Waits for the setting of the FTS bit in the LTRCn register by software. |
| (2) | Waits for an interrupt request. | Waits for detection of a break field. |
| (3) | | Detects a break field (in the case of LIN slave mode [fixed baud rate]; for break field detection timing in the case of LIN slave mode [auto baud rate], see **(1) Auto Baud Rate Correction Function)**. |
| (4) | | • Detects a sync field (55H)<br>• Sets the baud rate generator (in the case of LIN slave mode [auto baud rate])<br>• Clears the no-response request bit (LNRR bit). |
| (5) | | • Receives an ID field.<br>• Checks an ID parity bit |
| (6) | | Sets a header reception complete flag. |
| (7) | • Checks the LSTn register and clears flags.<br>• Checks the LIDBn register and prepares a response. | • Completes a header reception process.<br>• Waits for a response request. |

n = 0, 1

The LIN/UART module can receive a break field during transmission or reception of a frame. Here, a reception status interrupt might be generated on the detection of a framing error, bit error, etc, at the position of the stop bit of the frame preceding break field reception.

For information about error detection, refer to **17.4.6 Error Status**.

### (1) Auto Baud Rate Correction Function

In LIN slave mode [auto baud rate], the system always measures the low-level widths that are received. If the first “Low” width is 10 times (if the BLT bit in the LBFCn register is “0”) or 11 times (if the BLT bit in the LBFCn register is “1”) or greater calculated from the average of the starting 2 bits (the period of the consecutive fall edges from the beginning of the sync field) of the sync field, the system concludes that the detection of a break field was successful, and verifies that the data in the sync field is 55H. When confirming that the data is 55H and the reception of the sync field is successful, the system automatically sets the baud rate correction results in the LBRPn1 and LBRPn0 registers.
If data is received up to the ID field without error, a successful header reception interrupt is generated at the stop bit position.
If the sync field data is not 55H, the system concludes that the detection of a sync field failed, and sets a sync field error flag, and generates an error detection interrupt.
In such a case, the LIN/UART module waits for the detection of another break field (“Low”) without baud rate correction.

**Figure 17-9. Header Reception in LIN Slave Mode [Auto Baud Rate] (in Normal Operation)**

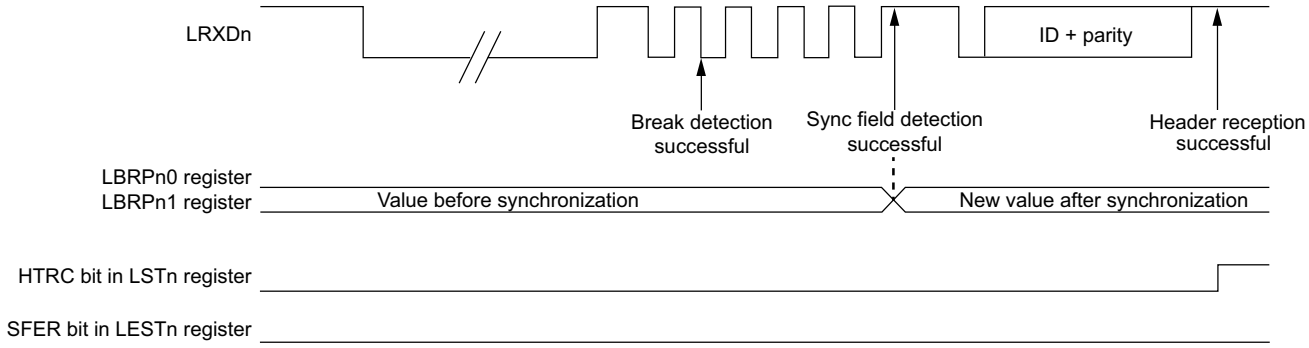
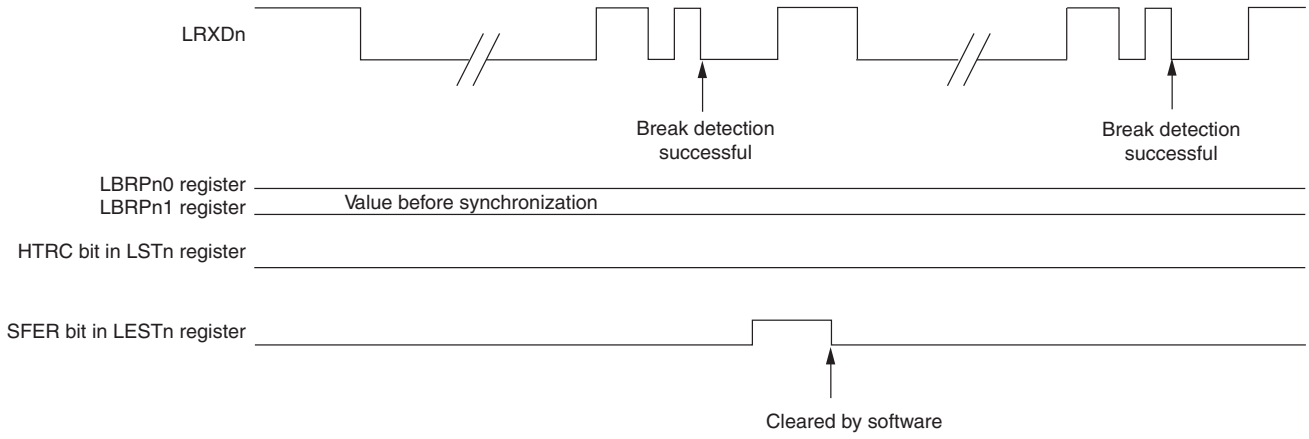



**Figure 17-10. Header Reception in LIN Slave Mode [Auto Baud Rate] (Sync Field Error)**



**(b) Response Transmission**

Figure 17-11 shows the operation of the LIN/UART module (in LIN slave mode) in response transmission. Table 17-11 provides processing in response transmission.

**Figure 17-11. Operation in Response Transmission**

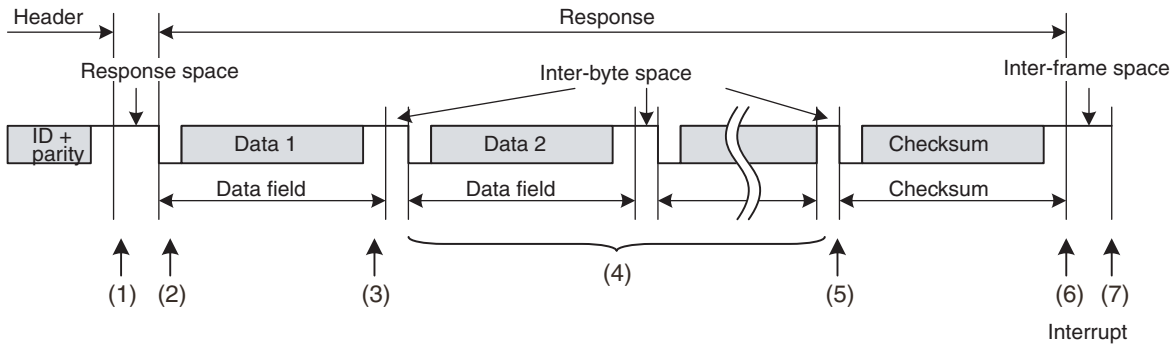



**Table 17-11. Processing in Response Transmission**

| Step | Software processing | LIN/UART module processing |
|---|---|---|
| (1) | • Sets the LDFCn register.<br>• Sets the LDBnm register.<br>• Sets the RTS bit in the LTRCn register to 1 (response transmission/reception started). | • Waits for the setting by software of the RTS bit or the LNRR bit in the LTRCn register.<br>• Sends a response space after the RTS bit in the LTRCn register is set to 1. |
| (2) | Waits for an interrupt request. | Transmits the data 1. |
| (3) | | Transmits the inter-byte space. |
| (4) | | • Transmits the data 2.<br>• Transmits an inter-byte space<br>• Transmits the data 3.<br>• Transmits an inter-byte space<br>(Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the LDFCn register, and stops the transmission when the BER bit in the LESTn register is 1 (bit error detected). If an error occurs, the checksum transmission in item (5) is not performed).<br>: |
| (5) | | Transmits the checksum. |
| (6) | | • Sets a successful frame/wake-up transmission flag or an error flag.<br>• Sets the RTS bit in the LTRCn register to 0 (response transmission/reception stopped) |
| (7) | • Processing after communication<br>Checks the LSTn register and clears flags. | • Completes the response transmission process.<br>• Waits for a new break. |

The LIN/UART module can receive a break field during the transmission or reception of a frame. Here, a reception status interrupt might be generated on the detection of a framing error, bit error, etc, at the position of the stop bit of the frame preceding break field reception.

For information about error detection, refer to **17.4.6 Error Status**.

### (c) Response Reception

Figure 17-12 shows the operation of the LIN/UART module (LIN slave mode) in response reception. Table 17-12 provides processing in response reception.

**Figure 17-12. Operation in Response Reception**

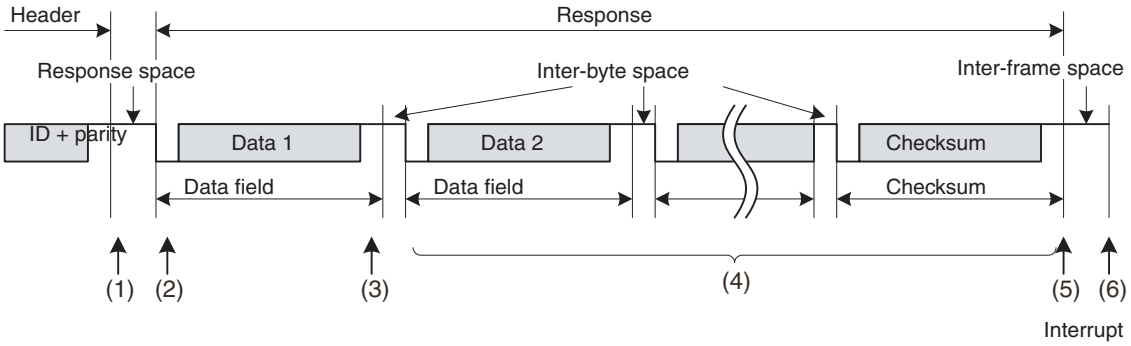



**Table 17-12. Processing in Response Reception**

| Step | Software processing | LIN/UART module processing |
|---|---|---|
| (1) | • Sets the LDFCn register.<br>• Sets the RTS bit in the LTRCn register to 1 (response transmission/reception started). | • Waits for the setting by software of the RTS bit or the LNRR bit in the LTRCn register.<br>• Waits for detection of the start bit. |
| (2) | Waits for an interrupt request. | Receives the data 1 when the start bit is detected. |
| (3) | | Sets the successful data 1 reception flag. |
| (4) | | • Receives the data 2 when the start bit is detected.<br>• Receives the data 3 when the start bit is detected.<br>Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the LDFCn register, and stops the transmission when any bit in the LESTn register is 1 (bit error detected). If an error occurs, the checksum determination in item (5) is not performed).<br>:<br>• Receives the checksum when the start bit is detected. |
| (5) | | • Determines the checksum.<br>• Sets a successful frame/wake-up reception flag or an error flag.<br>• Sets the RTS bit in the LTRCn register to 0 (response transmission/reception stopped). |
| (6) | • Processing after communication<br>Reads the received data.<br>Checks the LSTn register and clears flags. | • Completes the response process.<br>• Waits for a new break. |

The LIN/UART module can receive a break field during the transmission or reception of a frame. Here, a reception status interrupt might be generated on the detection of a framing error, bit error, etc, at the position of the stop bit of the frame preceding break field reception.
For information about error detection, refer to **17.4.6 Error Status**.

### (d) No-Response Request

Figure 17-13 shows the operation of the LIN/UART module (LIN slave mode) when no response is requested. Table 17-13 shows the processing that occurs when no response is requested.

**Figure 17-13. Operation when No Response is Requested**

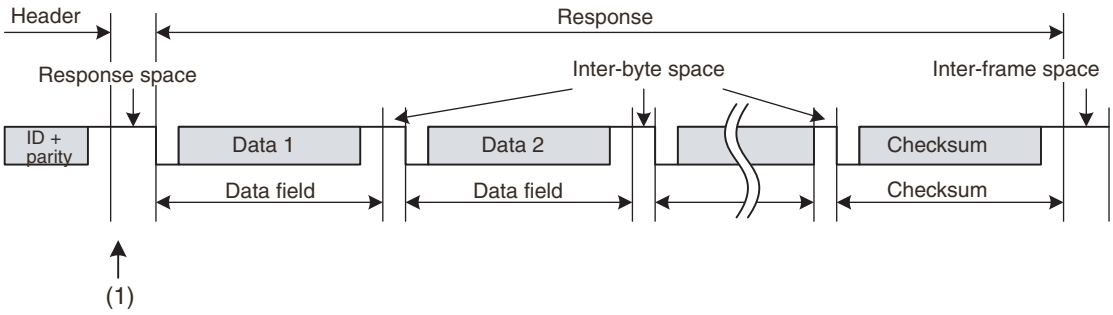



**Table 17-13. Processing when No Response is Requested**

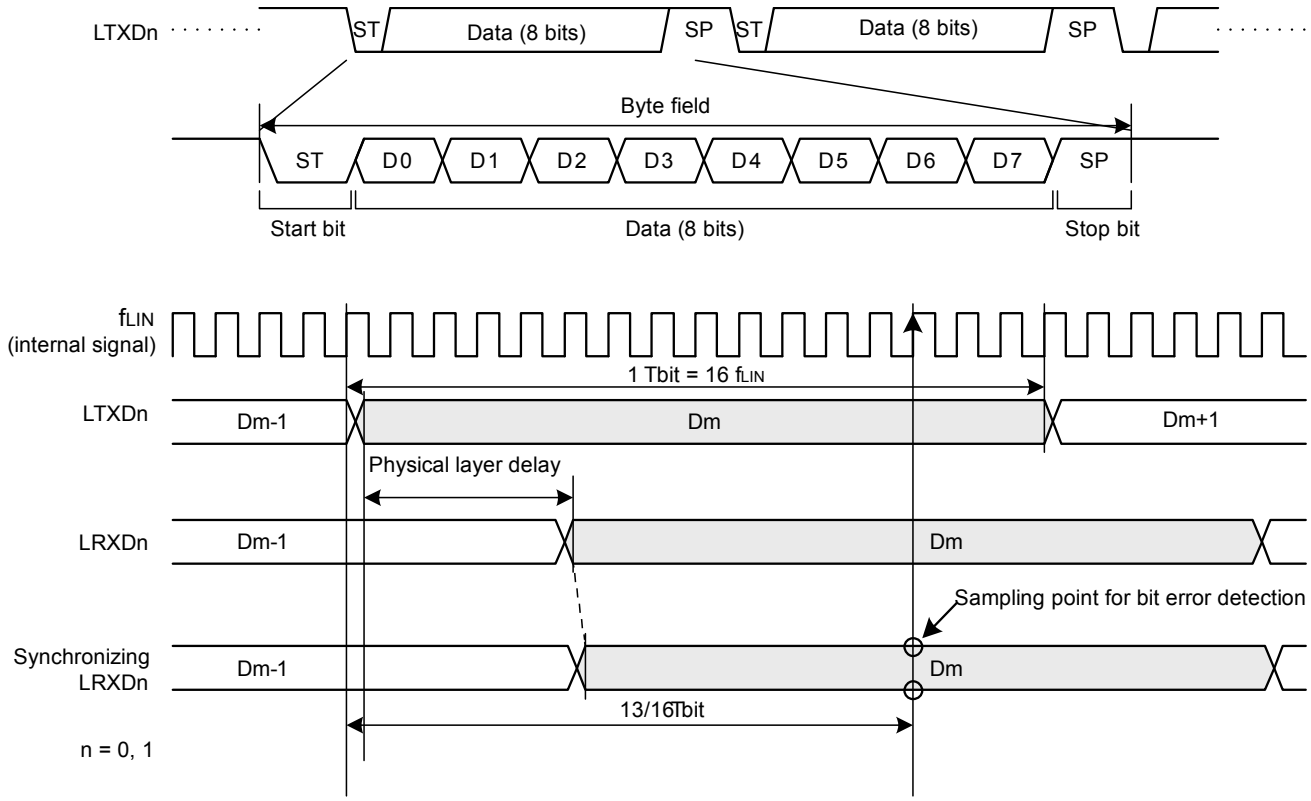
| Step | Software processing | LIN/UART module processing |
|---|---|---|
| (1) | • Sets the no-response request bit (LNRR bit) to 1. | • Waits for the setting of the no-response request bit (LNRR bit) by software.<br>• Completes the frame reception process.<br>• Waits for a new break. |

### 17.4.2 Data Transmission/Reception

#### (1) Data Transmission

One bit of data is transmitted per 1 Tbit.
The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data is compared bit by bit, and the results are stored in the BER flag in the LESTn register (see **17.4.6 Error Status**).
In LIN master mode and LIN slave mode [fixed baud rate], 1 Tbit is generated to be 16fLIN, and thus the sampling point for received data is at the 13th clock cycle (81.25% position).
In LIN slave mode [auto baud rate], if 1 Tbit is generated to be 4fLIN, the sampling point for received data is at the third clock cycle (75% position). If 1 Tbit is generated to be 8fLIN, the sampling point for received data is at the 7th clock cycle (87.5% position).
Figure 17-14 shows an example of data transmission timing.

**Figure 17-14. Example of Data Transmission Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])**

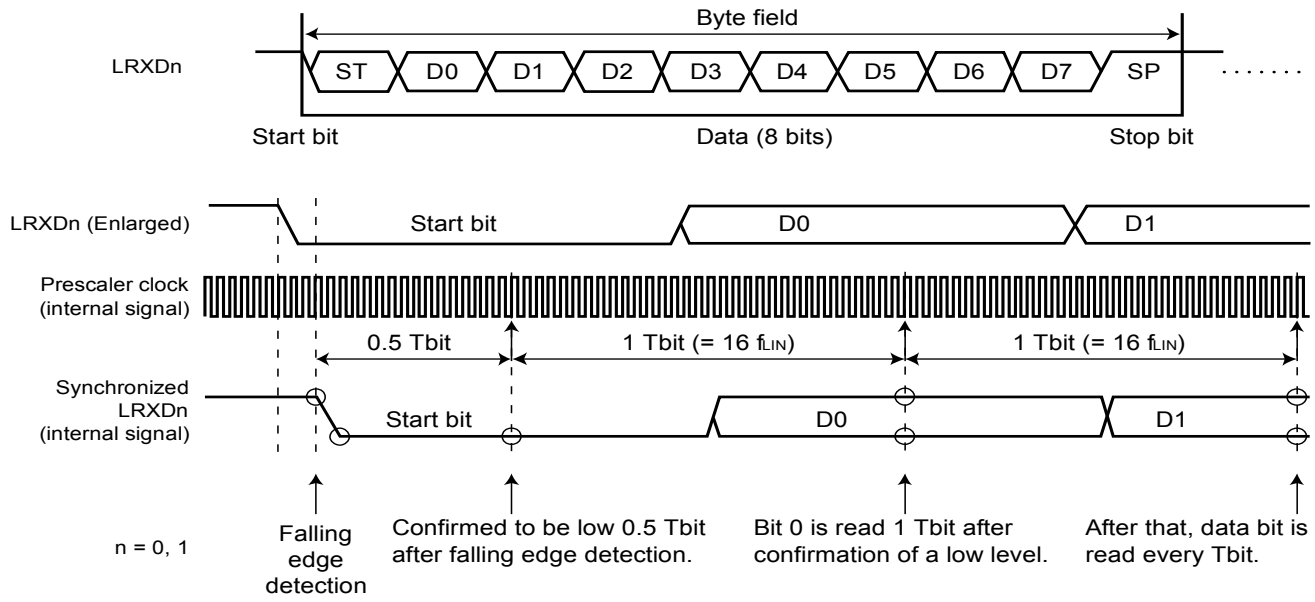


### (2) Data Reception

Data reception is performed by using the synchronized LRXDn signal (an internal signal) that is the input from the LRXDn pin synchronized with prescaler clock.
The byte field is synchronized at the falling edge of the start bit for the synchronized LRXDn signal. After the falling edge is detected, sampling is performed again 0.5 Tbit later, and the falling edge is recognized as a start bit if the synchronized LRXDn signal is low. The falling edge is not recognized as a start bit if the LRXDn signal after the clearing of the resetting is low-fixed or if a high level is detected on re-sampling.
After the start bit is detected, the system samples 1 bit per Tbit.
The LIN/UART module has a noise filter function with respect to reception data. If the LRDNFS bit in the LMDn register is 0, the LIN/UART module uses a noise filter, and for a sampling value the value determined by a 3-sampling majority rule on prescaler clocks is used. If the LRDNFS bit in the LMDn register is 1, the LIN/UART module does not use a noise filter, and for a sampling value the value of the synchronized LRXDn value at the sampling position is used as is.

Figure 17-15 shows an example of data reception timing.

**Figure 17-15. Example of Data Reception Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])**



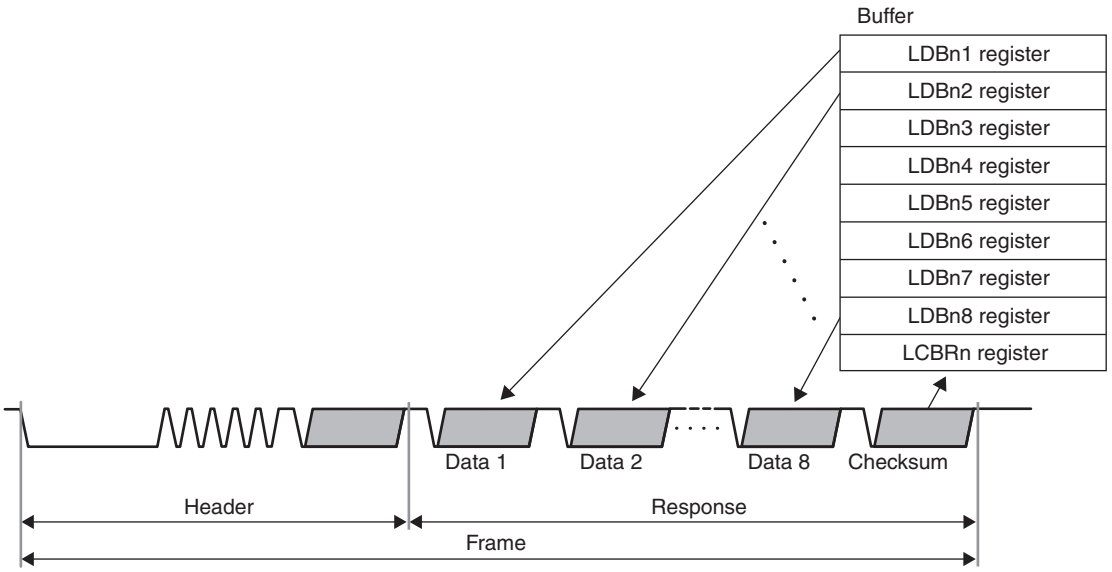
### 17.4.3 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN/UART module sends or receives data continuously.

#### (1) Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers LDBn1 to LDBn8 are sequentially transmitted to data areas 1 to 8 of the LIN frame. In the case of a 4-byte transmission, the contents stored in registers LDBn1 to LDBn4 are transmitted to data areas 1 to 4 of the LIN frame, but the contents of registers LDBn5 to LDBn8 are not transmitted. The transmitted checksum data is stored in the LCBRn register.
Figure 17-16 depicts the LIN transmission processing and the required buffer.

**Figure 17-16. LIN Transmission Processing and Required Buffer**




##### (a) Frame Separate Mode

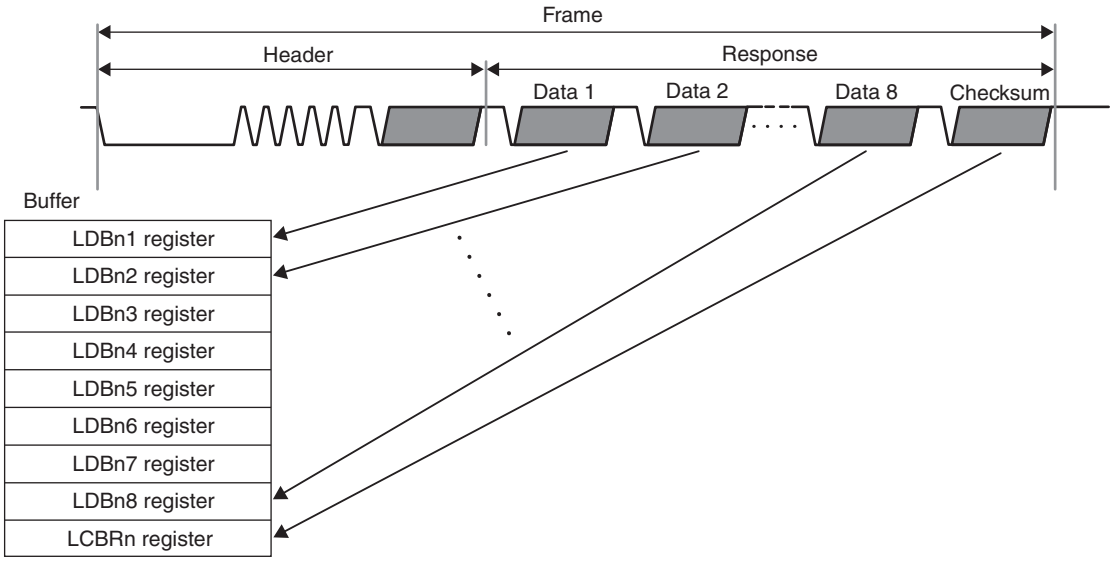
Setting the FSM bit in the LDFCn register to 1 turns on the frame separate mode.
In frame separate mode, a header and a response are transmitted when prompted by separate transmission start requests.
When the transmission of a header is finished, the HTRC flag in the LSTn register turns 1 (successful header transmission).

Use frame separate mode when sending or receiving response data of 9 bytes or greater in LIN master mode.

### (2) Reception of LIN Frames

For an 8-byte reception, the contents of data areas 1 to 8 of the LIN frame is stored in registers LDBn1 to LDBn8, respectively, upon receipt of a stop bit. In the case of a 4-byte reception, the contents of data areas 1 to 4 of the LIN frame are stored in registers LDBn1 to LDBn4, respectively; however, no data is stored in registers LDBn5 to LDBn8. Also, the received checksum data is stored in the LCBRn register.
Figure 17-17 depicts the LIN reception processing and the required buffer.

**Figure 17-17. LIN Reception Processing and Required Buffer**




#### (a) Reception of Data 1

When the reception of the first byte of data is finished, the D1RC flag in the LSTn register turns 1 (successful data 1 reception).

### (3) Multi-Byte Response Transmission/Reception Function

Normally in LIN communications, a response is 9 bytes or less (including a checksum field); however, responses in 10 bytes or greater can also be sent and received.
In such a case, the bit error, framing error, response preparation error detection, and auto checksum functions are enabled.

If the data length is greater than 8 bytes, the LSS bit should be set to 1 (indicating that the next data group to be sent or received is not the final data group) in the first data group (variable in 0 to 8 bytes) before sending or receiving the data group. After the transmission or reception, the user should determine whether the next data group is the final data group. If it is the final data group, the LSS bit should be set to 0 (indicating that the next data group to be sent or received is the final data group), and a checksum should be appended to the final data group.

By changing the RFDL bit settings when the RTS bit is 0, the user can change the data length for each data group.

When performing multi-byte response transmission/reception in LIN master mode, set the FSM bit in the LDFCn register to 1 (frame separate mode).

**Caution In LIN slave mode, the LIN/UART module can detect a new break field during the transmission or reception of a response.**

### 17.4.4 Wake-up Transmission/Reception

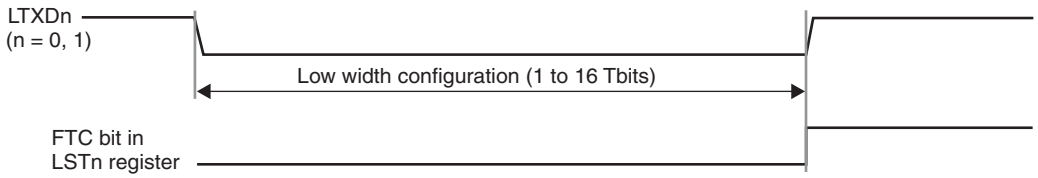
The wake-up transmission/reception can be used in LIN wake-up mode.

#### (1) Wake-up Transmission

In LIN wake-up mode, setting the RCDS bit in the LDFCn register to 1 (transmission) and the FTS bit in the LTRCn register to 1 (header reception or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low width of the wake-up signal should be set using the WUTL[3:0] bits in the LWUPn register. However, when the LWBR0 bit in the LWBRn register is 1 in LIN master mode, the low width is defined based on fa as the LIN system clock ($f_{LIN}$) regardless of the setting of LCKS bits in the LMDn register. Setting the baud rate to 19200 bps with fa selected and setting the WUTL[3:0] bits in the LWUPn register to 0100b (5 Tbits) allows 260 μs low level width of the signal to be output in LIN wake-up mode regardless of the setting of LCKS bits in the LMDn register.

If a wake-up low is output without any bit error, the FTC flag in the LSTn register turns 1 (successful response or wake-up transmission); when the FTCIE bit in the LIEn register is 1 (successful response/wakeup transmission interrupt enabled), an interrupt request is generated.
If a bit error is detected, wake-up transmission is canceled and the BER flag in the LESTn register is set to 1 (bit error detection).
Figure 17-18 shows the wake-up transmission timing.

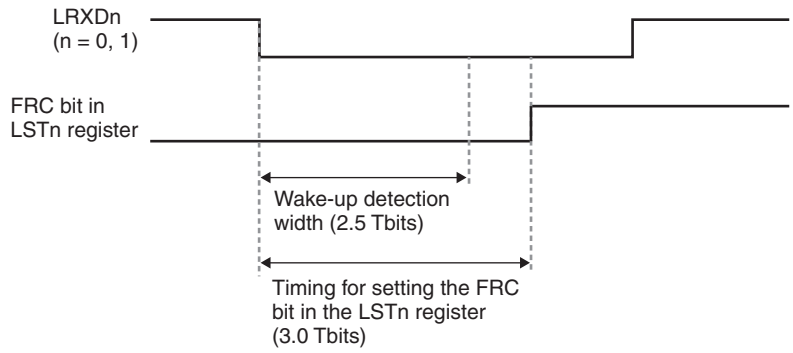
**Figure 17-18. Wake-up Transmission Timing**



**(2) Wake-up Reception**

The detection of a wake-up signal involves the use of an input signal low width count function.
The input signal low width count function measures the low width of the input signal to the LRXDn pin, using the same sampling point as data reception. The function can measure the input signal low width of 2.5 Tbits of fLIN or greater.
In LIN master mode, appropriately setting the LWBR0 bit in the LWBRn register allows switching between LIN operation mode and LIN wake-up mode without changing any baud rate generator setting.
Set the LWBR0 bit in the LWBRn register to 0 when LIN Specification Package Revision 1.3 is used, and set it to 1 when LIN Specification Package Revision 2.x is used. When the LWBR0 bit is set to 1, fa is always selected as the LIN system clock ($f_{LIN}$) regardless of the setting of LCKS bits in the LMDn register (setting of LCKS bits not affected).
Setting the baud rate to 19200 bps with fa selected allows 130 μs or longer low level width of the input signal to be detected regardless of the setting of LCKS bits in the LMDn register.
When using this function, in LIN wake-up mode, set the RFT bit in the LDFCn register to 0 (LIN master mode: reception), or RCDS bit to 0 (LIN slave mode: reception), and the FTS bit in the LTRCn register to 1 (LIN master mode: frame transmission or wake-up transmission/reception started; LIN slave mode: header reception or wake-up transmission/reception started).
When the low width to be measured is reached, the FRC flag in the LSTn register turns 1 (successful response/wake-up reception). If the FRCIE bit in the LIEn register is 1 (successful response or wake-up reception interrupt enabled), an interrupt request is generated.

**Figure 17-19. Input Signal Low Count Function**



### 17.4.5 Status

During LIN mode operation, the LIN/UART module can detect seven types of statuses.
The four statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, successful header transmission//header reception, can generate interrupt requests.
Table 17-14 shows the types of statuses available in LIN master mode. Table 17-15 lists the types of statuses available in LIN slave mode [auto baud rate] and in LIN slave mode [fixed baud rate].

**Table 17-14. Types of Statuses in LIN Master Mode**

| Status | Status set condition | Status clear condition | Operation mode capable of status detection | Corresponding bit | Interrupt |
|---|---|---|---|---|---|
| Reset | After the OM0 bit in the LCUCn register is set to not-LIN–reset-mode, if actually the LIN/UART module is cleared from LIN reset mode. | After the OM0 bit in the LCUCn register is set to LIN reset mode, if actually the LIN/UART module enters LIN reset mode. | All modes | OMM0 bit in LMSTn register | Not available |
| Operation mode | After the OM1 bit in the LCUCn register is set to LIN operation mode, if actually the LIN/UART module enters LIN operation mode. | After the OM1 bit in the LCUCn register is set to LIN wake-up mode, if actually the LIN/UART module enters LIN wake-up mode. | • LIN operation mode<br>• LIN wake-up mode | OMM1 bit in LMSTn register | Not available |
| Frame/wake-up transmission end | When a frame (header transmission + response transmission), a wake-up signal, or a data group is transmitted successfully. | • When another communication is started<br>• When cleared by software<br>• After transition to LIN reset mode | • LIN operation mode<br>• LIN wake-up mode | FTC flag in LSTn register | Available |
| Frame/wake-up reception end | When a frame (header transmission + response reception), a wake-up signal, or a data group is received successfully. | • When another communication is started<br>• When cleared by software<br>• After transition to LIN reset mode | • LIN operation mode<br>• LIN wake-up mode | FRC flag in LSTn register | Available |
| Error detection | If any of the PRER flag, CSER flag, FER flag, FTER flag, PBER flag, and BER flags in the LESTn register turns 1 (error detected). | • When another communication is started<br>• When cleared by software Note 1<br>• After transition to LIN reset mode | • LIN operation mode<br>• LIN wake-up mode | ERR flag in LSTn register | Available |
| Data 1 reception end | The RFT bit in the LDFCn register is 0 (reception) and the first byte of the response field is received Note 2. | • When another communication is started<br>• When cleared by software<br>• After transition to LIN reset mode | LIN operation mode | D1RC flag in LSTn register | Not available |
| Header reception end | When a header field is received successfully. | • When another communication is started<br>• When cleared by software<br>• After transition to LIN reset mode | LIN operation mode | HTRC flag in LSTn register | Available |

<R> **Notes 1.** In LIN wake-up mode and LIN operation mode, the ERR flag in the LSTn register is cleared to 0 by writing 0 to the PRER flag, CSER flag, FER flag, FTER flag, PBER flag or BER flags in the LESTn register.

**2.** Not detected when the RFDL[3:0] bits in the LDFCn register are 0000b (0-byte + checksum).

**Table 17-15. Types of Statuses in LIN Slave Mode**

| Status | Status set condition | Status clear condition | Operation mode capable of detecting a status | Corresponding bit | Interrupt |
|---|---|---|---|---|---|
| Reset | After the OM0 bit in the LCUCn register is set to not-LIN–reset-mode, if actually the LIN/UART module is cleared from LIN reset mode. | After the OM0 bit in the LCUCn register is set to LIN reset mode, if actually the LIN/UART module enters LIN reset mode. | All modes | OMM0 bit in LMSTn register | Not available |
| Operation mode | After the OM1 bit in the LCUCn register is set to LIN operation mode, if actually the LIN/UART module enters LIN operation mode. | After the OM1 bit in the LCUCn register is set to LIN wake-up mode, if actually the LIN/UART module enters LIN wake-up mode. | • LIN operation mode<br>• LIN wake-up mode | OMM1 bit in LMSTn register | Not available |
| Frame/wake-up transmission end | When a response field, a wake-up signal, or a data group is transmitted successfully. | • When cleared by software<br>• After transition to LIN reset mode | • LIN operation mode<br>• LIN wake-up mode | FTC flag in LSTn register | Available |
| Frame/wake-up reception end | When a response field, a wake-up signal, or a data group is received successfully. | • When cleared by software<br>• After transition to LIN reset mode | • LIN operation mode<br>• LIN wake-up mode | FRC flag in LSTn register | Available |
| Error detection | If any of the PRER flag, IPER flag, CSER flag, SFER flag, FER flag, TER flag ,and BER flags in the LESTn register turns 1 (error detected). | • When cleared by software Note 1<br>• After transition to LIN reset mode | • LIN operation mode<br>• LIN wake-up mode | ERR flag in LSTn register | Available |
| Data 1 reception end | The RCDS bit in the LDFCn register is 0 (reception) and the first byte of the response field is received Note 2. | • When cleared by software<br>• After transition to LIN reset mode | LIN operation mode | D1RC flag in LSTn register | Not available |
| Header reception end | When a header field is received successfully. | • When cleared by software<br>• After transition to LIN reset mode | LIN operation mode | HTRC flag in LSTn register | Available |

<R>

**Notes 1.** In LIN wake-up mode and LIN operation mode, the ERR flag in the LSTn register is cleared to 0 by writing 0 to the PRER flag, IPER flag, CSER flag, SFER flag, FER flag, TER flag or BER flags in the LESTn register.

**2.** Not detected when the RFDL[3:0] bits in the LDFCn register are 0000b (0-byte + checksum).

### 17.4.6 Error Status

#### (1) LIN Master Mode

##### (a) Types of Error Statuses

The LIN/UART module can detect six types of error statuses in LIN master mode. The condition of these error statuses can be checked by means of the corresponding bits in the LESTn register.

All error statuses represent interrupt events.

Table 17-16 shows the types of error statuses.

**Table 17-16. Types of Error Statuses in LIN Master Mode**

| Status | Error detection condition | Operation mode capable of error detection | Communication | Enable/disable detection | Corresponding bit |
|---|---|---|---|---|---|
| Bit error | The transmitted data and the data on the LIN bus monitored by the receive pin do not match[Note 1, 2] | • LIN operation mode<br>• LIN wake-up mode | Cancel | O | BER flag in LESTn register |
| Physical bus error | • LIN bus is detected to be high when sending a break<br>• LIN bus is detected to be low when sending a break delimiter<br>• LIN bus is detected to be high when sending a wake-up | • LIN operation mode<br>• LIN wake-up mode | Cancel | O | PBER flag in LESTn register |
| Timeout error | A frame or response transmission/reception does not terminate within a given time[Note 3] | LIN operation mode | Cancel | O | FTER flag in LESTn register |
| Framing error | In response field reception, a stop bit of each data byte is low | LIN operation mode | Cancel | O | FER flag in LESTn register |
| Checksum error | In response field reception, the result of checksum test gives an error | LIN operation mode | — | × | CSER flag in LESTn register |
| Response preparation error | The following conditions occur in frame separate mode:<br>• The first reception data byte is received after completion of header transmission but before a response transmission/reception request is set<br>• The first reception data byte is received after completion of the previous data group reception but before a transmission/reception request for another data group is set | LIN operation mode | Cancel | × | RPER flag in LESTn register |

**Notes 1.** If a bit error is detected, the process is canceled after a stop bit is sent. If a bit error is detected in a non-
<R> data area, such as an inter-byte space, the transmission is canceled immediately after transmission of error bit. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is canceled after the error-causing bit is sent.

**2.** In multi-byte response transmission, a bit error can be detected between data groups.

**3.** The timeout time depends on the response field data length (the RFDL[3:0] bits in the LDFCn register) and
<R> the checksum selection (the CSM bit in the LDFCn register), and this can be calculated according to the following formula:

Timeout time is 8 data bytes until setting of LTRCn register in frame separate mode (FSM bit of LDFCn register is set to 1).

[Frame timeout]

- On classic selection (when the CSM bit in the LDFCn register is 0):
  Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
- On enhanced selection (when the CSM bit in the LDFCn register is 1):
  Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is a time greater than the TFRAME_MAX of LIN Specification Package Revision 1.3 on classic selection, or the TFRAME_MAX of LIN Specification Package Revision 2.x on enhanced selection.

[Response timeout]

Timeout time = (number of data bytes + 1) × 14 [Tbit]

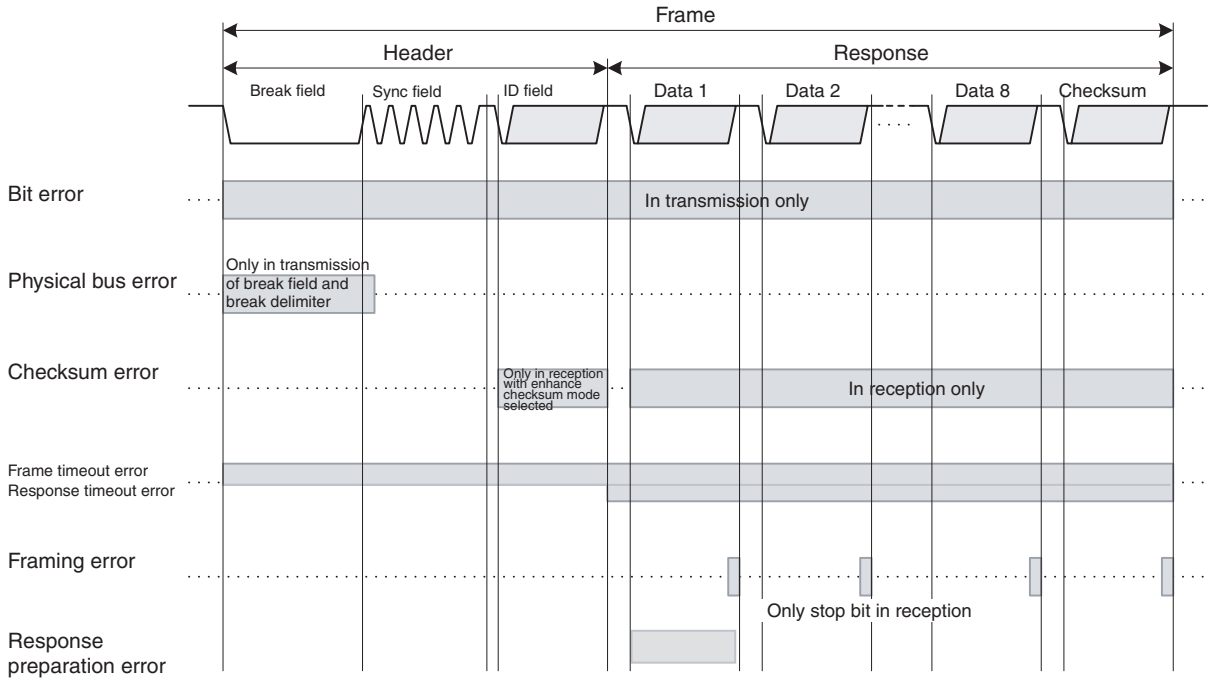
**Caution The error status is cleared when another communication is started, when cleared by software, or after transition to LIN reset mode.**

**(b) Target Time Area for LIN Error Detection**

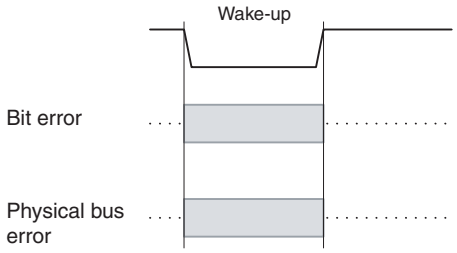
Figure 17-20 shows the time domain in which the LIN/UART module in master mode performs monitoring for error detection.

**Figure 17-20. Target Time Area for LIN Error Detection (LIN Master Mode)**

<Frame transmission/reception>




<Wake-up transmission>



### (2) LIN Slave Mode

#### (a) Types of Error Statuses

The LIN/UART module can detect seven types of error statuses in LIN slave mode [auto baud rate] or in LIN slave mode [fixed baud rate]. These error statuses can be verified by checking the corresponding bits in the LESTn register.

Table 17-17 shows the types of error statuses.

**Table 17-17. Types of Error Statuses in LIN Slave Mode**

| Status | Error detection condition | Operation mode capable of error detection | Communication | Enable/ disable detection | Corresponding bit |
|---|---|---|---|---|---|
| Bit error | The transmitted data and the data on the LIN bus monitored by the receive pin do not match Notes 1, 2 | • LIN operation mode<br>• LIN wake-up mode | Cancel | O | BER flag in LESTn register |
| Timeout error | A frame or response transmission/reception does not terminate within a given time Note 3 | LIN operation mode | Cancel | O | TER flag in LESTn register |
| Framing error | In frame reception, a stop bit of each data byte is low | LIN operation mode | Cancel | O | FER flag in LESTn register |
| Sync field error | If the width of the break low is greater than the width set by the BLT bit in the LBFCn register and the sync field is not 55H | LIN operation mode | Cancel | O Note 4 | SFER flag in LESTn register |
| Checksum error | In response frame reception, the result of checksum test gives an error | LIN operation mode | — Note 5 | × | CSER flag in LESTn register |
| ID parity error | If the received ID parity bit does not match the value that is automatically calculated by the LIN/UART module | LIN operation mode | Cancel | O | IPER flag in LESTn register |
| Response preparation error | • After the reception of a header, before the first reception data byte is received, response preparation is not made in time.<br>• During a multi-byte response transmission/reception, before the first reception data byte of another data group is received, preparation for another data group is not made in time. | LIN operation mode | Cancel | × | RPER flag in LESTn register |

**Notes 1.** If a bit error is detected, the process is canceled after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is canceled immediately after transmission of error bit. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is canceled after the error-causing bit is sent. <R>

**2.** In multi-byte response transmission, a bit error can be detected between data groups.

**3.** The timeout time depends on the response field data length (the RFDL[3:0] bits in the LDFCn register) and the checksum selection (the LCS bit in the LDFCn register), and this can be calculated according to the following formulae:

Until the RTS or LNRR bit in the LTRCn register is set, the timeout time is set based on 8-byte data. Once the RTS bit is set, the timeout time is re-set based on the response field data length (the RFDL[3:0] bits in the LDFCn register). When the LNRR bit is set, the timeout function is stopped.

[Frame timeout]

- On classic selection (when the CSM bit in the LDFCn register is 0)
  Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
- On enhanced selection (when the CSM bit in the LDFCn register is 1)
  Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is a time greater than the TFRAME_MAX of LIN Specification Package Revision 1.3 on classic selection, or the TFRAME_MAX of LIN Specification Package Revision 2.x on enhanced selection.

[Response timeout]

Timeout time = (number of data bytes + 1) × 14 [Tbit]

4. Only indication in the SFER flag can be enabled or disabled; error detection cannot be enabled or disabled.
5. Checksum determination is performed after response frame reception is completed. If the result is determined as an error, the successful reception flag is not set to 1.

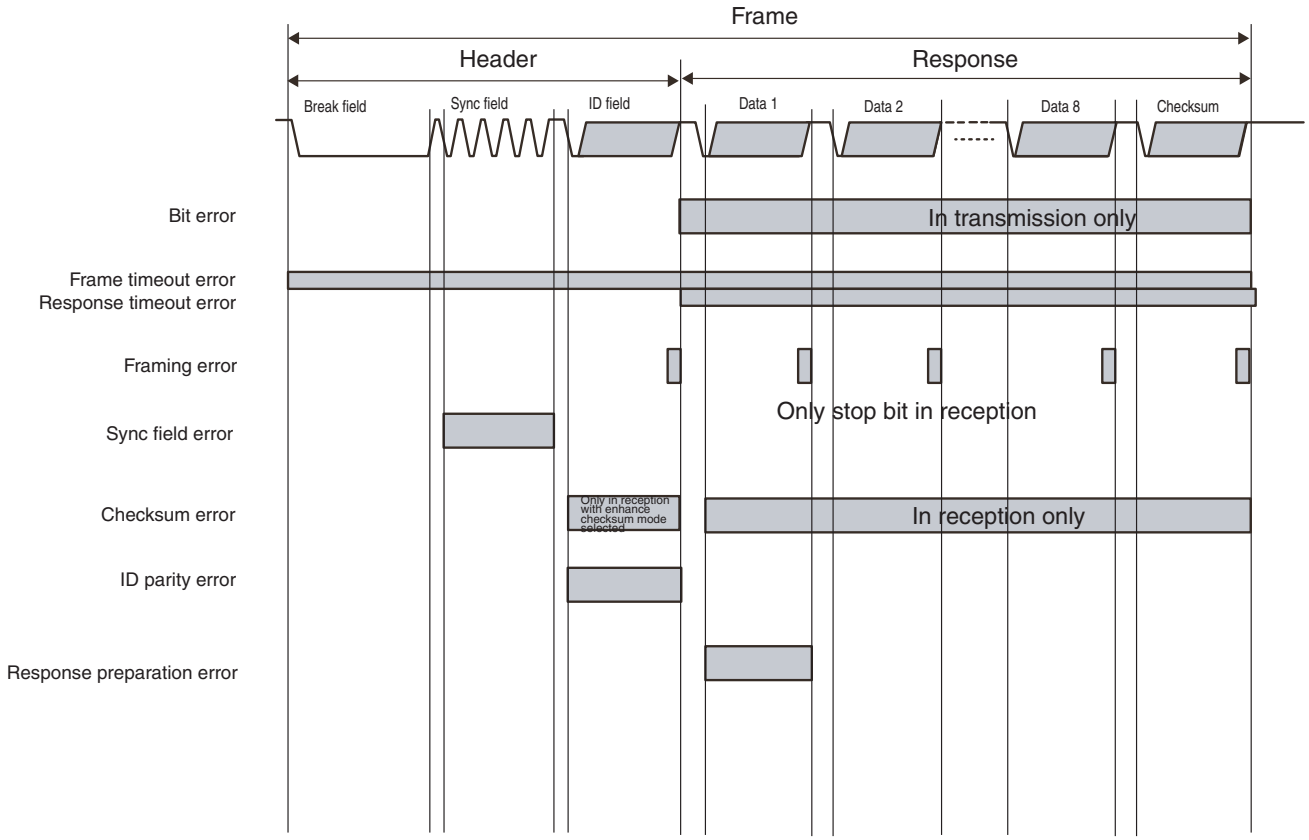
**Caution The error status is cleared when cleared by software or after transition to LIN reset mode.**

**(b) Target Time Area for LIN Error Detection**

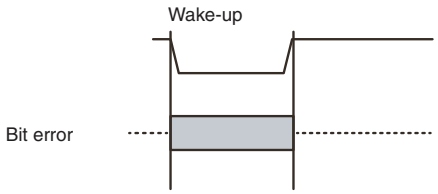
Figure 17-21 shows the time domain in which the LIN/UART module in slave mode performs monitoring for error detection.

**Figure 17-21. Target Time Area for LIN Error Detection (LIN Slave Mode)**

<Frame transmission/reception>




<Wake-up transmission>



## 17.5 UART Mode

### 17.5.1 Operation Overview

#### (1) Transmission

Figure 17-22 shows LIN/UART module (in UART mode) transmission operations; Table 17-18 shows LIN/UART module (in UART mode) transmission processing.

**Figure 17-22. LIN/UART Module (in UART mode) Transmission Operation**

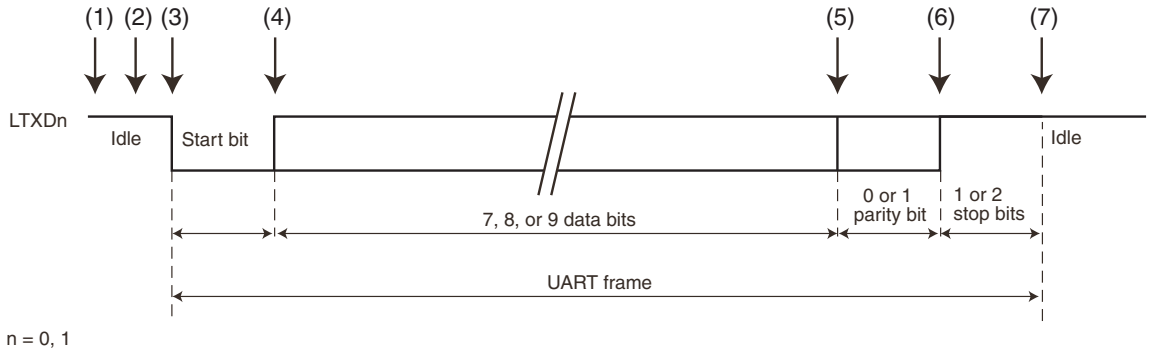



**Table 17-18. LIN/UART Module (UART Mode) Transmission Processing**

| Step | Software processing | LIN/UART module processing |
|---|---|---|
| (1) | • Sets a baud rate.<br>• Sets noise filter ON/OFF.<br>• Sets error detection enable.<br>• Sets data format<br>• Sets an interrupt generation timing.<br>• Clears the LIN/UART module from LIN reset mode<br>• Sets the transmit enable bit (UTOE bit) to 1 | • Waits for a transmission trigger (LUTDRn register) by software. |
| (2) | • Sets the transmission data in the UART transmission data register (LUTDRn register) or UART wait transmission data (LUWTDRn). | • Sets the transmission status flag. |
| (3) | • Waits for an interrupt request.<br><br>[When the UTIGTS bit is 0 (a transmission interrupt request is output upon start of transmission)]<br>• When transmitting data continuously, sets another piece of transmission data in the UART transmission data register (LUTDRn register), waits for the generation of an interrupt request. | • Transmits a start bit (for switching between transmission and reception in half duplex communication, transmits a start bit after receiving 1 stop bit. This function is referred to in **17.5.1 (4) Transmission Start Wait Function**).<br><br>[When the UTIGTS bit is 0 (a transmission interrupt request is output upon start of transmission)]<br>• Outputs a transmission interrupt. |
| (4) | | Transmits the data set in the UART (wait) transmission data register. |
| (5) | | Transmits a parity bit when parity is used. |
| (6) | | Transmits 1 or 2 stop bits. |
| (7) | [When the UTIGTS bit is 0 (a transmission interrupt request is output upon start of transmission)]<br>• If another piece of transmission data is set, goes to step (3).<br><br>[When the UTIGTS bit is 1 (a transmission interrupt request is output upon end of transmission)]<br>• When transmitting data continuously, goes to step (2). | [When the UTIGTS bit is 0 (a transmission interrupt request is output upon start of transmission)]<br>• If another piece of transmission data is set, goes to step (3).<br>• If another piece of transmission data is not set, clears the transmission status flag.<br><br>[When the UTIGTS bit is 1 (a transmission interrupt request is output upon end of transmission)]<br>• Outputs a transmission interrupt.<br>• Clears the transmission status flag |

**(a) Continuous Transmission**

The LIN/UART module (in UART mode) can transmit multiple sets of data continuously by using the LUTDRn register. Figure 17-23 shows an operating example where the transmission interrupt generation timing is the start of transmission. Figure 17-24 shows an operating example where the transmission interrupt generation timing is the end of transmission.

**Figure 17-23. LIN/UART Module (in UART mode) Continuous Transmission Operation (when UTIGTS Bit in LUORn1 Register is 0)**

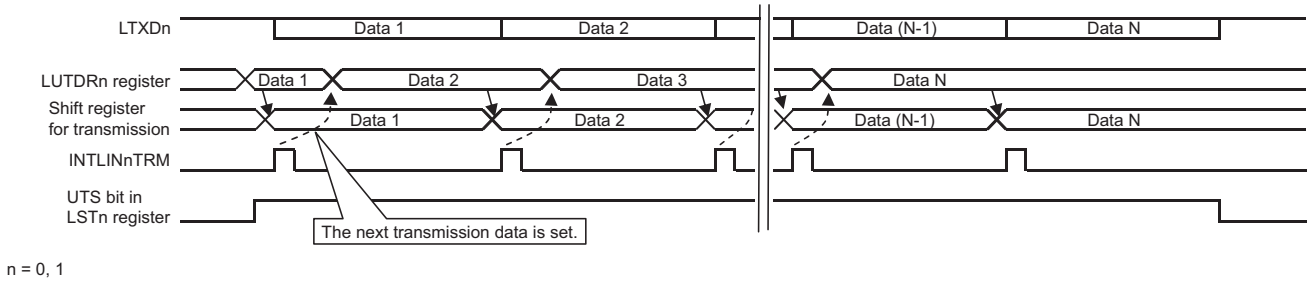
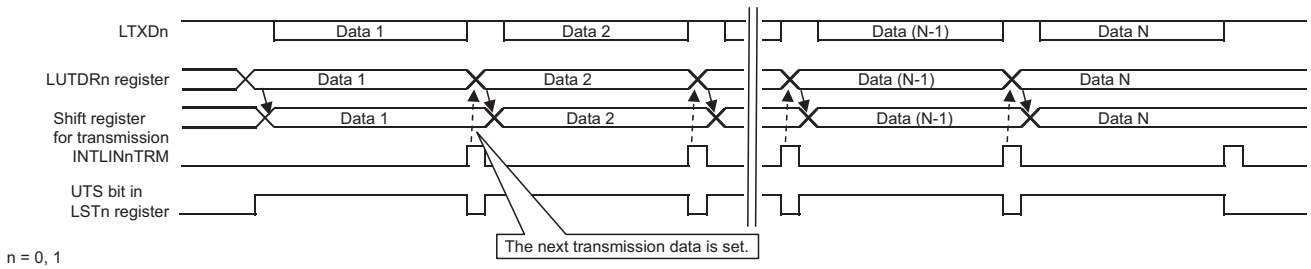



**Figure 17-24. LIN/UART Module (in UART mode) Continuous Transmission Operation (when UTIGTS Bit in LUORn1 Register is 1)**




An interrupt can be generated at the end of a transmission by changing the UTIGTS bit in the LUORn1 register from 0 to 1 after the start of transmission of final data, provided only that the transmission interrupt generation timing is the start of transmission and the end of transmission of final data needs to be known.

**(b) UART Buffer Transmission**

The LIN/UART module (in UART mode) has a maximum of nine bytes of UART buffers, and thus it is capable of performing continuous transmissions through the use of UART buffers.
Figure 17-25 shows the UART buffer transmission operation in the LIN/UART module (in UART mode). Table 17-19 shows the UART buffer transmission processing.

**Figure 17-25. UART Buffer Transmission in LIN/UART Module (in UART mode)**

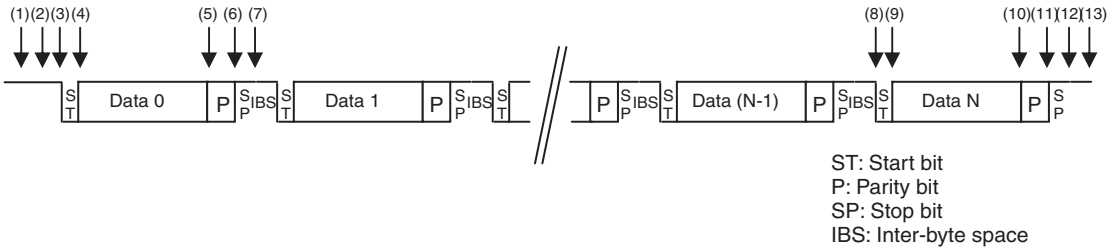


**Table 17-19. UART Buffer Transmission Processing in LIN/UART Module (in UART mode)**

| Step | Software processing | LIN/UART module processing |
|---|---|---|
| (1) | • Sets a baud rate.<br>• Sets noise filter ON/OFF.<br>• Sets error detection enable.<br>• Sets data format.<br>• Sets an interrupt generation timing to the end of transmission.<br>• Clears the LIN/UART module from LIN reset mode.<br>• Sets the transmit enable bit (UTOE bit) to 1 | • Waits for a transmission trigger (RTS bit) by software. |
| (2) | • Sets a UART buffer data length and whether the system must wait for the start of transmission.<br>• Sets the transmission data in the UART data buffer 0 register (LUDBn0) and the LIN/UART data buffer m register (LDBnm).<br>• Sets the UART buffer transmission start bit (RTS). | • Sets the transmission status flag. |
| (3) | • Waits for an interrupt request. | • Transmits a start bit (for switching between transmission and reception in half duplex communication, transmits a start bit after receiving 1 stop bit. This function is referred to in **17.5.1 (4) Transmission Start Wait Function**). |
| (4) | | Transmits the data set in the UART data buffer 0 register (LUDBn0) and the LIN/UART data buffer m register (LDBnm). |
| (5) | | Transmits a parity bit when parity is used. |
| (6) | | Transmits 1 or 2 stop bits |
| (7) | | Transmits an inter-byte space (idle). |
| | | Repeats steps (3) to (7) until frame count -1 that was set in the UART buffer data length select bits is reached. |
| (8) | | Transmits a start bit. |
| (9) | | Transmits the data set in the LIN/UART data buffer m register (LDBnm). |
| (10) | | Transmits a parity bit when parity is used. |
| (11) | | Transmits 1 or 2 stop bits. |
| (12) | | • Sets the buffer transmission end flag.<br>• Clears the UART buffer transmit start (RTS) bit.<br>• Outputs a transmission interrupt.<br>• Clears the transmission status flag. |
| (13) | • Checks the LSTn register and clears flags<br>• When transmitting data continuously, goes to step (2). | |

n = 0, 1

### (2) Reception

Figure 17-26 shows the LIN/UART module (in UART mode) reception operation. Table 17-20 shows the LIN/UART module (in UART mode) reception processing.

**Figure 17-26. LIN/UART Module (in UART Mode) Reception Operation**

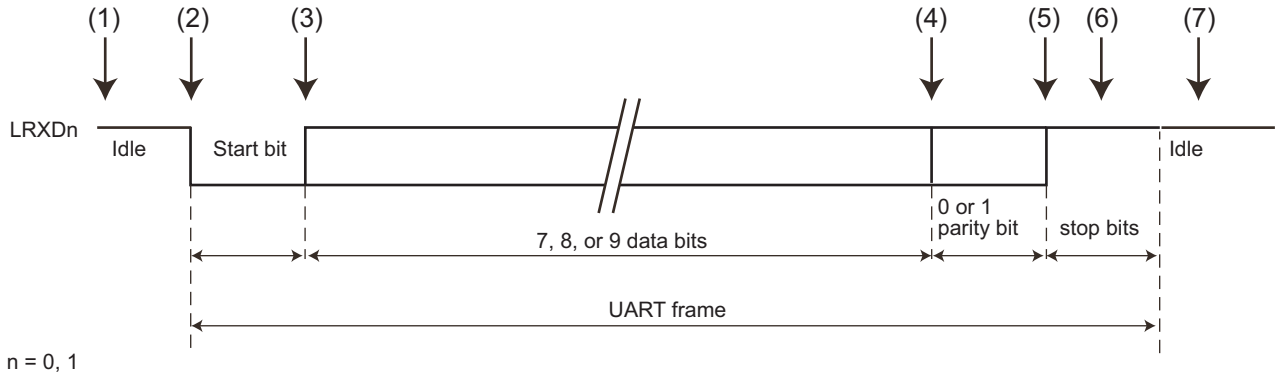



**Table 17-20. LIN/UART Module (in UART Mode) Reception Processing**

| Step | Software processing | LIN/UART module processing |
|---|---|---|
| (1) | • Sets a baud rate.<br>• Sets noise filter ON/OFF.<br>• Sets error detection enable.<br>• Sets data format.<br>• Clears the LIN/UART module from LIN reset mode.<br>• Sets the receive enable bit (UROE bit) to 1. | • Waits for reception enable state switching by software.<br>• Waits for detection of a start bit. |
| (2) | Waits for an interrupt request. | • Waits for a falling edge from the reception pin, and detects a start bit.<br>• Sets the reception status flag. |
| (3) | | Receives data. |
| (4) | | Receives a parity bit when parity is used. |
| (5) | | Receives only 1 stop bit. |
| (6) | | • Outputs a successful reception interrupt.<br>• Clears the reception status flag. |
| (7) | Checks the LSTn register and clears flags. | Waits for a falling edge from the reception pin. |

n = 0, 1

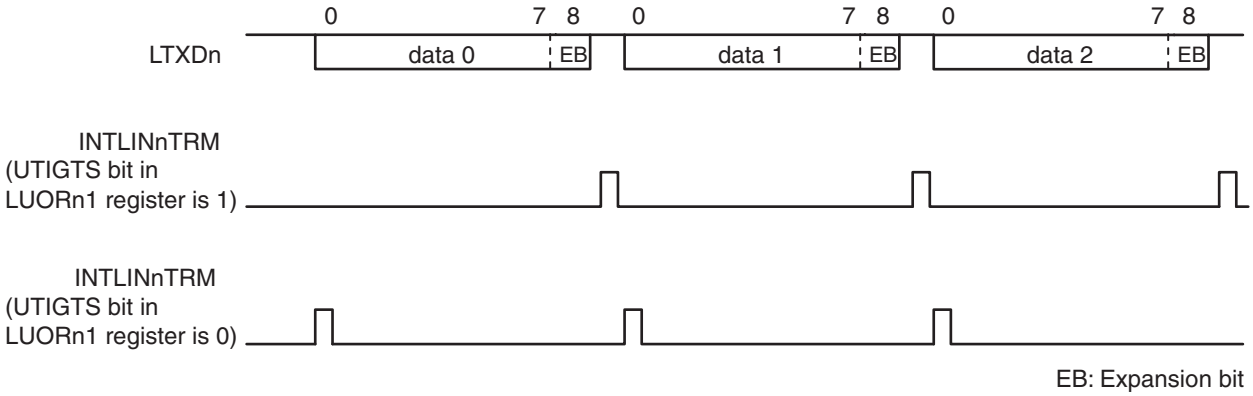
### (3) Expansion Bits

The LIN/UART module (in UART mode) can transmit and receive 9-bit long data by setting the UEBE bit in the LUORn1 register to 1.

#### (a) Expansion Bit Transmission

The LIN/UART module (in UART mode) can transmit 9-bit long data when the expansion bit enable bit (UEBE) in UART option register 1 (LUORn1) is 1 and by writing the 9-bit data to either the UART transmission data register (LUTDRn) or the UART wait transmission data register (LUWTDRn).

**Figure 17-27. Transmission Example When Expansion Bit is Enabled (LSB First)**

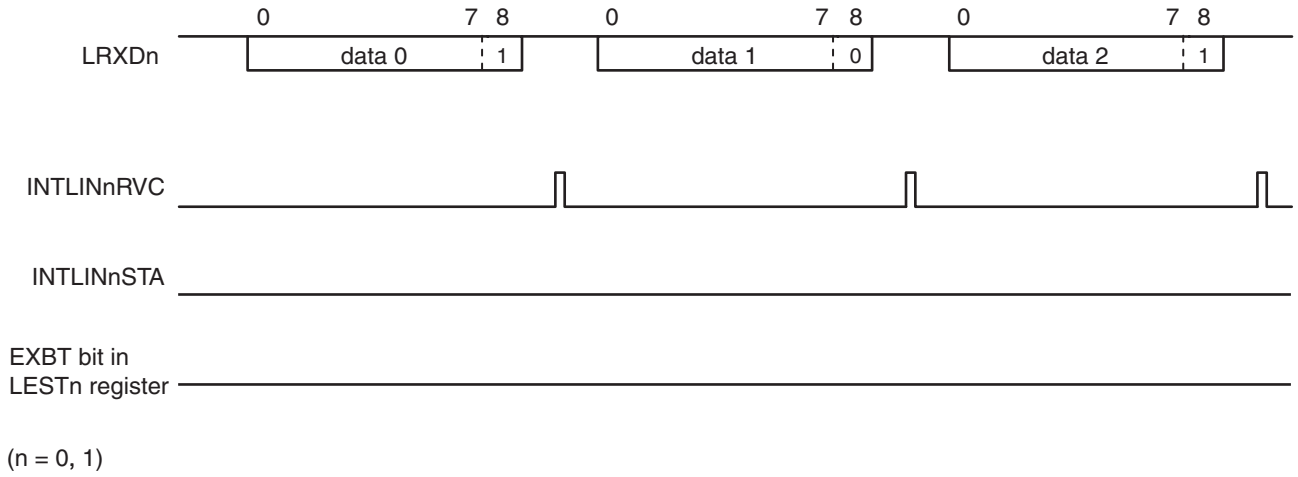



EB: Expansion bit

#### (b) Expansion Bit Reception

With the LIN/UART module (in UART mode), 9-bit data can always be received without requiring a comparison of expansion bits, provided that the expansion bit enable bit (UEBE) in UART option register 1 (LUORn1) is 1, the expansion bit comparison disable bit (UECD) is 1, and the expansion bit/data comparison enable bit (UEBDCE) is 0. Irrespective of the particular setting of the expansion bit detection level select bit (UEBDL) in UART option register 1 (LUORn1), a successful LINn reception interrupt is generated (n = 0, 1) when 9-bit data is received.

**Figure 17-28. Expansion Bit Reception Example (LSB First)**



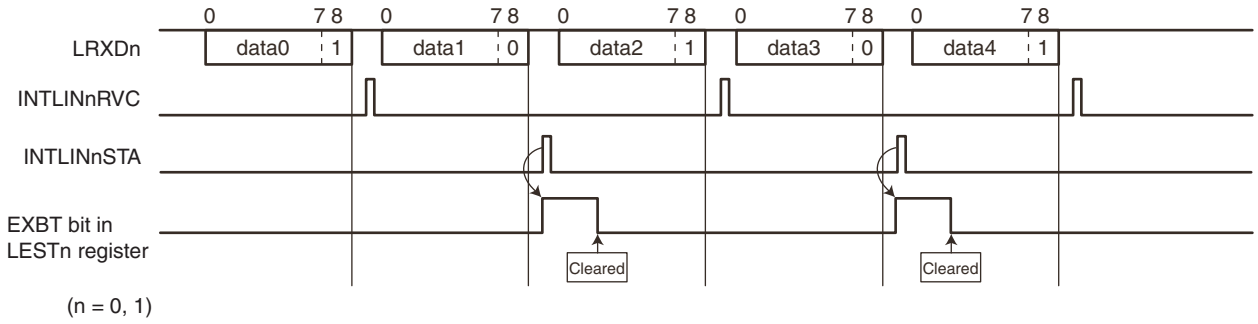

(n = 0, 1)

**(c) Expansion Bit Reception (with Expansion Bit Comparison)**

The LIN/UART module (in UART mode) can compare received expansion bits and the UEBDL bits when the expansion bit enable bit (UEBE) in UART option register 1 (LUORn1) is 1, the expansion bit comparison disable bit (UECD) is 0, and the expansion bit/data comparison enable bit (UEBDCE) is 0.
If the level that was set in the expansion bit detection level select bit (UEBDL) is detected, a LINn reception status interrupt is generated upon completion of data reception, and the expansion bit detection flag (EXBT) in the LIN/UART error status register (LESTn) is set. If the reversed value of an expansion bit detection level is detected, a successful LINn reception interrupt is generated. In either case, the received data is stored in the UART reception data register (LURDRn), unless there was an overrun error.

Figure 17-29 shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.

**Figure 17-29. Expansion Bit Reception Example (with Expansion Bit Comparison) (LSB First, UEBDL = 0)**




**Notes 1.** If a reception error (parity error, framing error, or overrun error) occurs in received data 0, 2, or 4 (if a reversed value of an expansion bit detection level is detected), a LINn reception status interrupt is generated, and the error flag is updated. In this case, a successful LINn reception interrupt is not generated

**2.** If a reception error (parity error, framing error, or overrun error) occurs in received data 1 or 3 (if an expansion bit detection level is detected), a LINn reception status interrupt is generated, and the error flag is updated. In the case of an overrun error, the expansion bit detection flag (EXBT) is also set.

**(d) Expansion Bit Reception (with Data Comparison)**

The LIN/UART module (in UART mode) compares the 8-bit received data excluding the expansion bits with the preset LIDBn register value if the level that was set in the expansion bit detection level select bit (UEBDL) is detected when the expansion bit enable bit (UEBE) in UART option register 1 (LUORn1) is 1, the expansion bit comparison disable bit (UECD) is 0, and the expansion bit/data comparison enable bit (UEBDCE) is 1. If the compared two values agree, the following operations are executed.

- A LINn reception status interrupt is generated (n = 0 or 1).
- The expansion bit detection flag (EXBT) is set.
- The ID match flag (IDMT) is set.
- The received data is stored in the UART reception data register (LURDRn).

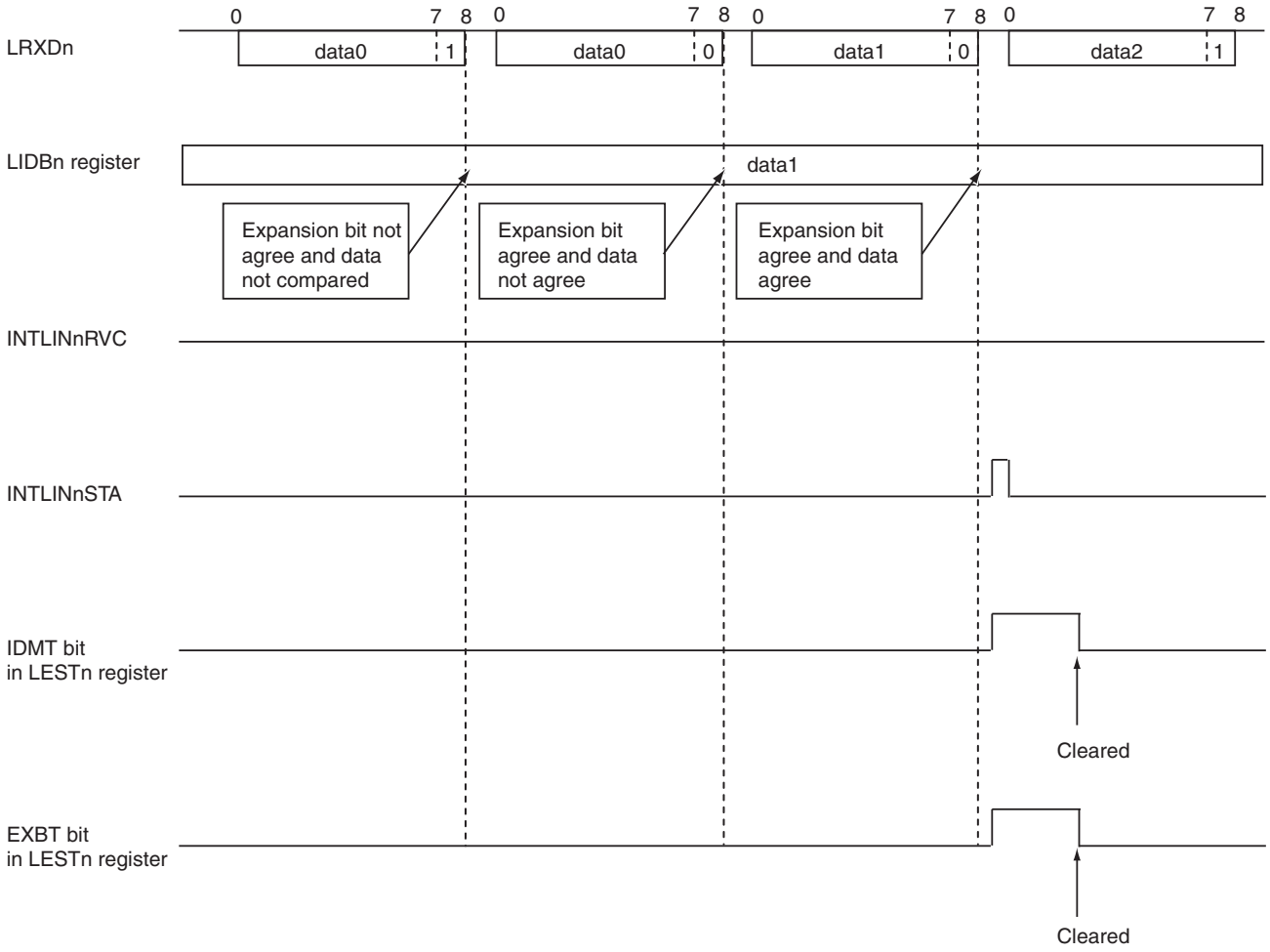
Even if the compared two values agree, a successful LINn reception interrupt is not generated.

If the compared two values do not agree, neither successful LINn reception interrupt nor LINn reception status interrupt is generated, thus not setting the EXBT or IDMT flag to 1. Here, the received data is not stored in the UART reception data register (LURDRn).

When changing the UEBDCE bit to 0, complete it before the next data is completely received.

Figure 17-30 shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.

**Figure 17-30. Expansion Bit Reception Example (with Data Comparison) (LSB First, UEBDL = 0)**



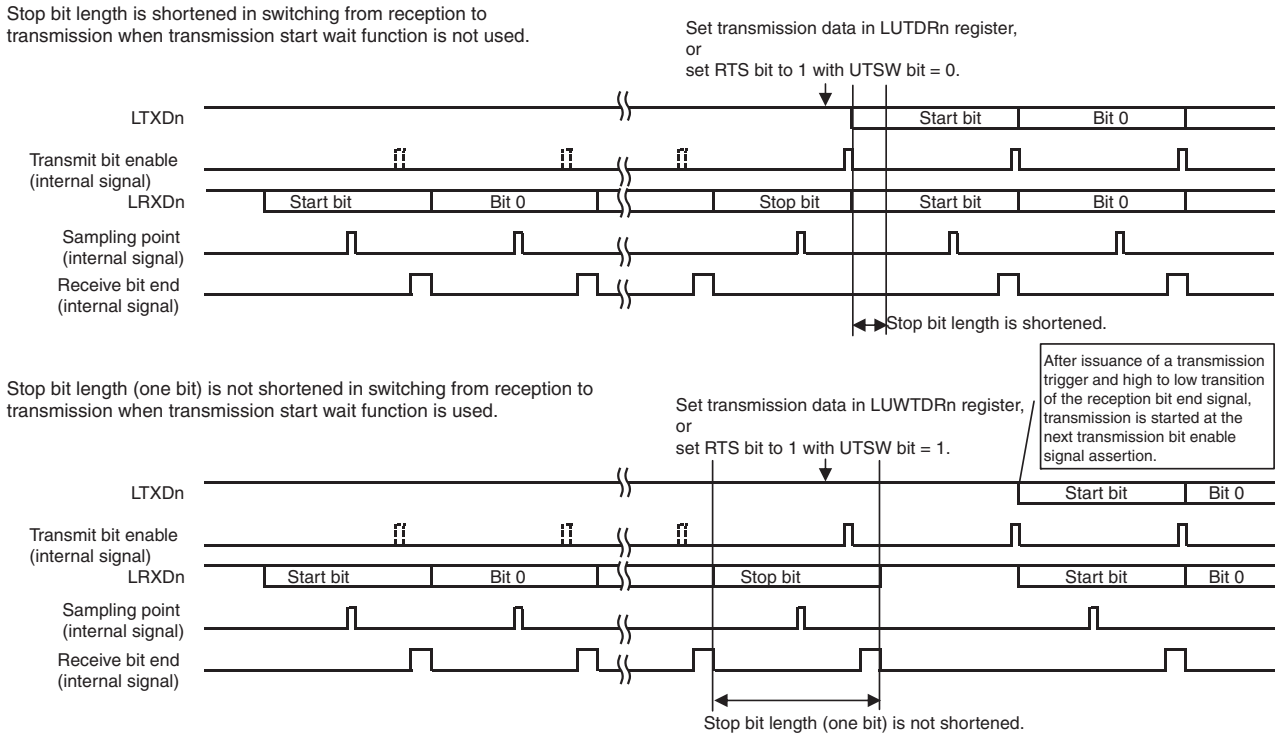

<R> **Caution If a reception error (parity error, framing error, or overrun error) occurs, a LINn reception status interrupt is generated, and the error flag is updated. In the case of an overrun error with matching of the compare result, EXBT and IDMT flags are also set to 1.**

### (4) Transmission Start Wait Function

For performing half-duplex communication, the LIN/UART module (in UART mode) has the function of securing the reception stop bit when switching from reception to transmission.
If it is desired to delay the start of transmission until the stop bits for the reception are completed, set data in the LUWTDRn register, which is used only for the wait function, instead of setting transmission data in the LUTDRn register as a start-of-transmission request. When transmitting from the UART buffer, set 1 (UART buffer transmission enabled) in the RST bit in the LTRCn register with 1 set in the UTSW bit in the LDFCn register.
In such a case, the LIN/UART module delays the start of transmission until the stop bits of reception data are completed.
It should be noted that even if the UART stop bit length select bit (USBLS) is 1 (stop bits = 2 bits), delay is made only for 1 bits.

Figure 17-31 shows the operation of transmission wait function.

**Figure 17-31. Transmission Wait Function (if transmission data is set during the stop bits in the received data)**



### (5) SNOOZE Mode Function

The LIN/UART module (UART mode) is provided with SNOOZE mode during reception. The SNOOZE mode allows data reception without CPU operation when the LRXDn pin input is detected in STOP mode.
To use the LIN/UART module (UART mode) in SNOOZE mode, make the following settings before entering STOP mode.

- In SNOOZE mode, it is necessary to set the different baud rate for UART reception from that in normal operation. Refer to tables 17-21 to 17-24 and set the LBRPn register and LPRS[2:0] bits and NSPB[3:0] bits in the LWBRn register appropriately.
- Set the UWC bit in the UART standby control register (LUSCn). Also set the USEC and URDCC bits in the LUSCn register to enable or disable error interrupt generation upon occurrence of a communication error and comparison of the received data and the LIDBn register value, respectively.
- Set the UROE bit to 1 in the UART operation enable register (LUOERn) immediately before entering STOP mode.

After entering STOP mode, UART reception starts upon detection of the LRXDn edge (start bit input).

**Cautions 1. SNOOZE mode can be set only when the LINnMCK bit in the LINCKSEL register is 0 ($f_{CLK}$ selected) and the high-speed on-chip oscillator clock ($f_{IH}$) is selected for $f_{CLK}$.**

**2. The maximum transfer rate in SNOOZE mode is 4800 bps when the FRQSEL4 in the user option byte (000C2H/020C2H) is set to 0, and 2400 bps when FRQSEL4 is set to 1.**

**3. With UWC = 1, the UART can be used only if reception is started during STOP mode.**
**If another SNOOZE function or interrupt is also used and reception is started during any state other than STOP mode as described below, data is not received correctly and a framing error or parity error may occur.**

- **After setting UWC to 1, reception is started before entering STOP mode.**
- **Reception is started during another SNOOZE mode.**
- **After returning to normal operation from STOP mode upon an interrupt or other cause, reception is started before setting UWC to 0.**

**4. With USEC = 1, if an error (parity error or framing error) or change in status (detection of the expansion bit) is detected in SNOOZE mode, the flag is not set to 1 thus generating no error interrupt.**

**Table 17-21. Baud Rate Setting for UART Reception in SNOOZE Mode**
**(LIN communication clock source = 32 MHz ± 2%, FRQSEL4 = 0)**

| UART Baud Rate (target) | Prescaler LPRS[2:0] | Baud Rate Generators 0 and 1 LBRP0 and LBRP1 | Maximum Allowable Value | Minimum Allowable Value |
|---|---|---|---|---|
| 1200 bps | 1/2 | 828 | 2.45% | -2.50% |
| 2400 bps | 1/2 | 412 | 2.07% | -2.30% |
| 4800 bps | 1/2 | 203 | 1.81% | -1.42% |

**Table 17-22. Baud Rate Setting for UART Reception in SNOOZE Mode**
**(LIN communication clock source = 24 MHz ± 2%, FRQSEL4 = 0)**

| UART Baud Rate (target) | Prescaler LPRS[2:0] | Baud Rate Generators 0 and 1 LBRP0 and LBRP1 | Maximum Allowable Value | Minimum Allowable Value |
|---|---|---|---|---|
| 1200 bps | 1/2 | 621 | 2.41% | -2.55% |
| 2400 bps | 1/2 | 308 | 2.31% | -2.08% |
| 4800 bps | 1/2 | 152 | 1.81% | -1.45% |

**Table 17-23. Baud Rate Setting for UART Reception in SNOOZE Mode**
**(LIN communication clock source = 32 MHz ± 2%, FRQSEL4 =1)**

| UART Baud Rate (target) | Prescaler LPRS[2:0] | Baud Rate Generators 0 and 1 LBRP0 and LBRP1 | Maximum Allowable Value | Minimum Allowable Value |
|---|---|---|---|---|
| 1200 bps | 1/2 | 826 | 2.23% | -2.26% |
| 2400 bps | 1/2 | 410 | 1.65% | -1.83% |

**Table 17-24. Baud Rate Setting for UART Reception in SNOOZE Mode**
**(LIN communication clock source = 24 MHz ± 2%, FRQSEL4 =1)**

| UART Baud Rate (target) | Prescaler LPRS[2:0] | Baud Rate Generators 0 and 1 LBRP0 and LBRP1 | Maximum Allowable Value | Minimum Allowable Value |
|---|---|---|---|---|
| 1200 bps | 1/2 | 619 | 2.28% | -2.23% |
| 2400 bps | 1/2 | 307 | 1.73% | -1.76% |

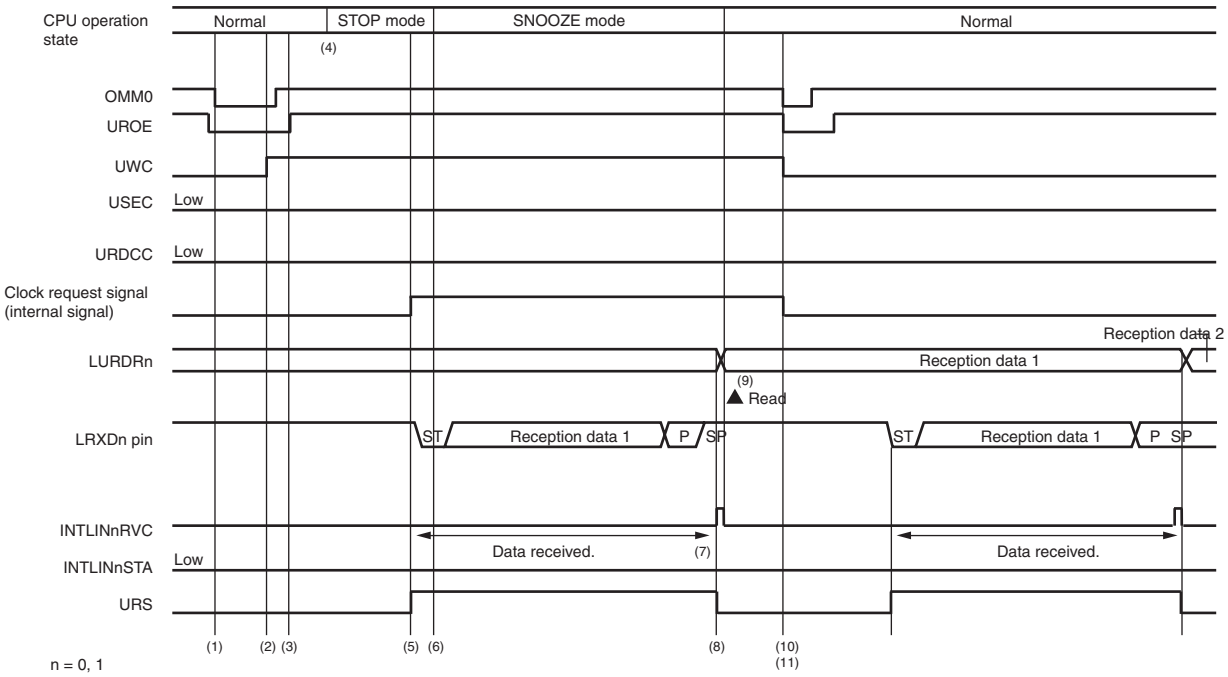
**Remarks 1.** The maximum and minimum allowable values are applied to the baud rates for UART reception. Set the parameters so that the baud rates for UART transmission should also fall within the allowable range.

**2.** The receive data length is 8 bits + a parity bit.

**3.** The numbers in the table are for sixteen samples per bit (i.e. when NSPB[3:0] = 0000b or 1111b).

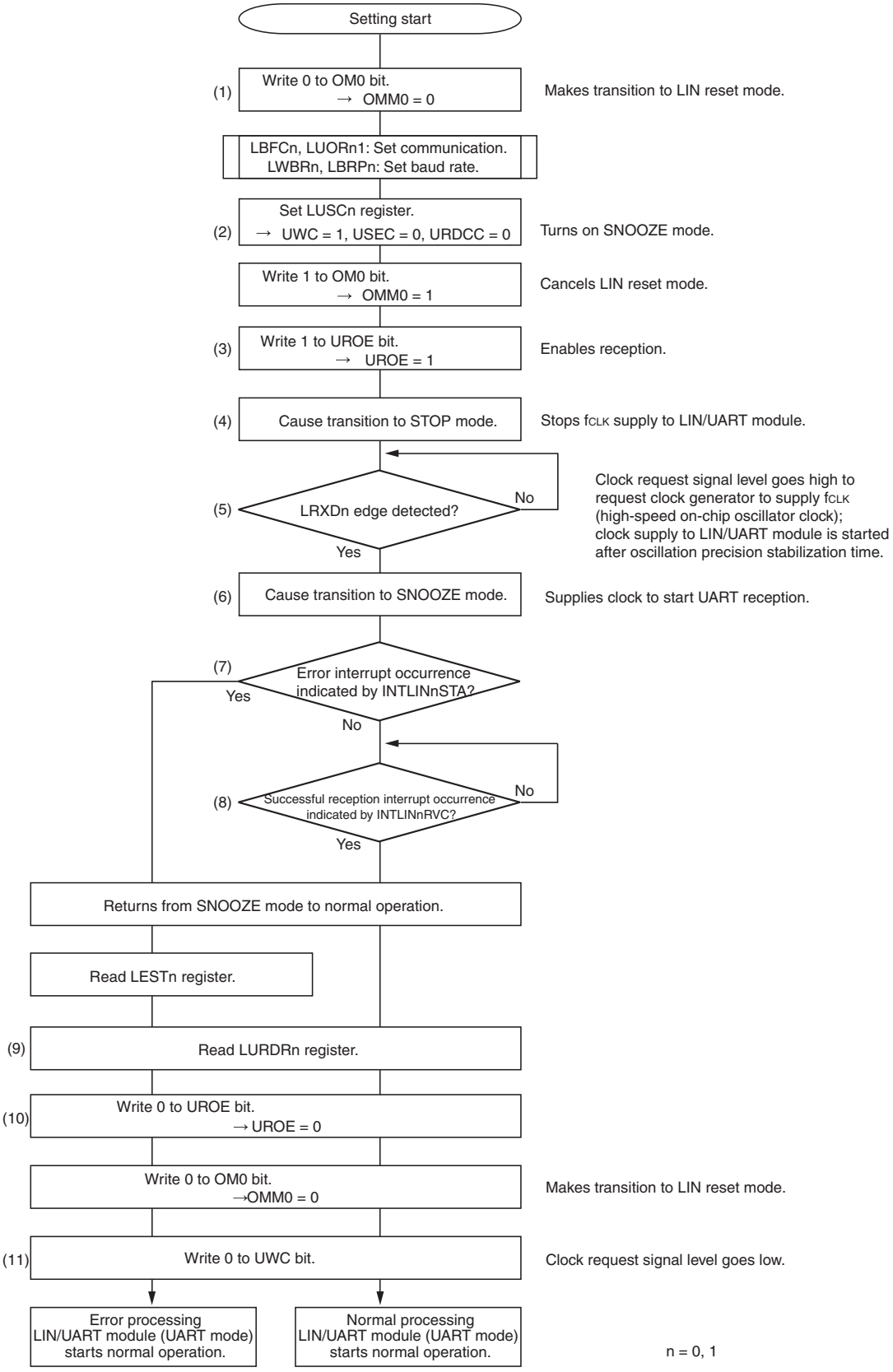
**(a) SNOOZE Mode Operation (returning to normal operation upon successful reception; UWC = 1, URDCC = 0)**

**Figure 17-32. Timing Chart of SNOOZE Mode Operation (returning to normal operation upon successful reception; UWC = 1, URDCC = 0)**




**Remark** (1) to (11) in Figure 17-32 correspond to (1) to (11) in Figure 17-33.

**Figure 17-33. Flowchart of SNOOZE Mode Operation**
**(returning to normal operation upon successful reception; UWC = 1, URDCC = 0)**




**Remark** (1) to (11) in Figure 17-33 correspond to (1) to (11) in Figure 17-32.

**(b) SNOOZE Mode Operation (returning to normal operation upon comparison result agreement of data received; UWC = 1, URDCC = 1)**

**Figure 17-34. Timing Chart of SNOOZE Mode Operation**
**(returning to normal operation upon comparison result agreement of data received; UWC = 1, URDCC = 1)**

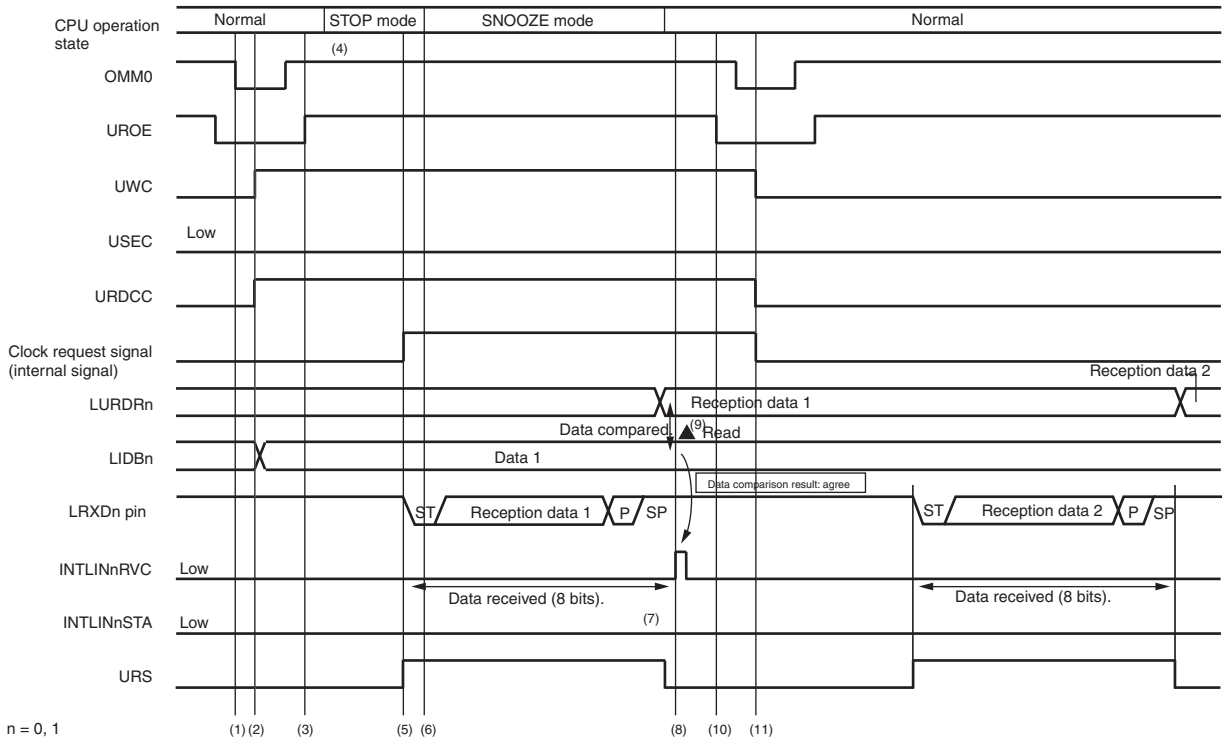
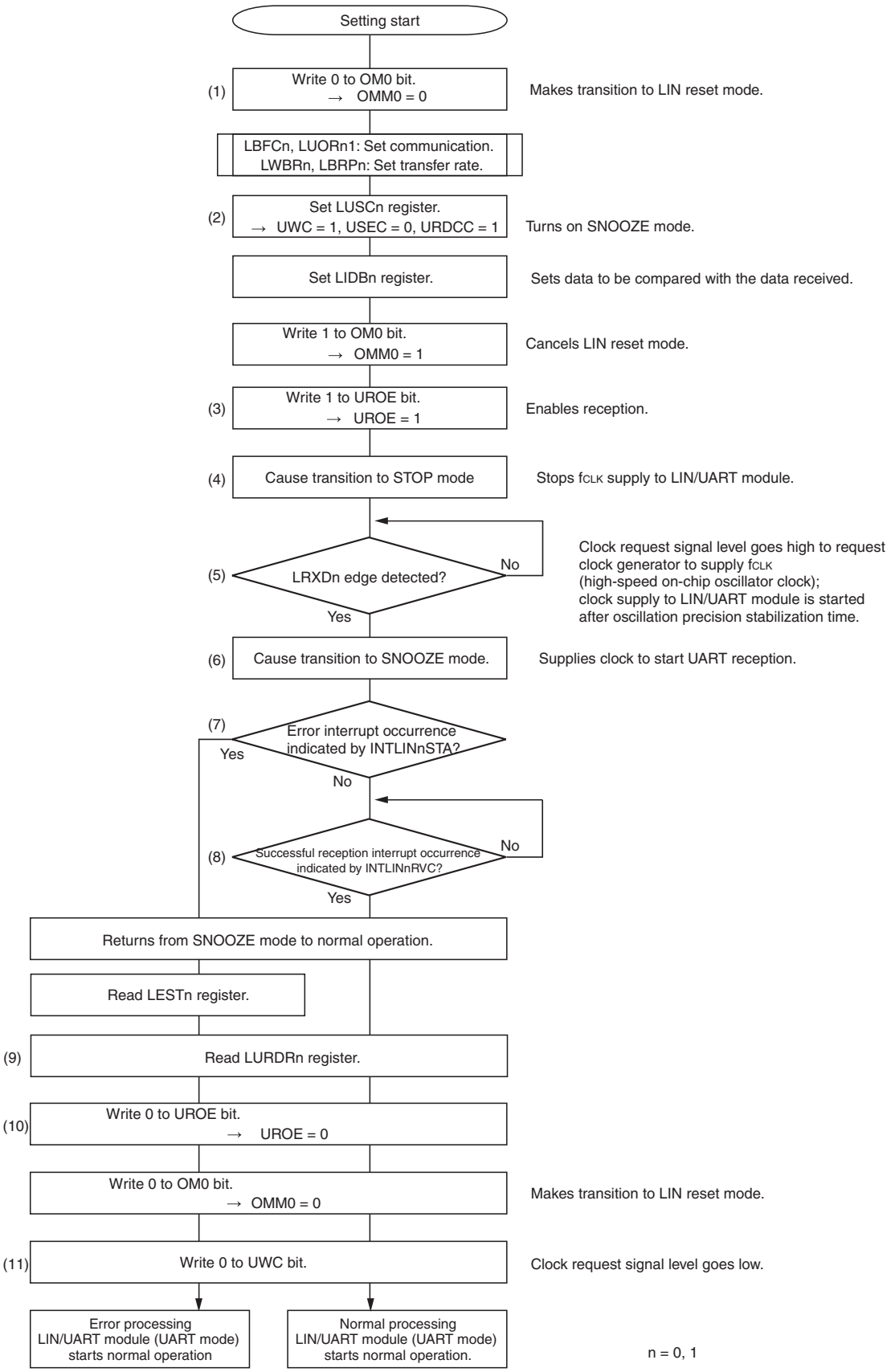


**Figure 17-35. Flowchart of SNOOZE Mode Operation**
**(returning to normal operation upon comparison result agreement of data received; UWC = 1, URDCC = 1)**



If it is necessary to set the URDCC bit in the LUSCn register to 1 (comparison of received data and LIDBn register data enabled in SNOOZE mode), only use SNOOZE mode with the UBLS bit in the LBFCn register set to 0 (UART 8-bit character communication) and UEBE bit in the LUORn1 register set to 0 (expansion bit operation disabled).

### 17.5.2 Data Transmission/Reception

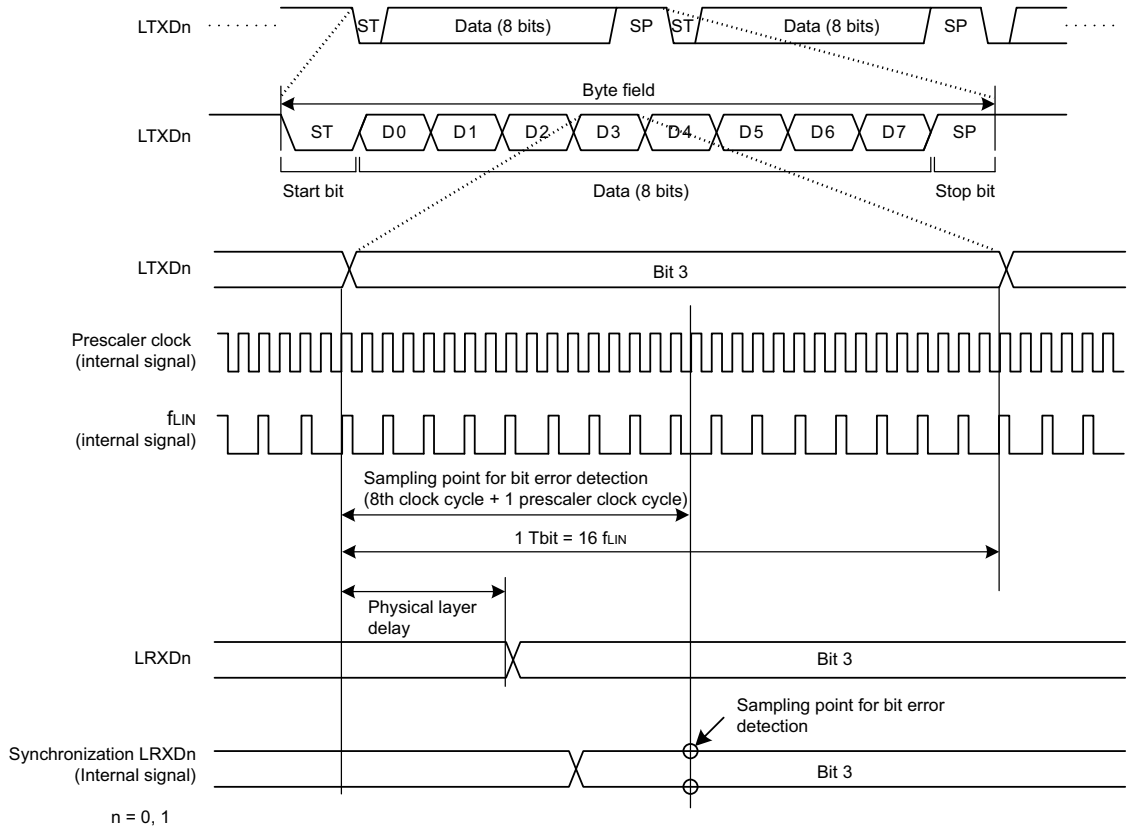
#### (1) Data Transmission

One bit of data is transmitted per Tbit.
In half-duplex communication, if the BERE bit in the LEDEn register is 1 (bit error detection enabled), the transmission data and the input pin level are compared bit by bit during data transmission, and the results are stored in the BER flag in the LESTn register (see **17.5.5 Error Status**). The timing at which the input pin is sampled during data transmission can vary depending upon the settings of the LPRS[2:0] and NSPB[3:0] bits in the LWBRn register.
The bit error detection timing in UART mode is shown in Table 17-25.

**Table 17-25. Error Detection Timing in UART Mode**

| Sampling count per bit | Bit error detection timing |
|---|---|
| 6 samples | 3rd clock cycle + one cycle of the prescaler clock |
| 7 samples | 4th clock cycle + one cycle of the prescaler clock |
| 8 samples | 4th clock cycle + one cycle of the prescaler clock |
| 9 samples | 5th clock cycle + one cycle of the prescaler clock |
| 10 samples | 5th clock cycle + one cycle of the prescaler clock |
| 11 samples | 6th clock cycle + one cycle of the prescaler clock |
| 12 samples | 6th clock cycle + one cycle of the prescaler clock |
| 13 samples | 7th clock cycle + one cycle of the prescaler clock |
| 14 samples | 7th clock cycle + one cycle of the prescaler clock |
| 15 samples | 8th clock cycle + one cycle of the prescaler clock |
| 16 samples | 8th clock cycle + one cycle of the prescaler clock |

**Figure 17-36. Example of Data Transmission Timing (when Sampling Count is 16 in 1 Tbit)**

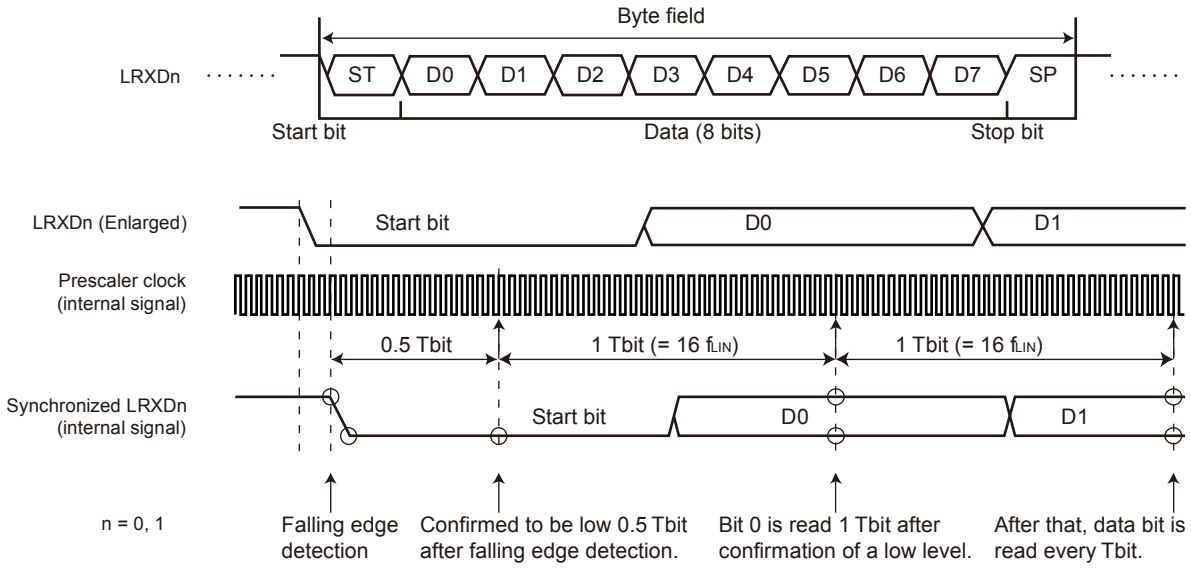


### (2) Data Reception

Data reception is performed by using the synchronized LRXDn (an internal signal) that is the input from the LRXDn pin synchronized with the prescaler clock.
The byte field is synchronized at the falling edge of the start bit for the synchronized LRXDn signal. After the falling edge is detected, resampling is performed 0.5 Tbits later if the sampling count per 1 Tbit is even and {(sampling count + 1)/2}/(sampling count) Tbits later if odd. If the synchronized LRXDn signal is low, the bit is recognized as a start bit. The bit is not recognized as a start bit if the LRXDn signal is fixed at low after the reset is cleared or if a high level is detected during the resampling.
After the start bit is detected, 1 bit is sampled per Tbit.
Note that when the BERE bit in the LEDEn register is set to 1, sampling proceeds at the same time as the detection of a bit error.
The LIN/UART module has a noise filter function with respect to received data. If the LRDNFS bit in the LMDn register is 0, the noise filter is used. For a sampling value, the value determined by a 3-sampling majority rule by the prescaler clock is used. If the LRDNFS bit in the LMDn register is 1, the noise filter is not used. In this case, for a sampling value, the synchronized LRXDn value at the sampling position is used as is.

Figure 17-37 shows an example of data reception timing.

**Figure 17-37. Example of Data Reception Timing (when Sampling Count is 16 in 1 Tbit)**

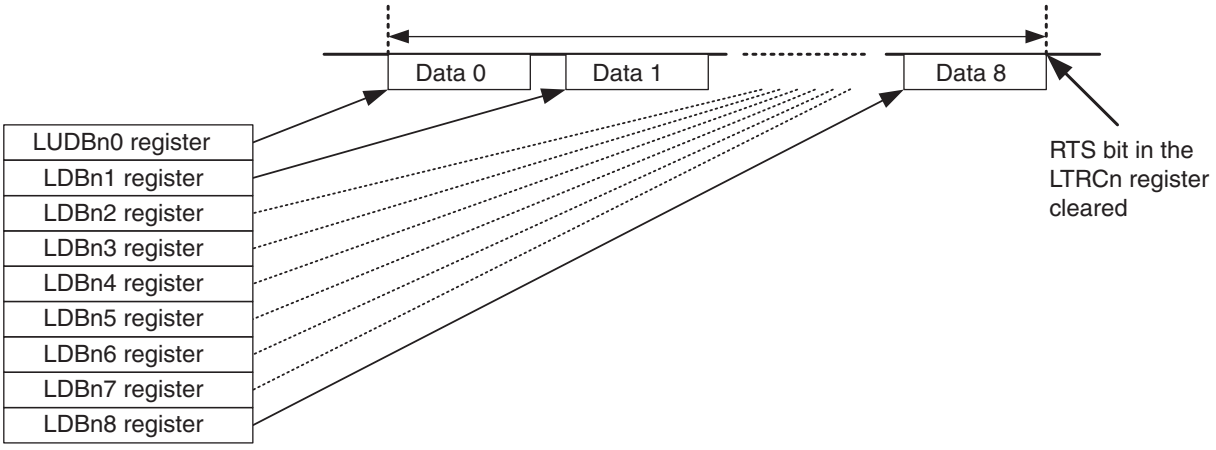


### 17.5.3 Buffer Processing of Transmission Data

#### (1) Transmission of UART Buffer

For a 9-byte transmission, the contents stored in the LUDBn0 and LDBn1 to LDBn8 registers are transmitted to data areas 0 to 8. The LUDBn0 register is used only if 9-byte transmission is set. In other cases, the LDBn1 to LDBn8 registers are selected depending upon the length of data involved. For a 4-byte transmission, the contents stored in the LDBn1 to LDBn4 registers are transmitted to data areas 1 to 4, but the contents of the LDBn5 to LDBn8 registers are not transmitted. A LINn transmission interrupt is generated after the transmission of the data that is set in the MDL[3:0] bits in the LDFCn register (n = 0, 1). The spaces between transmission data items can be set in the IBS bit in the LSCn register.
Figure 17-38 shows a 9-byte UART buffer and the transmission processing.

**Figure 17-38. UART Buffer and Transmission Processing (for 9-Byte Transmission)**



### 17.5.4 Status

In UART mode, the LIN/UART module can detect five types of statuses.
Two statuses, successful UART buffer transmission and error detection, can generate interrupt requests.
Table 17-26 shows the types of statuses available in UART mode.

**Table 17-26. Types of Statuses in UART Mode**

| Status | Status set condition | Status clear condition | Corresponding bit | Interrupt |
|---|---|---|---|---|
| Reset | After the OM0 bit in the LCUCn register is set to not-LIN–reset-mode, if actually the LIN/UART module is cleared from LIN reset mode. | After the OM0 bit in the LCUCn register is set to LIN reset mode, if actually the LIN/UART module enters LIN reset mode. | OMM0 bit in LMSTn register | Not available |
| Successful UART buffer transmission | • The transmission of the last data of data equal to the length set in the MDL bits in the LDFCn register is started while the UTIGTS bit in the LUORn1 register is 0 (transmission interrupt is generated at the start of transmission).<br>• The transmission of data equal to the length set in the MDL bits in the LDFCn register is completed while the UTIGTS bit in the LUORn1 register is 1 (transmission interrupt is generated at the completion of transmission). | • When cleared by software<br>• After transition to LIN reset mode | FTC flag in LSTn register | Available |
| Error detection | If any of the UPER flag, IDMT flag, EXBT flag, FER flag, OER flag, and BER flags in the LESTn register turns 1 (error detected). | • When cleared by software[Note]<br>• After transition to LIN reset mode | ERR flag in LSTn register | Available |
| Transmission status | • When data is written to the LUTDRn or LUWTDRn register.<br>• When a 1 is written to the RTS bit in the LTRCn register. | • The transmission of the data set in the LUTDRn or LUWTDRn register is complete, but another transmission data item is not set<br>• The transmission of the data in the UART buffer is complete, and the RTS bit in the LTRCn register is cleared<br>• After transition to LIN reset mode | UTS flag in LSTn register | Not available |
| Reception status | • When a start bit is detected. | • When a sampling point for stop bits is detected<br>• After transition to LIN reset mode | URS flag in LSTn register | Not available |

**Note** Writing a 0 to the UPER, IDMT, EXBT, FET, OER, and BER flags in the LESTn register when the LIN reset mode is being canceled turns the ERR flag in the LSTn register to 0.

### 17.5.5 Error Status

In UART mode, the LIN/UART module can detect four types of errors and two types of statuses. The condition of these error statuses can be checked by means of the corresponding bits in the LESTn register.
Table 17-27 lists applicable error status types.

**Table 17-27. Types of Error Statuses in UART Mode**

| Status | Error detection condition | Communication | Enable/disable detection | Corresponding bit |
|---|---|---|---|---|
| Bit error | The transmitted data and the data monitored on the receive pin do not match Note 1 | Continues until the transmission of the set transmission data is finished. | O | BER flag in LESTn register |
| Overrun error | After received data is stored in the LURDRn register, another data item is received before the data is read.<br>(In this case, no data is stored in the LURDRn register). | –<br>(Reception is finished by the time this error is detected) | O | OER flag in LESTn register |
| Framing error | When the first stop bit at the first bit is low in the reception processing. | –<br>(Reception is finished by the time this error is detected) | O | FER flag in LESTn register |
| Parity error | The received parity value fails to match the parity value calculated from the received data | Continues until the data reception is finished. | ×Note 2 | UPER flag in LESTn register |
| Expansion bit detection | The value of the received expansion bit matches the value of the UEBDL bit in the LUORn1 register. | — | O | EXBT flag in LESTn register |
| ID match | The value of the received expansion bit matches the value of the UEBDL bit in the LUORn1 register and the 8-bit received data excluding the expansion bit matches the value of the LIDBn register. | — | O | IDMT flag in LESTn register |

**Notes 1.** If data is transmitted from the UART buffer, a bit error is also detected in the space between UART frames (inter-byte space).

**2.** Setting the UPS[1:0] bits in the LBFCn register to 10b (0 parity) disables the checking of parity bit values. In this case, no parity error is generated.

**Caution The error status is cleared when cleared by software or after transition to LIN reset mode.**

## 17.6 LIN Self-Test Mode

The LIN/UART module has a LIN self-test mode. When LIN self-testing is turned on, the LTXDn and LRXDn signals are disconnected from the external pins and connected within the LIN/UART module. Frames transferred from the internal LTXDn terminal loop back to the internal LRXDn terminal. The LIN self-test mode is exclusively for testing LIN mode operation.
The following four types of tests are available:

- LIN master self-test mode (transmission): header transmission and response transmission
- LIN master self-test mode (reception): header transmission and response reception
- LIN slave self-test mode (transmission): header reception and response transmission
- LIN slave self-test mode (reception): header reception and response reception

In the LIN self-test mode, the LIN/UART module operates at the highest baud rate regardless of the setting of the baud rate generator. The baud rate is <frequency of the LIN communications clock source>/16 bps regardless of the settings of the baud rate related registers (the NSPB bit in the LWBRn register must be set to 0000b or 1111b).
In the LIN self-test mode, the following functions are not supported. Do not use these functions.

- LIN wake-up mode
- Frame separate mode
- Multi-byte response reception and transmission
- LIN slave mode [auto baud rate]
- Frame/response timeout error

**Figure 17-39. Connection in LIN Reset Mode, LIN Mode and UART Mode**

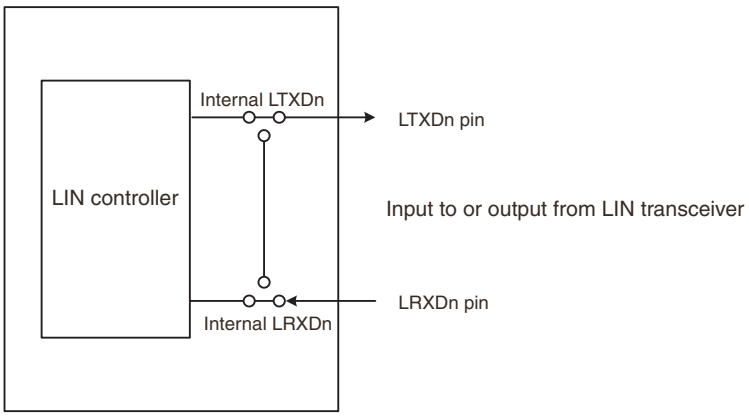
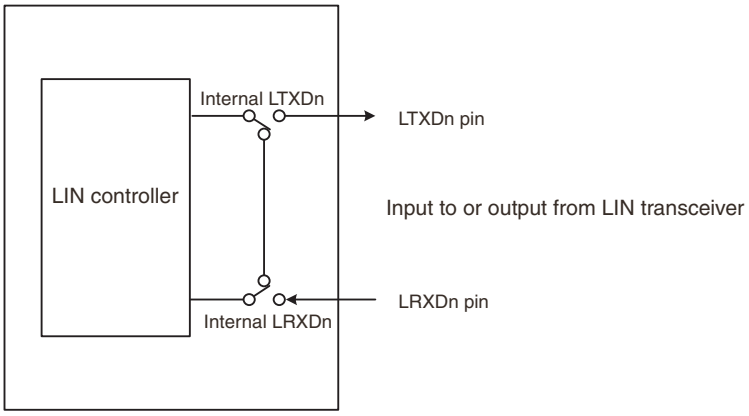



**Figure 17-40. Connection in LIN Self-Test Mode**



### 17.6.1 Change to LIN Self-Test Mode

LIN self-test mode is entered by writing to the LSTCn register.
The transition to LIN self-test mode can be confirmed when the LSTM bit in the LSTCn register becomes 1.
When changing to LIN self-test mode, be sure to execute a specific sequence. In that sequence, information must be written three times consecutively to the LIN self-test control register, as follows:

- Change to LIN reset mode
  Set the OM0 bit in the LCUCn register to 0 (LIN reset mode).
  Read the OMM0 bit in the LMSTn register; verify that it is 0 (LIN reset mode).
- Select a LIN mode
  LMD bits in LMDn register = 00b (LIN master mode) or 11b (LIN slave mode [fixed baud rate])
- 1st write: LSTCn register = 1010 0111b (A7H)
- 2nd write: LSTCn register = 0101 1000b (58H)
- 3rd write: LSTCn register = 0000 0001b (01H)
- Verify the transition to LIN self-test mode

  Read the LSTM bit in the LSTCn register; verify that it is 1 (LIN self-test mode).

If the key of the first write (A7H) is written twice by mistake, the transition to LIN self-test mode is canceled. The above sequence should be retried from the step of first write. In addition, if a write to another LIN-related register is performed during transition to LIN self-test mode (three consecutive write operations to the LSTCn register), the transition is also canceled.

### 17.6.2 Transmission in LIN Master Self-Test Mode

To execute a self-test on LIN master transmission, perform the procedure below:

- Set registers related to the baud rate, noise filter, and interrupt output.
  LWBRn register = 0000xxxxb[Note 1]
  LBRPn0 register = xxxxxxxxb[Note 1]
  LBRPn1 register = xxxxxxxxb[Note 1]
  LMDn register = 00xxxx00b[Note 1, 3]
- Set registers related to interrupt enabling and error enabling.
  LIEn register = 0000xxxxb[Note 2, 3]
  LEDEn register = x000x0xxb
- Set registers related to the break field and spaces.
  LBFCn register = 00xxxxxxb
  LSCn register = 00xx0xxxb
- Cancel the reset.
  Write 11b to the OM1 and OM0 bits in the LCUCn register and verify that the OMM1 and OMM0 bits in the LMSTn register become 11b.
- Set registers related to the transmission frame.
  LDFCn register = 00x1xxxxb
  LIDBn register = xxxxxxxxb
  LDBn1 to LDBn8 registers = xxxxxxxxb
- Start header transmission followed with response transmission
  LDFCn register = 00x1xxxxb
  Set the FTS bit in the LTRCn register to 1 (frame transmission or wake-up transmission/reception started).

<R> The LIN master self-test mode (transmission) is executed. In this mode, interrupts are generated, and status and error status are also updated appropriately. The checksum is automatically computed by the LIN/UART module.
When the execution of LIN master self-test mode (transmission) is aborted, set OM0 bit of LCUCn register to 0 (LIN reset mode).

- When the transmission is completed, the reversed value of the looped-back frame data is stored in the LIDBn, LDBnm (m = 1 to 8), and LCBRn registers (the data is reversed before being stored because the transmitted value should be compared with the looped-back value). The FTS bit in the LTRCn register is cleared.
- If the transmission fails to complete due to an error, the applicable error flag is set and the FTS bit in the LTRCn register is cleared.

**Notes 1.** The following register settings do not affect the operations in LIN self-test mode. Therefore, setting them is not mandatory.
LPRS bits in LWBRn register, LBRPn0 register, LBRPn1 register, and LCKS bits in LMDn register

**2.** As necessary, set the related registers in **CHAPTER 21 INTERRUPT FUNCTIONS**.

**3.** When the successful header transmission interrupt and successful frame transmission interrupt are used in the same interrupt, the SHIE bit in the LIEn register should not be set to 1 (successful header transmission interrupt enabled) if the software processing of the successful header transmission interrupt does not complete before the successful frame transmission interrupt is generated.
The period starting from when the successful header transmission flag is set until the successful frame/wake-up transmission flag is set can be calculated as follows:

$10 \times$ (number of data bytes + 1) [Tbit]

1 Tbit = LIN communication clock source $\times$ 16

**Remark** x: Any desired value

### 17.6.3 Reception in LIN Master Self-Test Mode

To execute a self-test on LIN master reception, perform the procedure below:

- Set registers related to the baud rate, noise filter, and interrupt output.
  LWBRn register = 0000xxxxb[Note 1]
  LBRPn0 register = xxxxxxxxb[Note 1]
  LBRPn1 register = xxxxxxxxb[Note 1]
  LMDn register = 00xxxx00b[Note 1, 3]
- Set registers related to interrupt enabling and error enabling.
  LIEn register = 0000xxxxb[Note 2, 3]
  LEDEn register = x000x0xxb
- Set registers related to the break field and spaces.
  LBFCn register = 00xxxxxxb
  LSCn register = 00xx0xxxb[Note 1]
- Cancel the reset.
  Write 11b to the OM1 and OM0 bits in the LCUCn register and verify that the OMM1 and OMM0 bits in the LMSTn register become 11b.
- Set registers related to the reception frame.
  LDFCn register = 00x0xxxxb
  LIDBn register = xxxxxxxxb
  LDBn1 to LDBn8 registers = xxxxxxxxb
  LCBRn register = xxxxxxxxb

Since the checksum is not computed automatically, store a computed value. By intentionally setting an incorrect computation result as the checksum, a test for checksum errors can be performed.

- Start header transmission followed with response reception
  Set the FTS bit in the LTRCn register to 1 (frame transmission or wake-up transmission/reception started).

<R> The LIN master self-test mode (reception) is executed. In this mode, interrupts are generated, and status and error status are also updated appropriately. When the execution of LIN master self-test mode (reception) is aborted, set OM0 bit of LCUCn register to 0 (LIN reset mode).

- When the reception is completed, the reversed value of the looped-back frame data is stored in the LIDBn, LDBnm (m = 1 to 8), and LCBRn registers (the data is reversed before being stored because the set value should be compared with the looped-back and received value). The FTS bit in the LTRCn register is cleared.
- If the reception fails to complete due to an error, the applicable error flag is set and the FTS bit in the LTRCn register is cleared.

**Notes 1.** The following register settings do not affect the operations in LIN self-test mode. Therefore, setting them is not mandatory.
LPRS bits in LWBRn register, LBRPn0 register, LBRPn1 register, LCKS bits in LMDn register, and IBS bits in LSCn register

**2.** As necessary, set the related registers in **CHAPTER 21 INTERRUPT FUNCTIONS**.

**3.** When the successful header transmission interrupt and successful frame transmission interrupt are used in the same interrupt, the SHIE bit in the LIEn register should not be set to 1 (successful header transmission interrupt enabled) if the software processing of the successful header transmission interrupt does not complete before the successful frame transmission interrupt is generated.
The period starting from when the successful header transmission flag is set until the successful frame/wake-up transmission flag is set can be calculated as follows:

$10 \times$ (number of data bytes + 1) [Tbit]

1 Tbit = LIN communication clock source $\times$ 16.

**Remark** x: Any desired value

### 17.6.4 Transmission in LIN Slave Self-Test Mode

To execute a self-test on LIN slave transmission, perform the procedure below:

- Set registers related to the baud rate, noise filter, and interrupt output.
  LWBRn register = 0000xxx0b[Note 1]
  LBRPn0 register = xxxxxxxxb[Note 1]
  LBRPn1 register = xxxxxxxxb[Note 1]
  LMDn register = 00xx0011b [Note 4]
- Set registers related to interrupt enabling and error enabling.
  LIEn register = 0000xxxxb[Note 2, 4]
  LEDEn register = xx0xx00xb
- Set registers related to the break field and spaces.
  LBFCn register = 0000000xb[Note 3]
  LSCn register = 00xx0001b
- Cancel the reset.
  Write 11b to the OM1 and OM0 bits in the LCUCn register and verify that the OMM1 and OMM0 bits in the LMSTn register become 11b.
- Set registers related to the transmission frame.
  LDFCn register = 00x1xxxxb
  LIDBn register = xxxxxxxxb
  LDBn1 to LDBn8 registers = xxxxxxxxb
- Start header reception followed with response transmission
  Set the FTS bit in the LTRCn register to 1 (header reception or wake-up transmission/reception started).
  (Without any operation involving the RTS bit in the LTRCn register, the reception of a header and the transmission of a response are executed, in the indicated order.)

<R> The LIN slave self-test mode (transmission) is executed. In this mode, interrupts are generated, and status and error status are also updated appropriately. The checksum is automatically computed by the LIN/UART module.
When the execution of LIN master self-test mode (transmission) is aborted, set OM0 bit of LCUCn register to 0 (LIN reset mode).

- When the transmission is completed, the reversed value of the looped-back frame data is stored in the LIDBn, LDBnm (m = 1 to 8), and LCBRn registers (the data is reversed before being stored because the transmitted value should be compared with the looped-back value). The FTS bit in the LTRCn register is cleared.
- If the transmission fails to complete due to an error, the applicable error flag is set and the FTS bit in the LTRCn register is cleared.

**Notes 1.** The following register settings do not affect the operations in LIN self-test mode. Therefore, setting them is not mandatory.
LPRS bits in LWBRn register, LBRPn0 register, and LBRPn1 register

**2.** As necessary, set the related registers in **CHAPTER 21 INTERRUPT FUNCTIONS**.

**3.** A break with a width of 9.5 or 10.5 Tbits is output from the internal LTXDn pin depending on this register setting.

**4.** When the successful header reception interrupt and successful response transmission interrupt are used in the same interrupt, the SHIE bit in the LIEn register should not be set to 1 (successful header reception interrupt enabled) if the software processing of the successful header reception interrupt does not complete before the successful response transmission interrupt is generated.
The period starting from when the successful header reception flag is set until the successful response/wake-up transmission flag is set can be calculated as follows:

10 × (number of data bytes + 1) [Tbit]

1 Tbit = LIN communication clock source × 16.

**Remark** x: Any desired value

### 17.6.5 Reception in LIN Slave Self-Test Mode

To execute a self-test on LIN slave reception, perform the procedure below:

- Set registers related to the baud rate, noise filter, and interrupt output.
  LWBRn register = 0000xxx0b[Note 1]
  LBRPn0 register = xxxxxxxxb[Note 1]
  LBRPn1 register = xxxxxxxxb[Note 1]
  LMDn register = 00xx0011b [Note 4]
- Set registers related to interrupt enabling and error enabling.
  LIEn register = 0000xxxxb[Note 2, 4]
  LEDEn register = xx0xx00xb
- Set registers related to the break field and spaces.
  LBFCn register = 0000000xb[Note 3]
  LSCn register = 00xx0001b[Note 1]
- Cancel the reset.
  Write 11b to the OM1 and OM0 bits in the LCUCn register and verify that the OMM1 and OMM0 bits in the LMSTn register become 11b.
- Set registers related to the reception frame.
  LDFCn register = 00x0xxxxb
  LIDBn register = xxxxxxxxb
  LDBn1 to LDBn8 registers = xxxxxxxxb
  LCBRn register = xxxxxxxxb

Since the checksum is not computed automatically, store a computed value. By intentionally setting an incorrect computation result as the checksum, a test for checksum errors can be performed.

- Start header reception followed with response reception
  Set the FTS bit in the LTRCn register to 1 (header reception or wake-up transmission/reception started).
  (Without any operation involving the RTS bit in the LTRCn register, the reception of a header and the reception of a response are executed, in the indicated order.)

<R> The LIN slave self-test mode (reception) is executed. In this mode, interrupts are generated, and status and error status are also updated appropriately. When the execution of LIN master self-test mode (reception) is aborted, set OM0 bit of LCUCn register to 0 (LIN reset mode).

- When the reception is completed, the reversed value of the looped-back frame data is stored in the LIDBn, LDBnm (m = 1 to 8), and LCBRn registers (the data is reversed before being stored because the set value should be compared with the looped-back and received value). The FTS bit in the LTRCn register is cleared.
- If the reception fails to complete due to an error, the applicable error flag is set and the FTS bit in the LTRCn register is cleared.

**Notes 1.** The following register settings do not affect the operations in LIN self-test mode. Therefore, setting them is not mandatory.
LPRS bits in LWBRn register, LBRPn0 register, LBRPn1 register, and IBS bits in LSCn register

**2.** As necessary, set the related registers in **CHAPTER 21 INTERRUPT FUNCTIONS**.

**3.** A break with a width of 9.5 or 10.5 Tbits is output from the internal LTXDn pin depending on this register setting.

**4.** When the successful header reception interrupt and successful response reception interrupt are used in the same interrupt, the SHIE bit in the LIEn register should not be set to 1 (successful header reception interrupt enabled) if the software processing of the successful header reception interrupt does not complete before the successful response reception interrupt is generated.
The period starting from when the successful header reception flag is set until the successful response/wake-up reception flag is set can be calculated as follows:

$10 \times$ (number of data bytes + 1) [Tbit]

1 Tbit = LIN communication clock source $\times$ 16

**Remark** x: Any desired value

### 17.6.6 Terminating LIN Self-Test Mode

To terminate LIN self-test mode, perform the procedure below:

- Write 0 (LIN reset mode) to the OM0 bit in the LCUCn register.
  If the OMM1 and OMM0 bits in the LMSTn register are not 11b, write 11b to the OM1 and OM0 bits in the LCUCn register. After confirming that the OMM1 and OMM0 bits in the LMSTn register have turned 11b, change to LIN reset mode.
- Verify the cancellation of LIN self-test mode.
  Read the LSTM bit in the LSTCn register; confirm that it is not 0 (not in LIN self-test)
- Verify the transition to LIN reset mode.
  Read the OMM0 bit in the LMSTn register; verify that it is 0 (LIN reset mode).

## 17.7 Baud Rate Generator

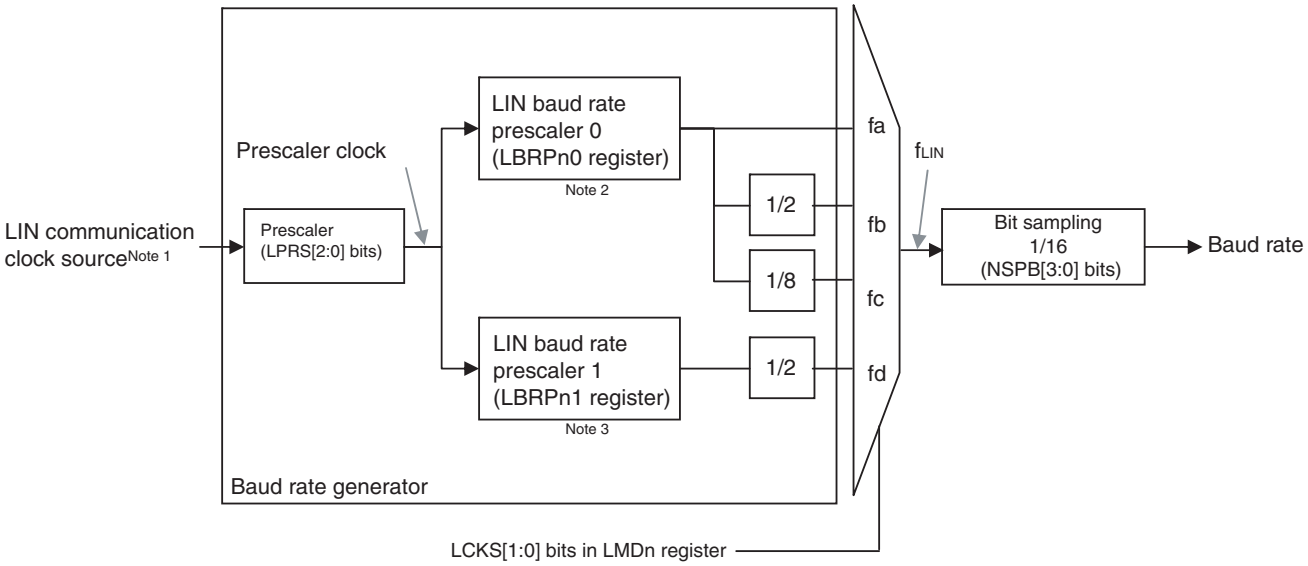
The prescaler clock is obtained by frequency-dividing the LIN communication clock source by the prescaler, and the LIN system clock ($f_{LIN}$) is obtained by frequency-dividing the prescaler clock by the baud rate generator. The clock obtained by frequency-dividing the LIN system clock ($f_{LIN}$) by the number of samples is the baud rate. The reciprocal of this baud rate is called the bit time (Tbit).

The LIN/UART module has two kinds of baud rate generators. The baud rate generators switch over according to the mode used.

### 17.7.1 LIN Master Mode

Figure 17-41 shows a block diagram of baud rate generation in LIN master mode.

**Figure 17-41. Block Diagram of Baud Rate Generation in LIN Master Mode**




Notes 1. For the LIN communication clock source, refer to CHAPTER 5 CLOCK GENERATOR.
2. When the value in the LBRPn0 register is N (N = 0 to 255), the clock frequency is divided by N+1.
3. When the value in the LBRPn1 register is M (M = 0 to 255), the clock frequency is divided by M+1.

Set the LIN communications clock source as follows.

<R>
- LIN communications clock source = $f_{CLK}$ Note 1
- In the range from 4 MHz to 32 MHz

**Note 1.** When the timeout error detection is not used, the $f_{MX}$ clock is selectable as the LIN communication clock source. In that case, set at least 1.2 times the frequency of the LIN communication clock source to the CPU/peripheral hardware clock($f_{CLK}$).

By setting the LBRPn0 register so that fa is 307200 Hz (= 19200 × 16), the resulting bit rates are fa = 19200 × 16, fb = 9600 × 16 and fc = 2400 × 16. These bit rates are frequency-divided by 16 in the bit timing generator, enabling bit rates of 19200 bps, 9600 bps and 2400 bps to be generated. Also, by setting the LBRPn1 register so that fd is 166672 Hz (= 10417 × 16), the resulting bit rate is fd = 10417 × 16. This bit rate is frequency-divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

Table 17-28 shows examples of baud rate (19200, 10417, 9600, and 2400 bps) generation for each LIN communication clock source frequency, and also the corresponding errors.

**Table 17-28. Examples of Baud Rate (19200, 10417, 9600, and 2400 bps) Generation in LIN Master Mode**

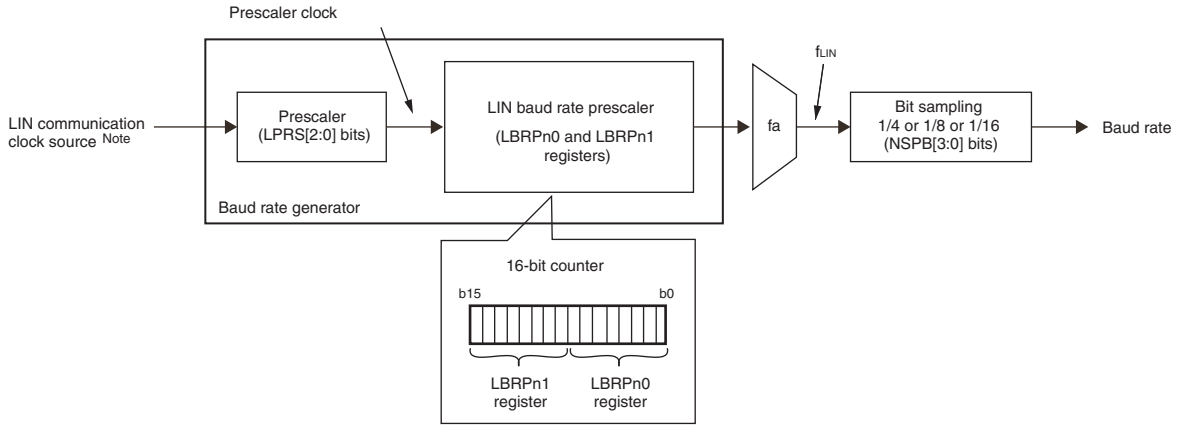
| LIN communication clock source | Prescaler | Baud rate generator 0 N+1 frequency-divided | Baud rate generator 1 M+1 frequency-divided | LIN system clock | Baud rate | Error |
|---|---|---|---|---|---|---|
| 32 MHz | 1/1 | 104 | — | fa | 19230.77 | +0.16% |
| | | — | 96 | fd | 10416.67 | -0.003% |
| | | 104 | — | fb | 9615.38 | +0.16% |
| | | 104 | — | fc | 2403.85 | +0.16% |
| 24 MHz | 1/1 | 78 | — | fa | 19230.77 | +0.16% |
| | | — | 72 | fd | 10416.67 | -0.003% |
| | | 78 | — | fb | 9615.38 | +0.16% |
| | | 78 | — | fc | 2403.85 | +0.16% |
| 16 MHz | 1/1 | 52 | — | fa | 19230.77 | +0.16% |
| | | — | 48 | fd | 10416.67 | -0.003% |
| | | 52 | — | fb | 9615.38 | +0.16% |
| | | 52 | — | fc | 2403.85 | +0.16% |
| 12 MHz | 1/1 | 39 | — | fa | 19230.77 | +0.16% |
| | | — | 36 | fd | 10416.67 | -0.003% |
| | | 39 | — | fb | 9615.38 | +0.16% |
| | | 39 | — | fc | 2403.85 | +0.16% |
| 8 MHz | 1/1 | 26 | — | fa | 19230.77 | +0.16% |
| | | — | 24 | fd | 10416.67 | -0.003% |
| | | 26 | — | fb | 9615.38 | +0.16% |
| | | 26 | — | fc | 2403.85 | +0.16% |

**Remark** The numbers in the table are for sixteen samples per bit (i.e. when NSPB[3:0] = 0000b or 1111b).

### 17.7.2 LIN Slave Mode

Figure 17-42 shows a block diagram of baud rate generation in LIN slave mode.

**Figure 17-42. Block Diagram of Baud Rate Generation in LIN Slave Mode**




Note For the LIN communication clock source, refer to CHAPTER 5 CLOCK GENERATOR.

Set the LIN communications clock source as follows.

<R>
- LIN communications clock source = $f_{CLK}$Note1
- In the range from 4 MHz to 32 MHz

**Note 1.** When the timeout error detection is not used, the $f_{MX}$ clock is selectable as the LIN communication clock source. In that case, set at least 1.2 times the frequency of the LIN communication clock source to the CPU/peripheral hardware clock($f_{CLK}$).

In LIN slave mode [Auto Baud Rate], operation is possible with the baud rates from 1 kbps to 20 kbps. Set the prescaler clock according to the target baud rate so that its frequency is the corresponding value from the list.

| [Target baud rate] | [Frequency of prescaler clock] |
|---|---|
| 1 kbps to 20 kbps: | 4 MHz **Note** |
| 1 kbps to less than 2.4 kbps: | 4 MHz |
| 2.4 kbps to 20 kbps: | 8 MHz to 12 MHz |

**Note** Set the NSPB[3:0] bits in the LWBRn register to 0011b (4 sampling).

Table 17-29 shows the examples of baud rate (19200, 10417, 9600, and 2400 bps) generation for each LIN communication clock source frequency in LIN slave mode [fixed baud rate] and errors.

**Table 17-29. Examples of Baud Rate Generation (19200 bps, 10417 bps, 9600 bps and 2400 bps) in LIN Slave Mode [Fixed Baud Rate]**

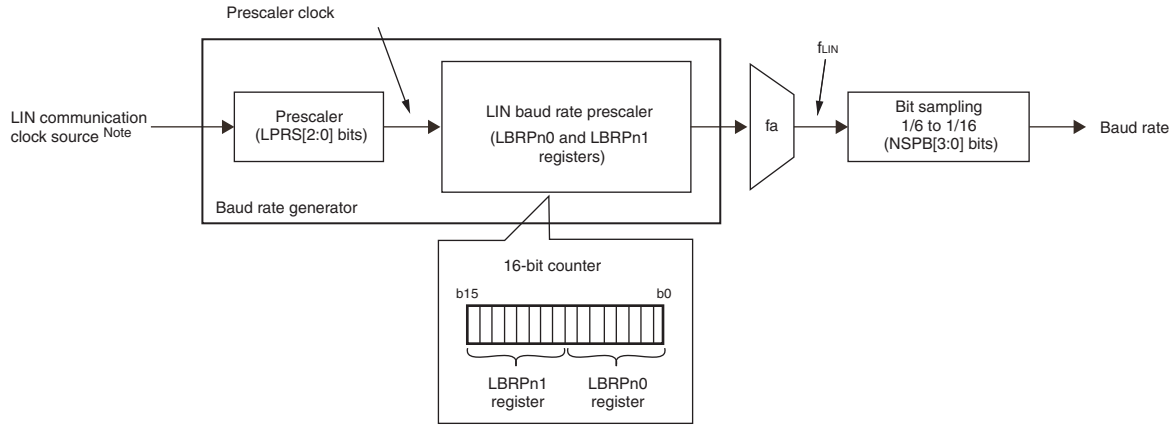
| LIN communication clock source | Prescaler | Baud rate generator 0-1 N+1 frequency-divided | Baud rate | Error |
|---|---|---|---|---|
| 32 MHz | 1/1 | 104 | 19230.77 | +0.16% |
| | | 192 | 10416.67 | -0.003% |
| | | 208 | 9615.38 | +0.16% |
| | | 833 | 2400.96 | +0.04% |
| 24 MHz | 1/1 | 78 | 19230.77 | +0.16% |
| | | 144 | 10416.67 | -0.003% |
| | | 156 | 9615.38 | +0.16% |
| | | 625 | 2400 | 0% |
| 16 MHz | 1/1 | 52 | 19230.77 | +0.16% |
| | | 96 | 10416.67 | -0.003% |
| | | 104 | 9615.38 | +0.16% |
| | | 417 | 2398.08 | -0.08% |
| 12 MHz | 1/1 | 39 | 19230.77 | +0.16% |
| | | 72 | 10416.67 | -0.003% |
| | | 78 | 9615.38 | +0.16% |
| | | 313 | 2396.17 | -0.16% |
| 8 MHz | 1/1 | 26 | 19230.77 | +0.16% |
| | | 48 | 10416.67 | -0.003% |
| | | 52 | 9615.38 | +0.16% |
| | | 208 | 2403.85 | +0.16% |

**Remark** The numbers in the table are for sixteen samples per bit (i.e. when NSPB[3:0] = 0000b or 1111b).

### 17.7.3 UART Mode

Figure 17-43 shows a block diagram of baud rate generation in UART mode.

**Figure 17-43. Block Diagram of Baud Rate Generation in UART Mode**




Note For the LIN communication clock source, refer to CHAPTER 5 CLOCK GENERATOR.

Set the LIN communications clock source as follows.

<R>
- LIN communications clock source = $f_{CLK}$Note1
- In the range from 4 MHz to 32 MHz

**Note 1.** It is available to select the $f_{MX}$ to the LIN communications clock source. In that case, set at least 1.2 times the frequency of the LIN communication clock source to the CPU/peripheral hardware clock($f_{CLK}$).

**Table 17-30. UART Baud Rate Setting Examples (when LIN communication clock source = 32 MHz)**

| UART Baud Rate (Target Baud Rate) | Prescaler | Baud rate generator 0-1 N+1 frequency-divided | Baud rate | Error |
|---|---|---|---|---|
| 1200 bps | 1/2 | 833 | 1200.48 | +0.04% |
| 2400 bps | 1/2 | 417 | 2398.08 | -0.08% |
| 4800 bps | 1/2 | 208 | 4807.69 | +0.16% |
| 9600 bps | 1/2 | 104 | 9615.38 | +0.16% |
| 19200 bps | 1/2 | 52 | 19230.77 | +0.16% |
| 31250 bps | 1/2 | 32 | 31250.00 | 0.00% |
| 38400 bps | 1/2 | 26 | 38461.54 | +0.16% |

**Remarks 1.** These examples are for sixteen samples per bit (i.e. when NSPB[3:0] = 0000b or 1111b).

**2.** The baud rate can be calculated by the following expression.

Baud rate = (LIN communications clock source ($f_{CLK}$ or $f_{MX}$: selected by the LINnMCK bits) frequency)
× (frequency divider selected by LPRS[2:0] ÷ (LBRPn0 + (0x0100 × LBRPn1) + 1))
÷ number selected by NSPB[3:0] bps

## 17.8 Noise Filter

The LIN/UART module has a noise filter for reducing erroneous receiving of data due to noise. By setting the LRDNFS bit in the LMDn register to 0 (to use the noise filter), the noise filter is activated. The noise filter samples the level of the synchronized LRXDn (n = 0, 1) based on the prescaler clock, and outputs the majority among three sampled levels. The value of each bit in the received data is determined by the noise filter output.

Figure 17-44 shows the configuration of the noise filter, figure 17-45 an example of a noise filter circuit, and figure 17-46 the determination of the received data when the noise filter is used.

**Figure 17-44. Configuration of Noise Filter**

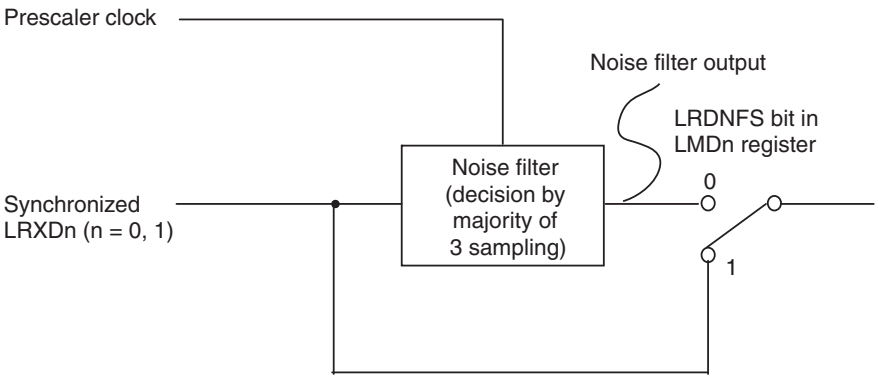



**Figure 17-45. Example of Noise Filter Circuit**

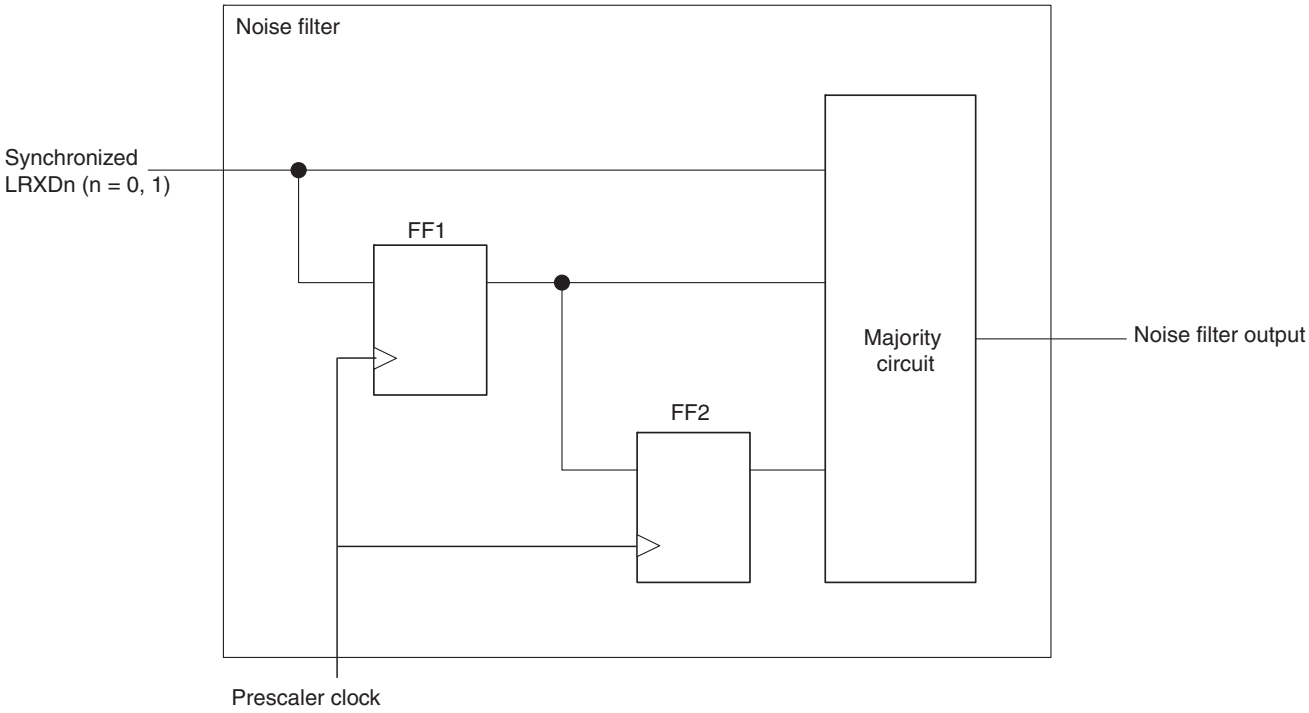
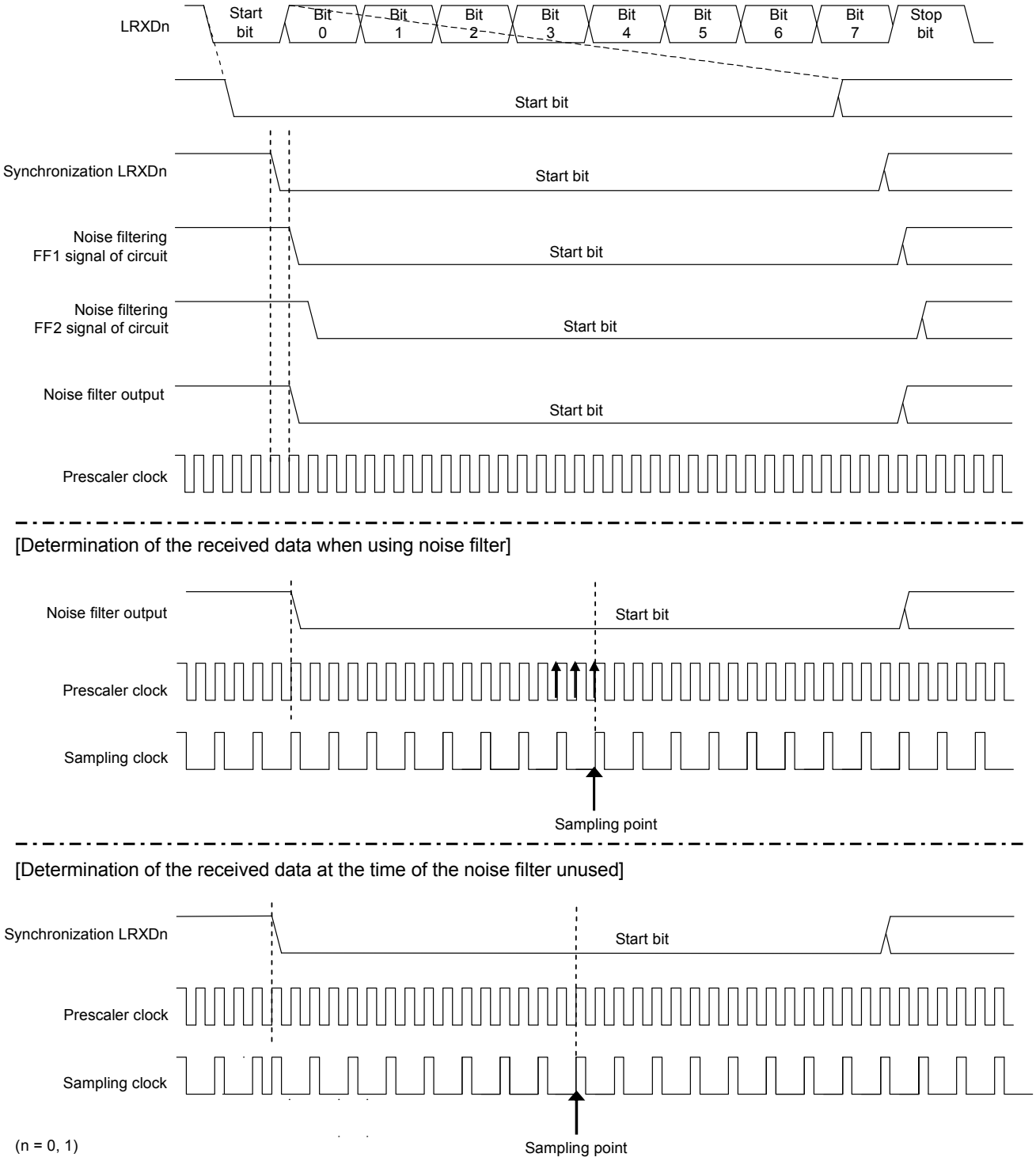


**Figure 17-46. Determination of Received Data when Noise Filter is Used**



## 17.9 Interrupts

The LIN/UART module generates four types of interrupt requests.

- LINn successful transmission interrupt
- LINn successful reception interrupt
- LINn reception status interrupt
- LINn interrupt

Setting the LIOS bit in the LMDn register to 0 allows to perform logical OR operation on all of the interrupt sources, outputting the interrupt request from the LINn interrupt.
Setting the LIOS bit in the LMDn register to 1 allows to output the LINn successful transmission interrupt, LINn successful reception interrupt, or LINn reception status interrupt depending on the interrupt sources.
Table 17-31 lists the sources for each interrupt.

**Table 17-31. Interrupt Sources**

| Mode | | LIOS bit in LMDn register is 0 | LIOS bit in LMDn register is 1 Note | | |
|---|---|---|---|---|---|
| | | LINn Interrupt | LINn Successful Transmission Interrupt | LINn Successful Reception Interrupt | LINn Reception Status Interrupt |
| LIN mode | LIN master mode | • Successful frame transmission<br>• Successful frame reception<br>• Successful wake-up transmission<br>• Successful wake-up reception<br>• Successful header transmission<br>• Bit error<br>• Physical bus error<br>• Frame/response timeout error<br>• Framing error<br>• Checksum error<br>• Response preparation error | • Successful frame transmission<br>• Successful wake-up transmission<br>• Successful header transmission | • Successful frame reception<br>• Successful wake-up reception | • Bit error<br>• Physical bus error<br>• Frame/response timeout error<br>• Framing error<br>• Checksum error<br>• Response preparation error |
| | LIN slave mode | • Successful response transmission<br>• Successful response reception<br>• Successful wake-up transmission<br>• Successful wake-up reception<br>• Successful header reception<br>• Bit error<br>• Frame/response timeout error<br>• Framing error<br>• Sync field error<br>• Checksum error<br>• ID parity error<br>• Response preparation error | • Successful response transmission<br>• Successful wake-up transmission | • Successful response reception<br>• Successful wake-up reception<br>• Successful header reception | • Bit error<br>• Frame/response timeout error<br>• Framing error<br>• Sync field error<br>• Checksum error<br>• ID parity error<br>• Response preparation error |
| UART mode | | — | • Transmission start/successful transmission | • Successful reception<br>• Expansion bit mismatch | • Bit error<br>• Overrun error<br>• Framing error<br>• Expansion bit detection<br>• ID match<br>• Parity error |

**Note** LIOS bit setting is enabled in LIN mode. LIOS bit setting is not required in UART mode.

Each interrupt request is output when the corresponding bit in the LIEn register is 1 (interrupt is enabled) and the corresponding flag in the LSTn register is 1.

In the RL78/F13 and RL78/F14, the LINn reception status interrupt and LINn interrupt are assigned to the same interrupt vector.

# CHAPTER 18 CAN INTERFACE (RS-CAN LITE)

The number of channels of the CAN module depends the product.

| | RL78/F13 (LIN incorporated) | RL78/F13 (CAN & LIN incorporated) | RL78/F14 |
|---|---|---|---|
| Channel | — | 1 | 1 |

## 18.1 Overview

The RL78/F13 and RL78/F14 incorporate one channel of the Controller Area Network (CAN) module conforming to the ISO11898-1 specifications. Table 18-1 shows the CAN module specifications. Figure 18-1 shows the CAN module block diagram.

In this chapter, the following variables indicate the number of channels and registers.

- i (i = 0): CAN channel number
- j (j = 0 to 15): CAN receive rule entry register number
  (GAFLIDLj, GAFLIDHj, GAFLMLj, GAFLMHj, GAFLPLj, GAFLPHj)
- k (k = 0) : Transmit/receive FIFO buffer number
- m (m = 0, 1): Receive FIFO buffer number
- n (n = 0 to 15): Receive buffer number
- p (p = 0 to 3): Transmit buffer number
- r (r = 0 to 127): CAN RAM test register (RPGACCr) number

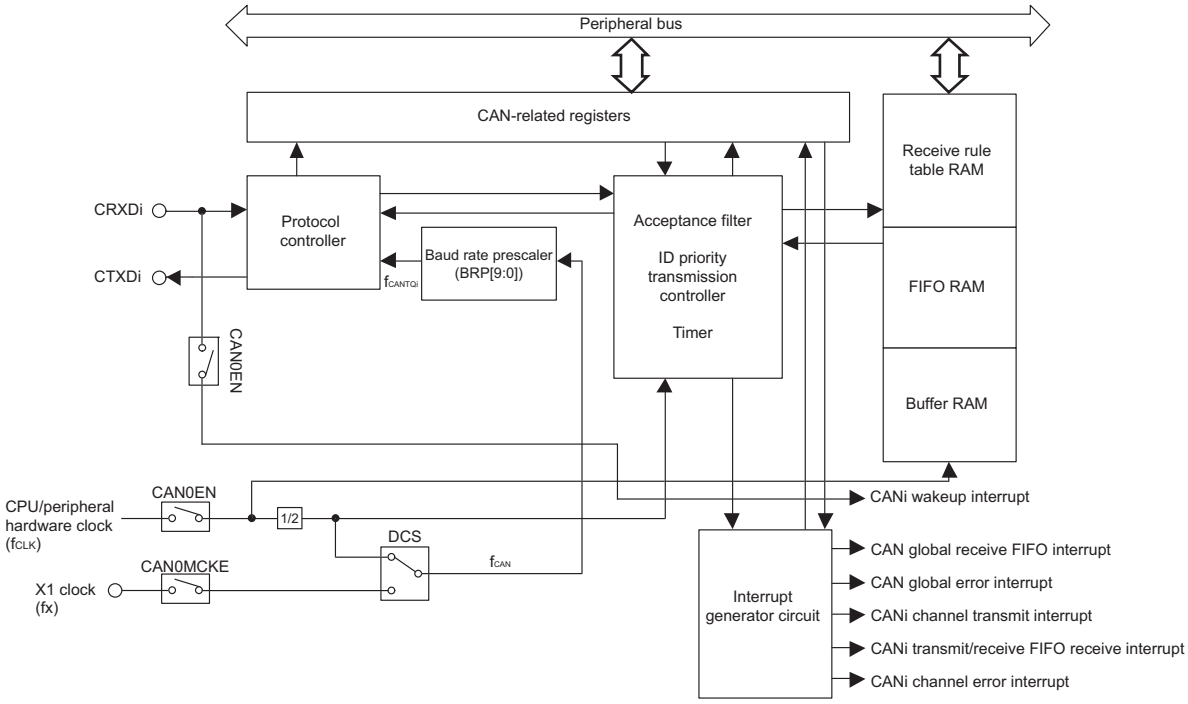
**Table 18-1. CAN Module Specifications (1/2)**

| Item | Specification |
|---|---|
| Number of channels | 1 |
| Protocol | ISO11898-1 compliant |
| Communication speed | • Maximum 1 Mbps<br>Communication speed (CANi bit time clock) = $\frac{1}{\text{CANi bit time}}$<br>CANi bit time = CANiTq x Tq count per bit<br>CANiTq = $\frac{\text{(BRP[9:0] bits in the CiCFGL register + 1)}}{f_{CAN}}$<br>i = 0<br>Tq: Time quantum<br>$f_{CAN}$: Frequency of CAN clock (selected by the DCS bit in the GCFGL register) |
| Buffer | 20 buffers in total<br>• Individual buffers: 4 buffers (4 buffers for one channel)<br>Transmit buffer: 4 buffers per a channel<br>• Shared buffers: 16 buffers<br>Receive buffer: 0 to 16 buffers<br>Receive FIFO buffer: 2 FIFO buffers (up to 16 buffers allocatable to each)<br>Transmit/receive FIFO buffer: A FIFO buffer per a channel (up to 16 buffers allocatable to each) |
| Reception function | • Receives data frames and remote frames.<br>• Selects ID format (standard ID, extended ID, or both IDs) to be received.<br>• Sets interrupt enable/disable for each FIFO.<br>• Mirror function (to receive messages transmitted from the own CAN node)<br>• Timestamp function (to record message reception time as a 16-bit timer value) |
| Reception filter function | • Selects receive messages according to 16 receive rules.<br>• Sets the number of receive rules (0 to 16) for each channel.<br>• Acceptance filter processing: Sets ID and mask for each receive rule.<br>• DLC filter processing: Sets DLC check value for each receive rule. |
| Receive message transfer function | • Routing function to transfer receive messages to arbitrary destinations (can be transferred to up to 2 buffers).<br>Transfer destination: Receive buffer, receive FIFO buffer, and transmit/receive FIFO buffer<br>• Label addition function<br>Stores label information together when storing a message in a receive buffer and FIFO buffer. |
| Transmission function | • Transmits data frames and remote frames.<br>• Selects ID format (standard ID, extended ID, or both IDs) to be transmitted.<br>• Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer.<br>• Selects ID priority transmission or transmit buffer number priority transmission.<br>• Transmit request can be aborted (possible to confirm with the flag)<br>• One-shot transmission function |
| Interval transmission function | Sets message transmission interval time (transmit mode of transmit/receive FIFO buffers) |
| Transmit history function | Stores the history information of transmitted messages. |

**Table 18-1. CAN Module Specifications (2/2)**

| Item | Specification |
|---|---|
| Bus off recovery mode selection | Selects a method of returning from bus off state.<br>• ISO11898-1 compliant<br>• Automatic entry to channel halt mode at bus-off entry<br>• Automatic entry to channel halt mode at bus-off end<br>• Entry to channel halt mode by a program<br>• Transition to the error-active state by a program (forcible return from the bus off state) |
| Error status monitoring | • Monitors CAN protocol errors (stuff error, form error, ACK error, CRC error, bit error, ACK delimiter error, and bus dominant lock).<br>• Detects error status transitions (error warning, error passive, bus off entry, and bus off recovery)<br>• Reads the error counter.<br>• Monitors DLC errors. |
| Interrupt source | 6 sources<br>• Global (2 sources)<br>CAN global receive FIFO interrupt<br>CAN global error interrupt<br>• Channel (4 sources/channel)<br>CANi channel transmit interrupt<br>CANi transmit complete interrupt<br>CANi transmit abort interrupt<br>CANi transmit/receive FIFO transmit complete interrupt<br>CANi transmit history interrupt<br>CANi transmit/receive FIFO receive interrupt<br>CANi channel error interrupt<br>CANi wakeup interrupt |
| CAN stop mode | Reduces power consumption by stopping clock supply to the CAN module. |
| CAN clock source | Selects the clock obtained by frequency-dividing $f_{CLK}$ by 2 ($f_{CLK}/2$) or the X1 clock (fx). |
| Test function | Test function for user evaluation<br>• Listen-only mode<br>• Self-test mode 0 (external loopback)<br>• Self-test mode 1 (internal loopback)<br>• RAM test (read/write test) |

**Figure 18-1. CAN Module Block Diagram (i = 0)**




Remark i = 0
BRP[9:0]: Bits in the CiCFGL register
DCS: Bit in the GCFGL register
$f_{CANTQi}$: CANi Tq clock
$f_{CAN}$: CAN clock
CAN0EN: Bit in the PER2 register
CAN0MCKE: Bit in the CANCKSEL register

## 18.2 Input/Output Pins

Table 18-2 lists the I/O pins of the CAN module.

**Table 18-2. I/O Pins of the CAN Module**

| Pin Name | I/O | Description |
|---|---|---|
| CRXDi | Input | Receive data input pins of the CAN communication function |
| CTXDi | Output | Transmit data output pins of the CAN communication function |

## 18.3 Register Descriptions

Table 18-3 lists the registers of the CAN module.

The values after reset of the registers allocated to the CAN RAM area (F03A0H to F0681H) are those after the CAN RAM is initialized.

**Table 18-3. List of CAN Module Registers (1/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F02C1H | Peripheral enable register 2 | PER2 | | R/W | √ | √ | — | 00H |
| F02C2H | CAN clock select register | CANCKSEL | | R/W | √ | √ | — | 00H |
| F0300H | CAN0 bit configuration register L | C0CFGLL | C0CFGL | R/W | — | √ | √ | 0000H |
| F0301H | | C0CFGLH | | | — | √ | | |
| F0302H | CAN0 bit configuration register H | C0CFGHL | C0CFGH | R/W | — | √ | √ | 0000H |
| F0303H | | C0CFGHH | | | — | √ | | |
| F0304H | CAN0 control register L | C0CTRLL | C0CTRL | R/W | — | √ | √ | 0005H Note |
| F0305H | | C0CTRLH | | | — | √ | | |
| F0306H | CAN0 control register H | C0CTRHL | C0CTRH | R/W | — | √ | √ | 0000H |
| F0307H | | C0CTRHH | | | — | √ | | |
| F0308H | CAN0 status register L | C0STSLL | C0STSL | R | — | √ | √ | 0005H Note |
| F0309H | | C0STSLH | | | — | √ | | |
| F030AH | CAN0 status register H | C0STSHL | C0STSH | R | — | √ | √ | 0000H |
| F030BH | | C0STSHH | | | — | √ | | |
| F030CH | CAN0 error flag register L | C0ERFLLL | C0ERFLL | R/W | — | √ | √ | 0000H |
| F030DH | | C0ERFLLH | | | — | √ | | |
| F030EH | CAN0 error flag register H | C0ERFLHL | C0ERFLH | R | — | √ | √ | 0000H |
| F030FH | | C0ERFLHH | | | — | √ | | |
| F0322H | CAN global configuration register L | GCFGLL | GCFGL | R/W | — | √ | √ | 0000H |
| F0323H | | GCFGLH | | | — | √ | | |
| F0324H | CAN global configuration register H | GCFGHL | GCFGH | R/W | — | √ | √ | 0000H |
| F0325H | | GCFGHH | | | — | √ | | |
| F0326H | CAN global control register L | GCTRLL | GCTRL | R/W | — | √ | √ | 0005H Note |
| F0327H | | GCTRLH | | | — | √ | | |
| F0328H | CAN global control register H | GCTRHL | GCTRH | R/W | — | √ | √ | 0000H |
| F0329H | | GCTRHH | | | — | √ | | |
| F032AH | CAN global status register | GSTSL | GSTS | R | — | √ | √ | 000DH Note |
| F032BH | | GSTSH | | | — | √ | | |
| F032CH | CAN global error flag register | GERFLL | | R/W | — | √ | — | 00H |
| F032EH | CAN timestamp register | GTSC | | R | — | — | √ | 0000H |
| F032FH | | | | | — | — | | |
| F0330H | CAN receive rule number configuration register | GAFLCFGL | GAFLCFG | R/W | — | √ | √ | 0000H |
| F0331H | | GAFLCFGH | | | — | √ | | |
| F0332H | CAN receive buffer number configuration register | RMNBL | RMNB | R/W | — | √ | √ | 0000H |
| F0333H | | — | | | — | — | | |
| F0334H | CAN receive buffer receive complete flag register | RMND0L | RMND0 | R/W | — | √ | √ | 0000H |
| F0335H | | RMND0H | | | — | √ | | |
| F0338H | CAN receive FIFO control register 0 | RFCC0L | RFCC0 | R/W | — | √ | √ | 0000H |
| F0339H | | RFCC0H | | | — | √ | | |
| F033AH | CAN receive FIFO control register 1 | RFCC1L | RFCC1 | R/W | — | √ | √ | 0000H |
| F033BH | | RFCC1H | | | — | √ | | |

**Note** When the CAN0EN bit in the PER2 register is set to 0, the read value is undefined.
When the CAN0EN bit in the PER2 register is set to 1, the read value is the initial value listed above.

**Table 18-3. List of CAN Module Registers (2/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F0340H | CAN receive FIFO status register 0 | RFSTS0L | RFSTS0 | R/W | — | √ | √ | 0001H Note |
| F0341H | | RFSTS0H | | R | — | √ | | |
| F0342H | CAN receive FIFO status register 1 | RFSTS1L | RFSTS1 | R/W | — | √ | √ | 0001H Note |
| F0343H | | RFSTS1H | | R | — | √ | | |
| F0348H | CAN receive FIFO pointer control register 0 | RFPCTR0L | RFPCTR0 | W | — | √ | √ | 0000H |
| F0349H | | RFPCTR0H | | | — | √ | | |
| F034AH | CAN receive FIFO pointer control register 1 | RFPCTR1L | RFPCTR1 | W | — | √ | √ | 0000H |
| F034BH | | RFPCTR1H | | | — | √ | | |
| F0350H | CAN0 transmit/receive FIFO control register 0L | CFCCL0L | CFCCL0 | R/W | — | √ | √ | 0000H |
| F0351H | | CFCCL0H | | | — | √ | | |
| F0352H | CAN0 transmit/receive FIFO control register 0H | CFCCH0L | CFCCH0 | R/W | — | √ | √ | 0000H |
| F0353H | | CFCCH0H | | | — | √ | | |
| F0358H | CAN0 transmit/receive FIFO status register 0 | CFSTS0L | CFSTS0 | R/W | — | √ | √ | 0001H Note |
| F0359H | | CFSTS0H | | R | — | √ | | |
| F035CH | CAN0 transmit/receive FIFO pointer control register 0 | CFPCTR0L | CFPCTR0 | W | — | √ | √ | 0000H |
| F035DH | | — | | | — | — | | |
| F0360H | Receive FIFO message lost status register | RFMSTS | | R | — | √ | — | 00H |
| F0361H | CAN0 transmit/receive FIFO message lost status register | CFMSTS | | R | — | √ | — | 00H |
| F0362H | CAN receive FIFO interrupt status register | RFISTS | | R | — | √ | — | 00H |
| F0363H | CAN transmit/receive FIFO receive interrupt status register | CFISTS | | R | — | √ | — | 00H |
| F0364H | CAN0 transmit buffer control register 0 | TMC0 | | R/W | — | √ | — | 00H |
| F0365H | CAN0 transmit buffer control register 1 | TMC1 | | R/W | — | √ | — | 00H |
| F0366H | CAN0 transmit buffer control register 2 | TMC2 | | R/W | — | √ | — | 00H |
| F0367H | CAN0 transmit buffer control register 3 | TMC3 | | R/W | — | √ | — | 00H |
| F036CH | CAN0 transmit buffer status register 0 | TMSTS0 | | R/W | — | √ | — | 00H |
| F036DH | CAN0 transmit buffer status register 1 | TMSTS1 | | R/W | — | √ | — | 00H |
| F036EH | CAN0 transmit buffer status register 2 | TMSTS2 | | R/W | — | √ | — | 00H |
| F036FH | CAN0 transmit buffer status register 3 | TMSTS3 | | R/W | — | √ | — | 00H |
| F0374H | CAN0 transmit buffer transmit request status register | TMTRSTSL | TMTRSTS | R | — | √ | √ | 0000H |
| F0375H | | TMTRSTSH | | | — | √ | | |
| F0376H | CAN0 transmit buffer transmit complete status register | TMTCSTSL | TMTCSTS | R | — | √ | √ | 0000H |
| F0377H | | TMTCSTSH | | | — | √ | | |
| F0378H | CAN0 transmit buffer transmit abort status register | TMTASTSL | TMTASTS | R | — | √ | √ | 0000H |
| F0379H | | TMTASTSH | | | — | √ | | |
| F037AH | CAN0 transmit buffer interrupt enable register | TMIECL | TMIEC | R/W | — | √ | √ | 0000H |
| F037BH | | TMIECH | | | — | √ | | |
| F037CH | CAN0 transmit history buffer control register | THLCC0L | THLCC0 | R/W | — | √ | √ | 0000H |
| F037DH | | THLCC0H | | | — | √ | | |
| F0380H | CAN0 transmit history buffer status register | THLSTS0L | THLSTS0 | R/W | — | √ | √ | 0001H Note |
| F0381H | | THLSTS0H | | | — | √ | | |
| F0384H | CAN0 transmit history buffer pointer control register | THLPCTR0L | THLPCTR0 | W | — | √ | √ | 0000H |
| F0385H | | THLPCTR0H | | | — | √ | | |
| F0388H | CAN global transmit interrupt status register | GTINTSTSL | GTINTSTS | R | — | √ | √ | 0000H |
| F0389H | | GTINTSTSH | | | — | √ | | |

**Note** When the CAN0EN bit in the PER2 register is set to 0, the read value is undefined.
When the CAN0EN bit in the PER2 register is set to 1, the read value is the initial value listed above.

**Table 18-3. List of CAN Module Registers (3/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F038AH | CAN global RAM window control register | GRWCRL | GRWCR | R/W | — | √ | √ | 0000H |
| F038BH | | GRWCRH | | | — | √ | | |
| F038CH | CAN global test configuration register | GTSTCFGL | GTSTCFG | R/W | — | √ | √ | 0000H |
| F038DH | | GTSTCFGH | | | — | √ | | |
| F038EH | CAN global test control register | GTSTCTRL | — | R/W | — | √ | — | 00H |
| F038FH | | — | | | — | — | | |
| F0394H | CAN global test protection unlock register | GLOCKK | | W | — | — | √ | 0000H |
| F0395H | | | | | — | — | | |
| F03A0H | CAN receive rule entry register 0AL Note 1 | GAFLIDL0L | GAFLIDL0 | R/W | — | √ | √ | 0000H |
| F03A1H | | GAFLIDL0H | | | — | √ | | |
| F03A0H | CAN receive buffer register 0AL Note 2 | RMIDL0L | RMIDL0 | R | — | √ | √ | 0000H |
| F03A1H | | RMIDL0H | | | — | √ | | |
| F03A2H | CAN receive rule entry register 0AH Note 1 | GAFLIDH0L | GAFLIDH0 | R/W | — | √ | √ | 0000H |
| F03A3H | | GAFLIDH0H | | | — | √ | | |
| F03A2H | CAN receive buffer register 0AH Note 2 | RMIDH0L | RMIDH0 | R | — | √ | √ | 0000H |
| F03A3H | | RMIDH0H | | | — | √ | | |
| F03A4H | CAN receive rule entry register 0BL Note 1 | GAFLML0L | GAFLML0 | R/W | — | √ | √ | 0000H |
| F03A5H | | GAFLML0H | | | — | √ | | |
| F03A4H | CAN receive buffer register 0BL Note 2 | RMTS0L | RMTS0 | R | — | √ | √ | 0000H |
| F03A5H | | RMTS0H | | | — | √ | | |
| F03A6H | CAN receive rule entry register 0BH Note 1 | GAFLMH0L | GAFLMH0 | R/W | — | √ | √ | 0000H |
| F03A7H | | GAFLMH0H | | | — | √ | | |
| F03A6H | CAN receive buffer register 0BH Note 2 | RMPTR0L | RMPTR0 | R | — | √ | √ | 0000H |
| F03A7H | | RMPTR0H | | | — | √ | | |
| F03A8H | CAN receive rule entry register 0CL Note 1 | GAFLPL0L | GAFLPL0 | R/W | — | √ | √ | 0000H |
| F03A9H | | GAFLPL0H | | | — | √ | | |
| F03A8H | CAN receive buffer register 0CL Note 2 | RMDF00L | RMDF00 | R | — | √ | √ | 0000H |
| F03A9H | | RMDF00H | | | — | √ | | |
| F03AAH | CAN receive rule entry register 0CH Note 1 | GAFLPH0L | GAFLPH0 | R/W | — | √ | √ | 0000H |
| F03ABH | | GAFLPH0H | | | — | √ | | |
| F03AAH | CAN receive buffer register 0CH Note 2 | RMDF10L | RMDF10 | R | — | √ | √ | 0000H |
| F03ABH | | RMDF10H | | | — | √ | | |
| F03ACH | CAN receive rule entry register 1AL Note 1 | GAFLIDL1L | GAFLIDL1 | R/W | — | √ | √ | 0000H |
| F03ADH | | GAFLIDL1H | | | — | √ | | |
| F03ACH | CAN receive buffer register 0DL Note 2 | RMDF20L | RMDF20 | R | — | √ | √ | 0000H |
| F03ADH | | RMDF20H | | | — | √ | | |
| F03AEH | CAN receive rule entry register 1AH Note 1 | GAFLIDH1L | GAFLIDH1 | R/W | — | √ | √ | 0000H |
| F03AFH | | GAFLIDH1H | | | — | √ | | |
| F03AEH | CAN receive buffer register 0DH Note 2 | RMDF30L | RMDF30 | R | — | √ | √ | 0000H |
| F03AFH | | RMDF30H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (4/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F03B0H | CAN receive rule entry register 1BL Note 1 | GAFLML1L | GAFLML1 | R/W | — | √ | √ | 0000H |
| F03B1H | | GAFLML1H | | | — | √ | | |
| F03B0H | CAN receive buffer register 1AL Note 2 | RMIDL1L | RMIDL1 | R | — | √ | √ | 0000H |
| F03B1H | | RMIDL1H | | | — | √ | | |
| F03B2H | CAN receive rule entry register 1BH Note 1 | GAFLMH1L | GAFLMH1 | R/W | — | √ | √ | 0000H |
| F03B3H | | GAFLMH1H | | | — | √ | | |
| F03B2H | CAN receive buffer register 1AH Note 2 | RMIDH1L | RMIDH1 | R | — | √ | √ | 0000H |
| F03B3H | | RMIDH1H | | | — | √ | | |
| F03B4H | CAN receive rule entry register 1CL Note 1 | GAFLPL1L | GAFLPL1 | R/W | — | √ | √ | 0000H |
| F03B5H | | GAFLPL1H | | | — | √ | | |
| F03B4H | CAN receive buffer register 1BL Note 2 | RMTS1L | RMTS1 | R | — | √ | √ | 0000H |
| F03B5H | | RMTS1H | | | — | √ | | |
| F03B6H | CAN receive rule entry register 1CH Note 1 | GAFLPH1L | GAFLPH1 | R/W | — | √ | √ | 0000H |
| F03B7H | | GAFLPH1H | | | — | √ | | |
| F03B6H | CAN receive buffer register 1BH Note 2 | RMPTR1L | RMPTR1 | R | — | √ | √ | 0000H |
| F03B7H | | RMPTR1H | | | — | √ | | |
| F03B8H | CAN receive rule entry register 2AL Note 1 | GAFLIDL2L | GAFLIDL2 | R/W | — | √ | √ | 0000H |
| F03B9H | | GAFLIDL2H | | | — | √ | | |
| F03B8H | CAN receive buffer register 1CL Note 2 | RMDF01L | RMDF01 | R | — | √ | √ | 0000H |
| F03B9H | | RMDF01H | | | — | √ | | |
| F03BAH | CAN receive rule entry register 2AH Note 1 | GAFLIDH2L | GAFLIDH2 | R/W | — | √ | √ | 0000H |
| F03BBH | | GAFLIDH2H | | | — | √ | | |
| F03BAH | CAN receive buffer register 1CH Note 2 | RMDF11L | RMDF11 | R | — | √ | √ | 0000H |
| F03BBH | | RMDF11H | | | — | √ | | |
| F03BCH | CAN receive rule entry register 2BL Note 1 | GAFLML2L | GAFLML2 | R/W | — | √ | √ | 0000H |
| F03BDH | | GAFLML2H | | | — | √ | | |
| F03BCH | CAN receive buffer register 1DL Note 2 | RMDF21L | RMDF21 | R | — | √ | √ | 0000H |
| F03BDH | | RMDF21H | | | — | √ | | |
| F03BEH | CAN receive rule entry register 2BH Note 1 | GAFLMH2L | GAFLMH2 | R/W | — | √ | √ | 0000H |
| F03BFH | | GAFLMH2H | | | — | √ | | |
| F03BEH | CAN receive buffer register 1DH Note 2 | RMDF31L | RMDF31 | R | — | √ | √ | 0000H |
| F03BFH | | RMDF31H | | | — | √ | | |
| F03C0H | CAN receive rule entry register 2CL Note 1 | GAFLPL2L | GAFLPL2 | R/W | — | √ | √ | 0000H |
| F03C1H | | GAFLPL2H | | | — | √ | | |
| F03C0H | CAN receive buffer register 2AL Note 2 | RMIDL2L | RMIDL2 | R | — | √ | √ | 0000H |
| F03C1H | | RMIDL2H | | | — | √ | | |
| F03C2H | CAN receive rule entry register 2CH Note 1 | GAFLPH2L | GAFLPH2 | R/W | — | √ | √ | 0000H |
| F03C3H | | GAFLPH2H | | | — | √ | | |
| F03C2H | CAN receive buffer register 2AH Note 2 | RMIDH2L | RMIDH2 | R | — | √ | √ | 0000H |
| F03C3H | | RMIDH2H | | | — | √ | | |
| F03C4H | CAN receive rule entry register 3AL Note 1 | GAFLIDL3L | GAFLIDL3 | R/W | — | √ | √ | 0000H |
| F03C5H | | GAFLIDL3H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (5/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F03C4H | CAN receive buffer register 2BL[Note 2] | RMTS2L | RMTS2 | R | — | √ | √ | 0000H |
| F03C5H | | RMTS2H | | | — | √ | | |
| F03C6H | CAN receive rule entry register 3AH[Note 1] | GAFLIDH3L | GAFLIDH3 | R/W | — | √ | √ | 0000H |
| F03C7H | | GAFLIDH3H | | | — | √ | | |
| F03C6H | CAN receive buffer register 2BH[Note 2] | RMPTR2L | RMPTR2 | R | — | √ | √ | 0000H |
| F03C7H | | RMPTR2H | | | — | √ | | |
| F03C8H | CAN receive rule entry register 3BL[Note 1] | GAFLML3L | GAFLML3 | R/W | — | √ | √ | 0000H |
| F03C9H | | GAFLML3H | | | — | √ | | |
| F03C8H | CAN receive buffer register 2CL[Note 2] | RMDF02L | RMDF02 | R | — | √ | √ | 0000H |
| F03C9H | | RMDF02H | | | — | √ | | |
| F03CAH | CAN receive rule entry register 3BH[Note 1] | GAFLMH3L | GAFLMH3 | R/W | — | √ | √ | 0000H |
| F03CBH | | GAFLMH3H | | | — | √ | | |
| F03CAH | CAN receive buffer register 2CH[Note 2] | RMDF12L | RMDF12 | R | — | √ | √ | 0000H |
| F03CBH | | RMDF12H | | | — | √ | | |
| F03CCH | CAN receive rule entry register 3CL[Note 1] | GAFLPL3L | GAFLPL3 | R/W | — | √ | √ | 0000H |
| F03CDH | | GAFLPL3H | | | — | √ | | |
| F03CCH | CAN receive buffer register 2DL[Note 2] | RMDF22L | RMDF22 | R | — | √ | √ | 0000H |
| F03CDH | | RMDF22H | | | — | √ | | |
| F03CEH | CAN receive rule entry register 3CH[Note 1] | GAFLPH3L | GAFLPH3 | R/W | — | √ | √ | 0000H |
| F03CFH | | GAFLPH3H | | | — | √ | | |
| F03CEH | CAN receive buffer register 2DH[Note 2] | RMDF32L | RMDF32 | R | — | √ | √ | 0000H |
| F03CFH | | RMDF32H | | | — | √ | | |
| F03D0H | CAN receive rule entry register 4AL[Note 1] | GAFLIDL4L | GAFLIDL4 | R/W | — | √ | √ | 0000H |
| F03D1H | | GAFLIDL4H | | | — | √ | | |
| F03D0H | CAN receive buffer register 3AL[Note 2] | RMIDL3L | RMIDL3 | R | — | √ | √ | 0000H |
| F03D1H | | RMIDL3H | | | — | √ | | |
| F03D2H | CAN receive rule entry register 4AH[Note 1] | GAFLIDH4L | GAFLIDH4 | R/W | — | √ | √ | 0000H |
| F03D3H | | GAFLIDH4H | | | — | √ | | |
| F03D2H | CAN receive buffer register 3AH[Note 2] | RMIDH3L | RMIDH3 | R | — | √ | √ | 0000H |
| F03D3H | | RMIDH3H | | | — | √ | | |
| F03D4H | CAN receive rule entry register 4BL[Note 1] | GAFLML4L | GAFLML4 | R/W | — | √ | √ | 0000H |
| F03D5H | | GAFLML4H | | | — | √ | | |
| F03D4H | CAN receive buffer register 3BL[Note 2] | RMTS3L | RMTS3 | R | — | √ | √ | 0000H |
| F03D5H | | RMTS3H | | | — | √ | | |
| F03D6H | CAN receive rule entry register 4BH[Note 1] | GAFLMH4L | GAFLMH4 | R/W | — | √ | √ | 0000H |
| F03D7H | | GAFLMH4H | | | — | √ | | |
| F03D6H | CAN receive buffer register 3BH[Note 2] | RMPTR3L | RMPTR3 | R | — | √ | √ | 0000H |
| F03D7H | | RMPTR3H | | | — | √ | | |
| F03D8H | CAN receive rule entry register 4CL[Note 1] | GAFLPL4L | GAFLPL4 | R/W | — | √ | √ | 0000H |
| F03D9H | | GAFLPL4H | | | — | √ | | |
| F03D8H | CAN receive buffer register 3CL[Note 2] | RMDF03L | RMDF03 | R | — | √ | √ | 0000H |
| F03D9H | | RMDF03H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (6/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F03DAH | CAN receive rule entry register 4CH[Note 1] | GAFLPH4L | GAFLPH4 | R/W | — | √ | √ | 0000H |
| F03DBH | | GAFLPH4H | | | — | √ | | |
| F03DAH | CAN receive buffer register 3CH[Note 2] | RMDF13L | RMDF13 | R | — | √ | √ | 0000H |
| F03DBH | | RMDF13H | | | — | √ | | |
| F03DCH | CAN receive rule entry register 5AL[Note 1] | GAFLIDL5L | GAFLIDL5 | R/W | — | √ | √ | 0000H |
| F03DDH | | GAFLIDL5H | | | — | √ | | |
| F03DCH | CAN receive buffer register 3DL[Note 2] | RMDF23L | RMDF23 | R | — | √ | √ | 0000H |
| F03DDH | | RMDF23H | | | — | √ | | |
| F03DEH | CAN receive rule entry register 5AH[Note 1] | GAFLIDH5L | GAFLIDH5 | R/W | — | √ | √ | 0000H |
| F03DFH | | GAFLIDH5H | | | — | √ | | |
| F03DEH | CAN receive buffer register 3DH[Note 2] | RMDF33L | RMDF33 | R | — | √ | √ | 0000H |
| F03DFH | | RMDF33H | | | — | √ | | |
| F03E0H | CAN receive rule entry register 5BL[Note 1] | GAFLML5L | GAFLML5 | R/W | — | √ | √ | 0000H |
| F03E1H | | GAFLML5H | | | — | √ | | |
| F03E0H | CAN receive buffer register 4AL[Note 2] | RMIDL4L | RMIDL4 | R | — | √ | √ | 0000H |
| F03E1H | | RMIDL4H | | | — | √ | | |
| F03E2H | CAN receive rule entry register 5BH[Note 1] | GAFLMH5L | GAFLMH5 | R/W | — | √ | √ | 0000H |
| F03E3H | | GAFLMH5H | | | — | √ | | |
| F03E2H | CAN receive buffer register 4AH[Note 2] | RMIDH4L | RMIDH4 | R | — | √ | √ | 0000H |
| F03E3H | | RMIDH4H | | | — | √ | | |
| F03E4H | CAN receive rule entry register 5CL[Note 1] | GAFLPL5L | GAFLPL5 | R/W | — | √ | √ | 0000H |
| F03E5H | | GAFLPL5H | | | — | √ | | |
| F03E4H | CAN receive buffer register 4BL[Note 2] | RMTS4L | RMTS4 | R | — | √ | √ | 0000H |
| F03E5H | | RMTS4H | | | — | √ | | |
| F03E6H | CAN receive rule entry register 5CH[Note 1] | GAFLPH5L | GAFLPH5 | R/W | — | √ | √ | 0000H |
| F03E7H | | GAFLPH5H | | | — | √ | | |
| F03E6H | CAN receive buffer register 4BH[Note 2] | RMPTR4L | RMPTR4 | R | — | √ | √ | 0000H |
| F03E7H | | RMPTR4H | | | — | √ | | |
| F03E8H | CAN receive rule entry register 6AL[Note 1] | GAFLIDL6L | GAFLIDL6 | R/W | — | √ | √ | 0000H |
| F03E9H | | GAFLIDL6H | | | — | √ | | |
| F03E8H | CAN receive buffer register 4CL[Note 2] | RMDF04L | RMDF04 | R | — | √ | √ | 0000H |
| F03E9H | | RMDF04H | | | — | √ | | |
| F03EAH | CAN receive rule entry register 6AH[Note 1] | GAFLIDH6L | GAFLIDH6 | R/W | — | √ | √ | 0000H |
| F03EBH | | GAFLIDH6H | | | — | √ | | |
| F03EAH | CAN receive buffer register 4CH[Note 2] | RMDF14L | RMDF14 | R | — | √ | √ | 0000H |
| F03EBH | | RMDF14H | | | — | √ | | |
| F03ECH | CAN receive rule entry register 6BL[Note 1] | GAFLML6L | GAFLML6 | R/W | — | √ | √ | 0000H |
| F03EDH | | GAFLML6H | | | — | √ | | |
| F03ECH | CAN receive buffer register 4DL[Note 2] | RMDF24L | RMDF24 | R | — | √ | √ | 0000H |
| F03EDH | | RMDF24H | | | — | √ | | |
| F03EEH | CAN receive rule entry register 6BH[Note 1] | GAFLMH6L | GAFLMH6 | R/W | — | √ | √ | 0000H |
| F03EFH | | GAFLMH6H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (7/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F03EEH | CAN receive buffer register 4DH[Note 2] | RMDF34L | RMDF34 | R/W | — | √ | √ | 0000H |
| F03EFH | | RMDF34H | | | — | √ | | |
| F03F0H | CAN receive rule entry register 6CL[Note 1] | GAFLPL6L | GAFLPL6 | R/W | — | √ | √ | 0000H |
| F03F1H | | GAFLPL6H | | | — | √ | | |
| F03F0H | CAN receive buffer register 5AL[Note 2] | RMIDL5L | RMIDL5 | R | — | √ | √ | 0000H |
| F03F1H | | RMIDL5H | | | — | √ | | |
| F03F2H | CAN receive rule entry register 6CH[Note 1] | GAFLPH6L | GAFLPH6 | R/W | — | √ | √ | 0000H |
| F03F3H | | GAFLPH6H | | | — | √ | | |
| F03F2H | CAN receive buffer register 5AH[Note 2] | RMIDH5L | RMIDH5 | R | — | √ | √ | 0000H |
| F03F3H | | RMIDH5H | | | — | √ | | |
| F03F4H | CAN receive rule entry register 7AL[Note 1] | GAFLIDL7L | GAFLIDL7 | R/W | — | √ | √ | 0000H |
| F03F5H | | GAFLIDL7H | | | — | √ | | |
| F03F4H | CAN receive buffer register 5BL[Note 2] | RMTS5L | RMTS5 | R | — | √ | √ | 0000H |
| F03F5H | | RMTS5H | | | — | √ | | |
| F03F6H | CAN receive rule entry register 7AH[Note 1] | GAFLIDH7L | GAFLIDH7 | R/W | — | √ | √ | 0000H |
| F03F7H | | GAFLIDH7H | | | — | √ | | |
| F03F6H | CAN receive buffer register 5BH[Note 2] | RMPTR5L | RMPTR5 | R | — | √ | √ | 0000H |
| F03F7H | | RMPTR5H | | | — | √ | | |
| F03F8H | CAN receive rule entry register 7BL[Note 1] | GAFLML7L | GAFLML7 | R/W | — | √ | √ | 0000H |
| F03F9H | | GAFLML7H | | | — | √ | | |
| F03F8H | CAN receive buffer register 5CL[Note 2] | RMDF05L | RMDF05 | R | — | √ | √ | 0000H |
| F03F9H | | RMDF05H | | | — | √ | | |
| F03FAH | CAN receive rule entry register 7BH[Note 1] | GAFLMH7L | GAFLMH7 | R/W | — | √ | √ | 0000H |
| F03FBH | | GAFLMH7H | | | — | √ | | |
| F03FAH | CAN receive buffer register 5CH[Note 2] | RMDF15L | RMDF15 | R | — | √ | √ | 0000H |
| F03FBH | | RMDF15H | | | — | √ | | |
| F03FCH | CAN receive rule entry register 7CL[Note 1] | GAFLPL7L | GAFLPL7 | R/W | — | √ | √ | 0000H |
| F03FDH | | GAFLPL7H | | | — | √ | | |
| F03FCH | CAN receive buffer register 5DL[Note 2] | RMDF25L | RMDF25 | R | — | √ | √ | 0000H |
| F03FDH | | RMDF25H | | | — | √ | | |
| F03FEH | CAN receive rule entry register 7CH[Note 1] | GAFLPH7L | GAFLPH7 | R/W | — | √ | √ | 0000H |
| F03FFH | | GAFLPH7H | | | — | √ | | |
| F03FEH | CAN receive buffer register 5DH[Note 2] | RMDF35L | RMDF35 | R | — | √ | √ | 0000H |
| F03FFH | | RMDF35H | | | — | √ | | |
| F0400H | CAN receive rule entry register 8AL[Note 1] | GAFLIDL8L | GAFLIDL8 | R/W | — | √ | √ | 0000H |
| F0401H | | GAFLIDL8H | | | — | √ | | |
| F0400H | CAN receive buffer register 6AL[Note 2] | RMIDL6L | RMIDL6 | R | — | √ | √ | 0000H |
| F0401H | | RMIDL6H | | | — | √ | | |
| F0402H | CAN receive rule entry register 8AH[Note 1] | GAFLIDH8L | GAFLIDH8 | R/W | — | √ | √ | 0000H |
| F0403H | | GAFLIDH8H | | | — | √ | | |
| F0402H | CAN receive buffer register 6AH[Note 2] | RMIDH6L | RMIDH6 | R | — | √ | √ | 0000H |
| F0403H | | RMIDH6H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (8/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F0404H | CAN receive rule entry register 8BL[Note 1] | GAFLML8L | GAFLML8 | R/W | — | √ | √ | 0000H |
| F0405H | | GAFLML8H | | | — | √ | | |
| F0404H | CAN receive buffer register 6BL[Note 2] | RMTS6L | RMTS6 | R | — | √ | √ | 0000H |
| F0405H | | RMTS6H | | | — | √ | | |
| F0406H | CAN receive rule entry register 8BH[Note 1] | GAFLMH8L | GAFLMH8 | R/W | — | √ | √ | 0000H |
| F0407H | | GAFLMH8H | | | — | √ | | |
| F0406H | CAN receive buffer register 6BH[Note 2] | RMPTR6L | RMPTR6 | R | — | √ | √ | 0000H |
| F0407H | | RMPTR6H | | | — | √ | | |
| F0408H | CAN receive rule entry register 8CL[Note 1] | GAFLPL8L | GAFLPL8 | R/W | — | √ | √ | 0000H |
| F0409H | | GAFLPL8H | | | — | √ | | |
| F0408H | CAN receive buffer register 6CL[Note 2] | RMDF06L | RMDF06 | R | — | √ | √ | 0000H |
| F0409H | | RMDF06H | | | — | √ | | |
| F040AH | CAN receive rule entry register 8CH[Note 1] | GAFLPH8L | GAFLPH8 | R/W | — | √ | √ | 0000H |
| F040BH | | GAFLPH8H | | | — | √ | | |
| F040AH | CAN receive buffer register 6CH[Note 2] | RMDF16L | RMDF16 | R | — | √ | √ | 0000H |
| F040BH | | RMDF16H | | | — | √ | | |
| F040CH | CAN receive rule entry register 9AL[Note 1] | GAFLIDL9L | GAFLIDL9 | R/W | — | √ | √ | 0000H |
| F040DH | | GAFLIDL9H | | | — | √ | | |
| F040CH | CAN receive buffer register 6DL[Note 2] | RMDF26L | RMDF26 | R | — | √ | √ | 0000H |
| F040DH | | RMDF26H | | | — | √ | | |
| F040EH | CAN receive rule entry register 9AH[Note 1] | GAFLIDH9L | GAFLIDH9 | R/W | — | √ | √ | 0000H |
| F040FH | | GAFLIDH9H | | | — | √ | | |
| F040EH | CAN receive buffer register 6DH[Note 2] | RMDF36L | RMDF36 | R | — | √ | √ | 0000H |
| F040FH | | RMDF36H | | | — | √ | | |
| F0410H | CAN receive rule entry register 9BL[Note 1] | GAFLML9L | GAFLML9 | R/W | — | √ | √ | 0000H |
| F0411H | | GAFLML9H | | | — | √ | | |
| F0410H | CAN receive buffer register 7AL[Note 2] | RMIDL7L | RMIDL7 | R | — | √ | √ | 0000H |
| F0411H | | RMIDL7H | | | — | √ | | |
| F0412H | CAN receive rule entry register 9BH[Note 1] | GAFLMH9L | GAFLMH9 | R/W | — | √ | √ | 0000H |
| F0413H | | GAFLMH9H | | | — | √ | | |
| F0412H | CAN receive buffer register 7AH[Note 2] | RMIDH7L | RMIDH7 | R | — | √ | √ | 0000H |
| F0413H | | RMIDH7H | | | — | √ | | |
| F0414H | CAN receive rule entry register 9CL[Note 1] | GAFLPL9L | GAFLPL9 | R/W | — | √ | √ | 0000H |
| F0415H | | GAFLPL9H | | | — | √ | | |
| F0414H | CAN receive buffer register 7BL[Note 2] | RMTS7L | RMTS7 | R | — | √ | √ | 0000H |
| F0415H | | RMTS7H | | | — | √ | | |
| F0416H | CAN receive rule entry register 9CH[Note 1] | GAFLPH9L | GAFLPH9 | R/W | — | √ | √ | 0000H |
| F0417H | | GAFLPH9H | | | — | √ | | |
| F0416H | CAN receive buffer register 7BH[Note 2] | RMPTR7L | RMPTR7 | R | — | √ | √ | 0000H |
| F0417H | | RMPTR7H | | | — | √ | | |
| F0418H | CAN receive rule entry register 10AL[Note 1] | GAFLIDL10L | GAFLIDL10 | R/W | — | √ | √ | 0000H |
| F0419H | | GAFLIDL10H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (9/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F0418H | CAN receive buffer register 7CL[Note 2] | RMDF07L | RMDF07 | R | — | √ | √ | 0000H |
| F0419H | | RMDF07H | | | — | √ | | |
| F041AH | CAN receive rule entry register 10AH[Note 1] | GAFLIDH10L | GAFLIDH10 | R/W | — | √ | √ | 0000H |
| F041BH | | GAFLIDH10H | | | — | √ | | |
| F041AH | CAN receive buffer register 7CH[Note 2] | RMDF17L | RMDF17 | R | — | √ | √ | 0000H |
| F041BH | | RMDF17H | | | — | √ | | |
| F041CH | CAN receive rule entry register 10BL[Note 1] | GAFLML10L | GAFLML10 | R/W | — | √ | √ | 0000H |
| F041DH | | GAFLML10H | | | — | √ | | |
| F041CH | CAN receive buffer register 7DL[Note 2] | RMDF27L | RMDF27 | R | — | √ | √ | 0000H |
| F041DH | | RMDF27H | | | — | √ | | |
| F041EH | CAN receive rule entry register 10BH[Note 1] | GAFLMH10L | GAFLMH10 | R/W | — | √ | √ | 0000H |
| F041FH | | GAFLMH10H | | | — | √ | | |
| F041EH | CAN receive buffer register 7DH[Note 2] | RMDF37L | RMDF37 | R | — | √ | √ | 0000H |
| F041FH | | RMDF37H | | | — | √ | | |
| F0420H | CAN receive rule entry register 10CL[Note 1] | GAFLPL10L | GAFLPL10 | R/W | — | √ | √ | 0000H |
| F0421H | | GAFLPL10H | | | — | √ | | |
| F0420H | CAN receive buffer register 8AL[Note 2] | RMIDL8L | RMIDL8 | R | — | √ | √ | 0000H |
| F0421H | | RMIDL8H | | | — | √ | | |
| F0422H | CAN receive rule entry register 10CH[Note 1] | GAFLPH10L | GAFLPH10 | R/W | — | √ | √ | 0000H |
| F0423H | | GAFLPH10H | | | — | √ | | |
| F0422H | CAN receive buffer register 8AH[Note 2] | RMIDH8L | RMIDH8 | R | — | √ | √ | 0000H |
| F0423H | | RMIDH8H | | | — | √ | | |
| F0424H | CAN receive rule entry register 11AL[Note 1] | GAFLIDL11L | GAFLIDL11 | R/W | — | √ | √ | 0000H |
| F0425H | | GAFLIDL11H | | | — | √ | | |
| F0424H | CAN receive buffer register 8BL[Note 2] | RMTS8L | RMTS8 | R | — | √ | √ | 0000H |
| F0425H | | RMTS8H | | | — | √ | | |
| F0426H | CAN receive rule entry register 11AH[Note 1] | GAFLIDH11L | GAFLIDH11 | R/W | — | √ | √ | 0000H |
| F0427H | | GAFLIDH11H | | | — | √ | | |
| F0426H | CAN receive buffer register 8BH[Note 2] | RMPTR8L | RMPTR8 | R | — | √ | √ | 0000H |
| F0427H | | RMPTR8H | | | — | √ | | |
| F0428H | CAN receive rule entry register 11BL[Note 1] | GAFLML11L | GAFLML11 | R/W | — | √ | √ | 0000H |
| F0429H | | GAFLML11H | | | — | √ | | |
| F0428H | CAN receive buffer register 8CL[Note 2] | RMDF08L | RMDF08 | R | — | √ | √ | 0000H |
| F0429H | | RMDF08H | | | — | √ | | |
| F042AH | CAN receive rule entry register 11BH[Note 1] | GAFLMH11L | GAFLMH11 | R/W | — | √ | √ | 0000H |
| F042BH | | GAFLMH11H | | | — | √ | | |
| F042AH | CAN receive buffer register 8CH[Note 2] | RMDF18L | RMDF18 | R | — | √ | √ | 0000H |
| F042BH | | RMDF18H | | | — | √ | | |
| F042CH | CAN receive rule entry register 11CL[Note 1] | GAFLPL11L | GAFLPL11 | R/W | — | √ | √ | 0000H |
| F042DH | | GAFLPL11H | | | — | √ | | |
| F042CH | CAN receive buffer register 8DL[Note 2] | RMDF28L | RMDF28 | R | — | √ | √ | 0000H |
| F042DH | | RMDF28H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (10/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F042EH | CAN receive rule entry register 11CH[Note 1] | GAFLPH11L | GAFLPH11 | R/W | — | √ | √ | 0000H |
| F042FH | | GAFLPH11H | | | — | √ | | |
| F042EH | CAN receive buffer register 8DH[Note 2] | RMDF38L | RMDF38 | R | — | √ | √ | 0000H |
| F042FH | | RMDF38H | | | — | √ | | |
| F0430H | CAN receive rule entry register 12AL[Note 1] | GAFLIDL12L | GAFLIDL12 | R/W | — | √ | √ | 0000H |
| F0431H | | GAFLIDL12H | | | — | √ | | |
| F0430H | CAN receive buffer register 9AL[Note 2] | RMIDL9L | RMIDL9 | R | — | √ | √ | 0000H |
| F0431H | | RMIDL9H | | | — | √ | | |
| F0432H | CAN receive rule entry register 12AH[Note 1] | GAFLIDH12L | GAFLIDH12 | R/W | — | √ | √ | 0000H |
| F0433H | | GAFLIDH12H | | | — | √ | | |
| F0432H | CAN receive buffer register 9AH[Note 2] | RMIDH9L | RMIDH9 | R | — | √ | √ | 0000H |
| F0433H | | RMIDH9H | | | — | √ | | |
| F0434H | CAN receive rule entry register 12BL[Note 1] | GAFLML12L | GAFLML12 | R/W | — | √ | √ | 0000H |
| F0435H | | GAFLML12H | | | — | √ | | |
| F0434H | CAN receive buffer register 9BL[Note 2] | RMTS9L | RMTS9 | R | — | √ | √ | 0000H |
| F0435H | | RMTS9H | | | — | √ | | |
| F0436H | CAN receive rule entry register 12BH[Note 1] | GAFLMH12L | GAFLMH12 | R/W | — | √ | √ | 0000H |
| F0437H | | GAFLMH12H | | | — | √ | | |
| F0436H | CAN receive buffer register 9BH[Note 2] | RMPTR9L | RMPTR9 | R | — | √ | √ | 0000H |
| F0437H | | RMPTR9H | | | — | √ | | |
| F0438H | CAN receive rule entry register 12CL[Note 1] | GAFLPL12L | GAFLPL12 | R/W | — | √ | √ | 0000H |
| F0439H | | GAFLPL12H | | | — | √ | | |
| F0438H | CAN receive buffer register 9CL[Note 2] | RMDF09L | RMDF09 | R | — | √ | √ | 0000H |
| F0439H | | RMDF09H | | | — | √ | | |
| F043AH | CAN receive rule entry register 12CH[Note 1] | GAFLPH12L | GAFLPH12 | R/W | — | √ | √ | 0000H |
| F043BH | | GAFLPH12H | | | — | √ | | |
| F043AH | CAN receive buffer register 9CH[Note 2] | RMDF19L | RMDF19 | R | — | √ | √ | 0000H |
| F043BH | | RMDF19H | | | — | √ | | |
| F043CH | CAN receive rule entry register 13AL[Note 1] | GAFLIDL13L | GAFLIDL13 | R/W | — | √ | √ | 0000H |
| F043DH | | GAFLIDL13H | | | — | √ | | |
| F043CH | CAN receive buffer register 9DL[Note 2] | RMDF29L | RMDF29 | R | — | √ | √ | 0000H |
| F043DH | | RMDF29H | | | — | √ | | |
| F043EH | CAN receive rule entry register 13AH[Note 1] | GAFLIDH13L | GAFLIDH13 | R/W | — | √ | √ | 0000H |
| F043FH | | GAFLIDH13H | | | — | √ | | |
| F043EH | CAN receive buffer register 9DH[Note 2] | RMDF39L | RMDF39 | R | — | √ | √ | 0000H |
| F043FH | | RMDF39H | | | — | √ | | |
| F0440H | CAN receive rule entry register 13BL[Note 1] | GAFLML13L | GAFLML13 | R/W | — | √ | √ | 0000H |
| F0441H | | GAFLML13H | | | — | √ | | |
| F0440H | CAN receive buffer register 10AL[Note 2] | RMIDL10L | RMIDL10 | R | — | √ | √ | 0000H |
| F0441H | | RMIDL10H | | | — | √ | | |
| F0442H | CAN receive rule entry register 13BH[Note 1] | GAFLMH13L | GAFLMH13 | R/W | — | √ | √ | 0000H |
| F0443H | | GAFLMH13H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (11/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F0442H | CAN receive buffer register 10AH[Note 2] | RMIDH10L | RMIDH10 | R | — | √ | √ | 0000H |
| F0443H | | RMIDH10H | | | — | √ | | |
| F0444H | CAN receive rule entry register 13CL[Note 1] | GAFLPL13L | GAFLPL13 | R/W | — | √ | √ | 0000H |
| F0445H | | GAFLPL13H | | | — | √ | | |
| F0444H | CAN receive buffer register 10BL[Note 2] | RMTS10L | RMTS10 | R | — | √ | √ | 0000H |
| F0445H | | RMTS10H | | | — | √ | | |
| F0446H | CAN receive rule entry register 13CH[Note 1] | GAFLPH13L | GAFLPH13 | R/W | — | √ | √ | 0000H |
| F0447H | | GAFLPH13H | | | — | √ | | |
| F0446H | CAN receive buffer register 10BH[Note 2] | RMPTR10L | RMPTR10 | R | — | √ | √ | 0000H |
| F0447H | | RMPTR10H | | | — | √ | | |
| F0448H | CAN receive rule entry register 14AL[Note 1] | GAFLIDL14L | GAFLIDL14 | R/W | — | √ | √ | 0000H |
| F0449H | | GAFLIDL14H | | | — | √ | | |
| F0448H | CAN receive buffer register 10CL[Note 2] | RMDF010L | RMDF010 | R | — | √ | √ | 0000H |
| F0449H | | RMDF010H | | | — | √ | | |
| F044AH | CAN receive rule entry register 14AH[Note 1] | GAFLIDH14L | GAFLIDH14 | R/W | — | √ | √ | 0000H |
| F044BH | | GAFLIDH14H | | | — | √ | | |
| F044AH | CAN receive buffer register 10CH[Note 2] | RMDF110L | RMDF110 | R | — | √ | √ | 0000H |
| F044BH | | RMDF110H | | | — | √ | | |
| F044CH | CAN receive rule entry register 14BL[Note 1] | GAFLML14L | GAFLML14 | R/W | — | √ | √ | 0000H |
| F044DH | | GAFLML14H | | | — | √ | | |
| F044CH | CAN receive buffer register 10DL[Note 2] | RMDF210L | RMDF210 | R | — | √ | √ | 0000H |
| F044DH | | RMDF210H | | | — | √ | | |
| F044EH | CAN receive rule entry register 14BH[Note 1] | GAFLMH14L | GAFLMH14 | R/W | — | √ | √ | 0000H |
| F044FH | | GAFLMH14H | | | — | √ | | |
| F044EH | CAN receive buffer register 10DH[Note 2] | RMDF310L | RMDF310 | R | — | √ | √ | 0000H |
| F044FH | | RMDF310H | | | — | √ | | |
| F0450H | CAN receive rule entry register 14CL[Note 1] | GAFLPL14L | GAFLPL14 | R/W | — | √ | √ | 0000H |
| F0451H | | GAFLPL14H | | | — | √ | | |
| F0450H | CAN receive buffer register 11AL[Note 2] | RMIDL11L | RMIDL11 | R | — | √ | √ | 0000H |
| F0451H | | RMIDL11H | | | — | √ | | |
| F0452H | CAN receive rule entry register 14CH[Note 1] | GAFLPH14L | GAFLPH14 | R/W | — | √ | √ | 0000H |
| F0453H | | GAFLPH14H | | | — | √ | | |
| F0452H | CAN receive buffer register 11AH[Note 2] | RMIDH11L | RMIDH11 | R | — | √ | √ | 0000H |
| F0453H | | RMIDH11H | | | — | √ | | |
| F0454H | CAN receive rule entry register 15AL[Note 1] | GAFLIDL15L | GAFLIDL15 | R/W | — | √ | √ | 0000H |
| F0455H | | GAFLIDL15H | | | — | √ | | |
| F0454H | CAN receive buffer register 11BL[Note 2] | RMTS11L | RMTS11 | R | — | √ | √ | 0000H |
| F0455H | | RMTS11H | | | — | √ | | |
| F0456H | CAN receive rule entry register 15AH[Note 1] | GAFLIDH15L | GAFLIDH15 | R/W | — | √ | √ | 0000H |
| F0457H | | GAFLIDH15H | | | — | √ | | |
| F0456H | CAN receive buffer register 11BH[Note 2] | RMPTR11L | RMPTR11 | R | — | √ | √ | 0000H |
| F0457H | | RMPTR11H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (12/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F0458H | CAN receive rule entry register 15BL[Note 1] | GAFLML15L | GAFLML15 | R/W | — | √ | √ | 0000H |
| F0459H | | GAFLML15H | | | — | √ | | |
| F0458H | CAN receive buffer register 11CL[Note 2] | RMDF011L | RMDF011 | R | — | √ | √ | 0000H |
| F0459H | | RMDF011H | | | — | √ | | |
| F045AH | CAN receive rule entry register 15BH[Note 1] | GAFLMH15L | GAFLMH15 | R/W | — | √ | √ | 0000H |
| F045BH | | GAFLMH15H | | | — | √ | | |
| F045AH | CAN receive buffer register 11CH[Note 2] | RMDF111L | RMDF111 | R | — | √ | √ | 0000H |
| F045BH | | RMDF111H | | | — | √ | | |
| F045CH | CAN receive rule entry register 15CL[Note 1] | GAFLPL15L | GAFLPL15 | R/W | — | √ | √ | 0000H |
| F045DH | | GAFLPL15H | | | — | √ | | |
| F045CH | CAN receive buffer register 11DL[Note 2] | RMDF211L | RMDF211 | R | — | √ | √ | 0000H |
| F045DH | | RMDF211H | | | — | √ | | |
| F045EH | CAN receive rule entry register 15CH[Note 1] | GAFLPH15L | GAFLPH15 | R/W | — | √ | √ | 0000H |
| F045FH | | GAFLPH15H | | | — | √ | | |
| F045EH | CAN receive buffer register 11DH[Note 2] | RMDF311L | RMDF311 | R | — | √ | √ | 0000H |
| F045FH | | RMDF311H | | | — | √ | | |
| F0460H | CAN receive buffer register 12AL[Note 2] | RMIDL12L | RMIDL12 | R | — | √ | √ | 0000H |
| F0461H | | RMIDL12H | | | — | √ | | |
| F0462H | CAN receive buffer register 12AH[Note 2] | RMIDH12L | RMIDH12 | R | — | √ | √ | 0000H |
| F0463H | | RMIDH12H | | | — | √ | | |
| F0464H | CAN receive buffer register 12BL[Note 2] | RMTS12L | RMTS12 | R | — | √ | √ | 0000H |
| F0465H | | RMTS12H | | | — | √ | | |
| F0466H | CAN receive buffer register 12BH[Note 2] | RMPTR12L | RMPTR12 | R | — | √ | √ | 0000H |
| F0467H | | RMPTR12H | | | — | √ | | |
| F0468H | CAN receive buffer register 12CL[Note 2] | RMDF012L | RMDF012 | R | — | √ | √ | 0000H |
| F0469H | | RMDF012H | | | — | √ | | |
| F046AH | CAN receive buffer register 12CH[Note 2] | RMDF112L | RMDF112 | R | — | √ | √ | 0000H |
| F046BH | | RMDF112H | | | — | √ | | |
| F046CH | CAN receive buffer register 12DL[Note 2] | RMDF212L | RMDF212 | R | — | √ | √ | 0000H |
| F046DH | | RMDF212H | | | — | √ | | |
| F046EH | CAN receive buffer register 12DH[Note 2] | RMDF312L | RMDF312 | R | — | √ | √ | 0000H |
| F046FH | | RMDF312H | | | — | √ | | |
| F0470H | CAN receive buffer register 13AL[Note 2] | RMIDL13L | RMIDL13 | R | — | √ | √ | 0000H |
| F0471H | | RMIDL13H | | | — | √ | | |
| F0472H | CAN receive buffer register 13AH[Note 2] | RMIDH13L | RMIDH13 | R | — | √ | √ | 0000H |
| F0473H | | RMIDH13H | | | — | √ | | |
| F0474H | CAN receive buffer register 13BL[Note 2] | RMTS13L | RMTS13 | R | — | √ | √ | 0000H |
| F0475H | | RMTS13H | | | — | √ | | |
| F0476H | CAN receive buffer register 13BH[Note 2] | RMPTR13L | RMPTR13 | R | — | √ | √ | 0000H |
| F0477H | | RMPTR13H | | | — | √ | | |
| F0478H | CAN receive buffer register 13CL[Note 2] | RMDF013L | RMDF013 | R | — | √ | √ | 0000H |
| F0479H | | RMDF013H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (13/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F047AH | CAN receive buffer register 13CH[Note 2] | RMDF113L | RMDF113 | R | — | √ | √ | 0000H |
| F047BH | | RMDF113H | | | — | √ | | |
| F047CH | CAN receive buffer register 13DL[Note 2] | RMDF213L | RMDF213 | R | — | √ | √ | 0000H |
| F047DH | | RMDF213H | | | — | √ | | |
| F047EH | CAN receive buffer register 13DH[Note 2] | RMDF313L | RMDF313 | R | — | √ | √ | 0000H |
| F047FH | | RMDF313H | | | — | √ | | |
| F0480H | CAN receive buffer register 14AL[Note 2] | RMIDL14L | RMIDL14 | R | — | √ | √ | 0000H |
| F0481H | | RMIDL14H | | | — | √ | | |
| F0482H | CAN receive buffer register 14AH[Note 2] | RMIDH14L | RMIDH14 | R | — | √ | √ | 0000H |
| F0483H | | RMIDH14H | | | — | √ | | |
| F0484H | CAN receive buffer register 14BL[Note 2] | RMTS14L | RMTS14 | R | — | √ | √ | 0000H |
| F0485H | | RMTS14H | | | — | √ | | |
| F0486H | CAN receive buffer register 14BH[Note 2] | RMPTR14L | RMPTR14 | R | — | √ | √ | 0000H |
| F0487H | | RMPTR14H | | | — | √ | | |
| F0488H | CAN receive buffer register 14CL[Note 2] | RMDF014L | RMDF014 | R | — | √ | √ | 0000H |
| F0489H | | RMDF014H | | | — | √ | | |
| F048AH | CAN receive buffer register 14CH[Note 2] | RMDF114L | RMDF114 | R | — | √ | √ | 0000H |
| F048BH | | RMDF114H | | | — | √ | | |
| F048CH | CAN receive buffer register 14DL[Note 2] | RMDF214L | RMDF214 | R | — | √ | √ | 0000H |
| F048DH | | RMDF214H | | | — | √ | | |
| F048EH | CAN receive buffer register 14DH[Note 2] | RMDF314L | RMDF314 | R | — | √ | √ | 0000H |
| F048FH | | RMDF314H | | | — | √ | | |
| F0490H | CAN receive buffer register 15AL[Note 2] | RMIDL15L | RMIDL15 | R | — | √ | √ | 0000H |
| F0491H | | RMIDL15H | | | — | √ | | |
| F0492H | CAN receive buffer register 15AH[Note 2] | RMIDH15L | RMIDH15 | R | — | √ | √ | 0000H |
| F0493H | | RMIDH15H | | | — | √ | | |
| F0494H | CAN receive buffer register 15BL[Note 2] | RMTS15L | RMTS15 | R | — | √ | √ | 0000H |
| F0495H | | RMTS15H | | | — | √ | | |
| F0496H | CAN receive buffer register 15BH[Note 2] | RMPTR15L | RMPTR15 | R | — | √ | √ | 0000H |
| F0497H | | RMPTR15H | | | — | √ | | |
| F0498H | CAN receive buffer register 15CL[Note 2] | RMDF015L | RMDF015 | R | — | √ | √ | 0000H |
| F0499H | | RMDF015H | | | — | √ | | |
| F049AH | CAN receive buffer register 15CH[Note 2] | RMDF115L | RMDF115 | R | — | √ | √ | 0000H |
| F049BH | | RMDF115H | | | — | √ | | |
| F049CH | CAN receive buffer register 15DL[Note 2] | RMDF215L | RMDF215 | R | — | √ | √ | 0000H |
| F049DH | | RMDF215H | | | — | √ | | |
| F049EH | CAN receive buffer register 15DH[Note 2] | RMDF315L | RMDF315 | R | — | √ | √ | 0000H |
| F049FH | | RMDF315H | | | — | √ | | |
| F0580H | CAN RAM test register 0[Note 1] | RPGACC0L | RPGACC0 | R/W | — | √ | √ | 0000H |
| F0581H | | RPGACC0H | | | — | √ | | |
| F0582H | CAN RAM test register 1[Note 1] | RPGACC1L | RPGACC1 | R/W | — | √ | √ | 0000H |
| F0583H | | RPGACC1H | | | — | √ | | |
| F0584H | CAN RAM test register 2[Note 1] | RPGACC2L | RPGACC2 | R/W | — | √ | √ | 0000H |
| F0585H | | RPGACC2H | | | — | √ | | |
| F0586H | CAN RAM test register 3[Note 1] | RPGACC3L | RPGACC3 | R/W | — | √ | √ | 0000H |
| F0587H | | RPGACC3H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (14/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F0588H | CAN RAM test register 4[Note 1] | RPGACC4L | RPGACC4 | R/W | — | √ | √ | 0000H |
| F0589H | | RPGACC4H | | | — | √ | | |
| F058AH | CAN RAM test register 5[Note 1] | RPGACC5L | RPGACC5 | R/W | — | √ | √ | 0000H |
| F058BH | | RPGACC5H | | | — | √ | | |
| F058CH | CAN RAM test register 6[Note 1] | RPGACC6L | RPGACC6 | R/W | — | √ | √ | 0000H |
| F058DH | | RPGACC6H | | | — | √ | | |
| F058EH | CAN RAM test register 7[Note 1] | RPGACC7L | RPGACC7 | R/W | — | √ | √ | 0000H |
| F058FH | | RPGACC7H | | | — | √ | | |
| F0590H | CAN RAM test register 8[Note 1] | RPGACC8L | RPGACC8 | R/W | — | √ | √ | 0000H |
| F0591H | | RPGACC8H | | | — | √ | | |
| F0592H | CAN RAM test register 9[Note 1] | RPGACC9L | RPGACC9 | R/W | — | √ | √ | 0000H |
| F0593H | | RPGACC9H | | | — | √ | | |
| F0594H | CAN RAM test register 10[Note 1] | RPGACC10L | RPGACC10 | R/W | — | √ | √ | 0000H |
| F0595H | | RPGACC10H | | | — | √ | | |
| F0596H | CAN RAM test register 11[Note 1] | RPGACC11L | RPGACC11 | R/W | — | √ | √ | 0000H |
| F0597H | | RPGACC11H | | | — | √ | | |
| F0598H | CAN RAM test register 12[Note 1] | RPGACC12L | RPGACC12 | R/W | — | √ | √ | 0000H |
| F0599H | | RPGACC12H | | | — | √ | | |
| F059AH | CAN RAM test register 13[Note 1] | RPGACC13L | RPGACC13 | R/W | — | √ | √ | 0000H |
| F059BH | | RPGACC13H | | | — | √ | | |
| F059CH | CAN RAM test register 14[Note 1] | RPGACC14L | RPGACC14 | R/W | — | √ | √ | 0000H |
| F059DH | | RPGACC14H | | | — | √ | | |
| F059EH | CAN RAM test register 15[Note 1] | RPGACC15L | RPGACC15 | R/W | — | √ | √ | 0000H |
| F059FH | | RPGACC15H | | | — | √ | | |
| F05A0H | CAN RAM test register 16[Note 1] | RPGACC16L | RPGACC16 | R/W | — | √ | √ | 0000H |
| F05A1H | | RPGACC16H | | | — | √ | | |
| F05A0H | CAN receive FIFO access register 0AL[Note 2] | RFIDL0L | RFIDL0 | R | — | √ | √ | 0000H |
| F05A1H | | RFIDL0H | | | — | √ | | |
| F05A2H | CAN RAM test register 17[Note 1] | RPGACC17L | RPGACC17 | R/W | — | √ | √ | 0000H |
| F05A3H | | RPGACC17H | | | — | √ | | |
| F05A2H | CAN receive FIFO access register 0AH[Note 2] | RFIDH0L | RFIDH0 | R | — | √ | √ | 0000H |
| F05A3H | | RFIDH0H | | | — | √ | | |
| F05A4H | CAN RAM test register 18[Note 1] | RPGACC18L | RPGACC18 | R/W | — | √ | √ | 0000H |
| F05A5H | | RPGACC18H | | | — | √ | | |
| F05A4H | CAN receive FIFO access register 0BL[Note 2] | RFTS0L | RFTS0 | R | — | √ | √ | 0000H |
| F05A5H | | RFTS0H | | | — | √ | | |
| F05A6H | CAN RAM test register 19[Note 1] | RPGACC19L | RPGACC19 | R/W | — | √ | √ | 0000H |
| F05A7H | | RPGACC19H | | | — | √ | | |
| F05A6H | CAN receive FIFO access register 0BH[Note 2] | RFPTR0L | RFPTR0 | R | — | √ | √ | 0000H |
| F05A7H | | RFPTR0H | | | — | √ | | |
| F05A8H | CAN RAM test register 20[Note 1] | RPGACC20L | RPGACC20 | R/W | — | √ | √ | 0000H |
| F05A9H | | RPGACC20H | | | — | √ | | |
| F05A8H | CAN receive FIFO access register 0CL[Note 2] | RFDF00L | RFDF00 | R | — | √ | √ | 0000H |
| F05A9H | | RFDF00H | | | — | √ | | |
| F05AAH | CAN RAM test register 21[Note 1] | RPGACC21L | RPGACC21 | R/W | — | √ | √ | 0000H |
| F05ABH | | RPGACC21H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (15/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F05AAH | CAN receive FIFO access register 0CH[Note 2] | RFDF10L | RFDF10 | R | — | √ | √ | 0000H |
| F05ABH | | RFDF10H | | | — | √ | | |
| F05ACH | CAN RAM test register 22[Note 1] | RPGACC22L | RPGACC22 | R/W | — | √ | √ | 0000H |
| F05ADH | | RPGACC22H | | | — | √ | | |
| F05ACH | CAN receive FIFO access register 0DL[Note 2] | RFDF20L | RFDF20 | R | — | √ | √ | 0000H |
| F05ADH | | RFDF20H | | | — | √ | | |
| F05AEH | CAN RAM test register 23[Note 1] | RPGACC23L | RPGACC23 | R/W | — | √ | √ | 0000H |
| F05AFH | | RPGACC23H | | | — | √ | | |
| F05AEH | CAN receive FIFO access register 0DH[Note 2] | RFDF30L | RFDF30 | R | — | √ | √ | 0000H |
| F05AFH | | RFDF30H | | | — | √ | | |
| F05B0H | CAN RAM test register 24[Note 1] | RPGACC24L | RPGACC24 | R/W | — | √ | √ | 0000H |
| F05B1H | | RPGACC24H | | | — | √ | | |
| F05B0H | CAN receive FIFO access register 1AL[Note 2] | RFIDL1L | RFIDL1 | R | — | √ | √ | 0000H |
| F05B1H | | RFIDL1H | | | — | √ | | |
| F05B2H | CAN RAM test register 25[Note 1] | RPGACC25L | RPGACC25 | R/W | — | √ | √ | 0000H |
| F05B3H | | RPGACC25H | | | — | √ | | |
| F05B2H | CAN receive FIFO access register 1AH[Note 2] | RFIDH1L | RFIDH1 | R | — | √ | √ | 0000H |
| F05B3H | | RFIDH1H | | | — | √ | | |
| F05B4H | CAN RAM test register 26[Note 1] | RPGACC26L | RPGACC26 | R/W | — | √ | √ | 0000H |
| F05B5H | | RPGACC26H | | | — | √ | | |
| F05B4H | CAN receive FIFO access register 1BL[Note 2] | RFTS1L | RFTS1 | R | — | √ | √ | 0000H |
| F05B5H | | RFTS1H | | | — | √ | | |
| F05B6H | CAN RAM test register 27[Note 1] | RPGACC27L | RPGACC27 | R/W | — | √ | √ | 0000H |
| F05B7H | | RPGACC27H | | | — | √ | | |
| F05B6H | CAN receive FIFO access register 1BH[Note 2] | RFPTR1L | RFPTR1 | R | — | √ | √ | 0000H |
| F05B7H | | RFPTR1H | | | — | √ | | |
| F05B8H | CAN RAM test register 28[Note 1] | RPGACC28L | RPGACC28 | R/W | — | √ | √ | 0000H |
| F05B9H | | RPGACC28H | | | — | √ | | |
| F05B8H | CAN receive FIFO access register 1CL[Note 2] | RFDF01L | RFDF01 | R | — | √ | √ | 0000H |
| F05B9H | | RFDF01H | | | — | √ | | |
| F05BAH | CAN RAM test register 29[Note 1] | RPGACC29L | RPGACC29 | R/W | — | √ | √ | 0000H |
| F05BBH | | RPGACC29H | | | — | √ | | |
| F05BAH | CAN receive FIFO access register 1CH[Note 2] | RFDF11L | RFDF11 | R | — | √ | √ | 0000H |
| F05BBH | | RFDF11H | | | — | √ | | |
| F05BCH | CAN RAM test register 30[Note 1] | RPGACC30L | RPGACC30 | R/W | — | √ | √ | 0000H |
| F05BDH | | RPGACC30H | | | — | √ | | |
| F05BCH | CAN receive FIFO access register 1DL[Note 2] | RFDF21L | RFDF21 | R | — | √ | √ | 0000H |
| F05BDH | | RFDF21H | | | — | √ | | |
| F05BEH | CAN RAM test register 31[Note 1] | RPGACC31L | RPGACC31 | R/W | — | √ | √ | 0000H |
| F05BFH | | RPGACC31H | | | — | √ | | |
| F05BEH | CAN receive FIFO access register 1DH[Note 2] | RFDF31L | RFDF31 | R | — | √ | √ | 0000H |
| F05BFH | | RFDF31H | | | — | √ | | |
| F05C0H | CAN RAM test register 32[Note 1] | RPGACC32L | RPGACC32 | R/W | — | √ | √ | 0000H |
| F05C1H | | RPGACC32H | | | — | √ | | |
| F05C2H | CAN RAM test register 33[Note 1] | RPGACC33L | RPGACC33 | R/W | — | √ | √ | 0000H |
| F05C3H | | RPGACC33H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (16/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F05C4H | CAN RAM test register 34[Note 1] | RPGACC34L | RPGACC34 | R/W | — | √ | √ | 0000H |
| F05C5H | | RPGACC34H | | | — | √ | | |
| F05C6H | CAN RAM test register 35[Note 1] | RPGACC35L | RPGACC35 | R/W | — | √ | √ | 0000H |
| F05C7H | | RPGACC35H | | | — | √ | | |
| F05C8H | CAN RAM test register 36[Note 1] | RPGACC36L | RPGACC36 | R/W | — | √ | √ | 0000H |
| F05C9H | | RPGACC36H | | | — | √ | | |
| F05CAH | CAN RAM test register 37[Note 1] | RPGACC37L | RPGACC37 | R/W | — | √ | √ | 0000H |
| F05CBH | | RPGACC37H | | | — | √ | | |
| F05CCH | CAN RAM test register 38[Note 1] | RPGACC38L | RPGACC38 | R/W | — | √ | √ | 0000H |
| F05CDH | | RPGACC38H | | | — | √ | | |
| F05CEH | CAN RAM test register 39[Note 1] | RPGACC39L | RPGACC39 | R/W | — | √ | √ | 0000H |
| F05CFH | | RPGACC39H | | | — | √ | | |
| F05D0H | CAN RAM test register 40[Note 1] | RPGACC40L | RPGACC40 | R/W | — | √ | √ | 0000H |
| F05D1H | | RPGACC40H | | | — | √ | | |
| F05D2H | CAN RAM test register 41[Note 1] | RPGACC41L | RPGACC41 | R/W | — | √ | √ | 0000H |
| F05D3H | | RPGACC41H | | | — | √ | | |
| F05D4H | CAN RAM test register 42[Note 1] | RPGACC42L | RPGACC42 | R/W | — | √ | √ | 0000H |
| F05D5H | | RPGACC42H | | | — | √ | | |
| F05D6H | CAN RAM test register 43[Note 1] | RPGACC43L | RPGACC43 | R/W | — | √ | √ | 0000H |
| F05D7H | | RPGACC43H | | | — | √ | | |
| F05D8H | CAN RAM test register 44[Note 1] | RPGACC44L | RPGACC44 | R/W | — | √ | √ | 0000H |
| F05D9H | | RPGACC44H | | | — | √ | | |
| F05DAH | CAN RAM test register 45[Note 1] | RPGACC45L | RPGACC45 | R/W | — | √ | √ | 0000H |
| F05DBH | | RPGACC45H | | | — | √ | | |
| F05DCH | CAN RAM test register 46[Note 1] | RPGACC46L | RPGACC46 | R/W | — | √ | √ | 0000H |
| F05DDH | | RPGACC46H | | | — | √ | | |
| F05DEH | CAN RAM test register 47[Note 1] | RPGACC47L | RPGACC47 | R/W | — | √ | √ | 0000H |
| F05DFH | | RPGACC47H | | | — | √ | | |
| F05E0H | CAN RAM test register 48[Note 1] | RPGACC48L | RPGACC48 | R/W | — | √ | √ | 0000H |
| F05E1H | | RPGACC48H | | | — | √ | | |
| F05E0H | CAN0 transmit/receive FIFO access | CFIDL0L | CFIDL0 | R/W | — | √ | √ | 0000H |
| F05E1H | register 0AL[Note 2] | CFIDL0H | | | — | √ | | |
| F05E2H | CAN RAM test register 49[Note 1] | RPGACC49L | RPGACC49 | R/W | — | √ | √ | 0000H |
| F05E3H | | RPGACC49H | | | — | √ | | |
| F05E2H | CAN0 transmit/receive FIFO access | CFIDH0L | CFIDH0 | R/W | — | √ | √ | 0000H |
| F05E3H | register 0AH[Note 2] | CFIDH0H | | | — | √ | | |
| F05E4H | CAN RAM test register 50[Note 1] | RPGACC50L | RPGACC50 | R/W | — | √ | √ | 0000H |
| F05E5H | | RPGACC50H | | | — | √ | | |
| F05E4H | CAN0 transmit/receive FIFO access | CFTS0L | CFTS0 | R | — | √ | √ | 0000H |
| F05E5H | register 0BL[Note 2] | CFTS0H | | | — | √ | | |
| F05E6H | CAN RAM test register 51[Note 1] | RPGACC51L | RPGACC51 | R/W | — | √ | √ | 0000H |
| F05E7H | | RPGACC51H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (17/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F05E6H | CAN0 transmit/receive FIFO access register 0BH[Note 2] | CFPTR0L | CFPTR0 | R/W | — | √ | √ | 0000H |
| F05E7H | | CFPTR0H | | | — | √ | | |
| F05E8H | CAN RAM test register 52[Note 1] | RPGACC52L | RPGACC52 | R/W | — | √ | √ | 0000H |
| F05E9H | | RPGACC52H | | | — | √ | | |
| F05E8H | CAN0 transmit/receive FIFO access register 0CL[Note 2] | CFDF00L | CFDF00 | R/W | — | √ | √ | 0000H |
| F05E9H | | CFDF00H | | | — | √ | | |
| F05EAH | CAN RAM test register 53[Note 1] | RPGACC53L | RPGACC53 | R/W | — | √ | √ | 0000H |
| F05EBH | | RPGACC53H | | | — | √ | | |
| F05EAH | CAN0 transmit/receive FIFO access register 0CH[Note 2] | CFDF10L | CFDF10 | R/W | — | √ | √ | 0000H |
| F05EBH | | CFDF10H | | | — | √ | | |
| F05ECH | CAN RAM test register 54[Note 1] | RPGACC54L | RPGACC54 | R/W | — | √ | √ | 0000H |
| F05EDH | | RPGACC54H | | | — | √ | | |
| F05ECH | CAN0 transmit/receive FIFO access register 0DL[Note 2] | CFDF20L | CFDF20 | R/W | — | √ | √ | 0000H |
| F05EDH | | CFDF20H | | | — | √ | | |
| F05EEH | CAN RAM test register 55[Note 1] | RPGACC55L | RPGACC55 | R/W | — | √ | √ | 0000H |
| F05EFH | | RPGACC55H | | | — | √ | | |
| F05EEH | CAN0 transmit/receive FIFO access register 0DH[Note 2] | CFDF30L | CFDF30 | R/W | — | √ | √ | 0000H |
| F05EFH | | CFDF30H | | | — | √ | | |
| F05F0H | CAN RAM test register 56[Note 1] | RPGACC56L | RPGACC56 | R/W | — | √ | √ | 0000H |
| F05F1H | | RPGACC56H | | | — | √ | | |
| F05F2H | CAN RAM test register 57[Note 1] | RPGACC57L | RPGACC57 | R/W | — | √ | √ | 0000H |
| F05F3H | | RPGACC57H | | | — | √ | | |
| F05F4H | CAN RAM test register 58[Note 1] | RPGACC58L | RPGACC58 | R/W | — | √ | √ | 0000H |
| F05F5H | | RPGACC58H | | | — | √ | | |
| F05F6H | CAN RAM test register 59[Note 1] | RPGACC59L | RPGACC59 | R/W | — | √ | √ | 0000H |
| F05F7H | | RPGACC59H | | | — | √ | | |
| F05F8H | CAN RAM test register 60[Note 1] | RPGACC60L | RPGACC60 | R/W | — | √ | √ | 0000H |
| F05F9H | | RPGACC60H | | | — | √ | | |
| F05FAH | CAN RAM test register 61[Note 1] | RPGACC61L | RPGACC61 | R/W | — | √ | √ | 0000H |
| F05FBH | | RPGACC61H | | | — | √ | | |
| F05FCH | CAN RAM test register 62[Note 1] | RPGACC62L | RPGACC62 | R/W | — | √ | √ | 0000H |
| F05FDH | | RPGACC62H | | | — | √ | | |
| F05FEH | CAN RAM test register 63[Note 1] | RPGACC63L | RPGACC63 | R/W | — | √ | √ | 0000H |
| F05FFH | | RPGACC63H | | | — | √ | | |
| F0600H | CAN RAM test register 64[Note 1] | RPGACC64L | RPGACC64 | R/W | — | √ | √ | 0000H |
| F0601H | | RPGACC64H | | | — | √ | | |
| F0600H | CAN0 transmit buffer register 0AL[Note 2] | TMIDL0L | TMIDL0 | R/W | — | √ | √ | 0000H |
| F0601H | | TMIDL0H | | | — | √ | | |
| F0602H | CAN RAM test register 65[Note 1] | RPGACC65L | RPGACC65 | R/W | — | √ | √ | 0000H |
| F0603H | | RPGACC65H | | | — | √ | | |
| F0602H | CAN0 transmit buffer register 0AH[Note 2] | TMIDH0L | TMIDH0 | R/W | — | √ | √ | 0000H |
| F0603H | | TMIDH0H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (18/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F0604H | CAN RAM test register 66[Note 1] | RPGACC66L | RPGACC66 | R/W | — | √ | √ | 0000H |
| F0605H | | RPGACC66H | | | — | √ | | |
| F0606H | CAN RAM test register 67[Note 1] | RPGACC67L | RPGACC67 | R/W | — | √ | √ | 0000H |
| F0607H | | RPGACC67H | | | — | √ | | |
| F0606H | CAN0 transmit buffer register 0BH[Note 2] | TMPTR0L | TMPTR0 | R/W | — | √ | √ | 0000H |
| F0607H | | TMPTR0H | | | — | √ | | |
| F0608H | CAN RAM test register 68[Note 1] | RPGACC68L | RPGACC68 | R/W | — | √ | √ | 0000H |
| F0609H | | RPGACC68H | | | — | √ | | |
| F0608H | CAN0 transmit buffer register 0CL[Note 2] | TMDF00L | TMDF00 | R/W | — | √ | √ | 0000H |
| F0609H | | TMDF00H | | | — | √ | | |
| F060AH | CAN RAM test register 69[Note 1] | RPGACC69L | RPGACC69 | R/W | — | √ | √ | 0000H |
| F060BH | | RPGACC69H | | | — | √ | | |
| F060AH | CAN0 transmit buffer register 0CH[Note 2] | TMDF10L | TMDF10 | R/W | — | √ | √ | 0000H |
| F060BH | | TMDF10H | | | — | √ | | |
| F060CH | CAN RAM test register 70[Note 1] | RPGACC70L | RPGACC70 | R/W | — | √ | √ | 0000H |
| F060DH | | RPGACC70H | | | — | √ | | |
| F060CH | CAN0 transmit buffer register 0DL[Note 2] | TMDF20L | TMDF20 | R/W | — | √ | √ | 0000H |
| F060DH | | TMDF20H | | | — | √ | | |
| F060EH | CAN RAM test register 71[Note 1] | RPGACC71L | RPGACC71 | R/W | — | √ | √ | 0000H |
| F060FH | | RPGACC71H | | | — | √ | | |
| F060EH | CAN0 transmit buffer register 0DH[Note 2] | TMDF30L | TMDF30 | R/W | — | √ | √ | 0000H |
| F060FH | | TMDF30H | | | — | √ | | |
| F0610H | CAN RAM test register 72[Note 1] | RPGACC72L | RPGACC72 | R/W | — | √ | √ | 0000H |
| F0611H | | RPGACC72H | | | — | √ | | |
| F0610H | CAN0 transmit buffer register 1AL[Note 2] | TMIDL1L | TMIDL1 | R/W | — | √ | √ | 0000H |
| F0611H | | TMIDL1H | | | — | √ | | |
| F0612H | CAN RAM test register 73[Note 1] | RPGACC73L | RPGACC73 | R/W | — | √ | √ | 0000H |
| F0613H | | RPGACC73H | | | — | √ | | |
| F0612H | CAN0 transmit buffer register 1AH[Note 2] | TMIDH1L | TMIDH1 | R/W | — | √ | √ | 0000H |
| F0613H | | TMIDH1H | | | — | √ | | |
| F0614H | CAN RAM test register 74[Note 1] | RPGACC74L | RPGACC74 | R/W | — | √ | √ | 0000H |
| F0615H | | RPGACC74H | | | — | √ | | |
| F0616H | CAN RAM test register 75[Note 1] | RPGACC75L | RPGACC75 | R/W | — | √ | √ | 0000H |
| F0617H | | RPGACC75H | | | — | √ | | |
| F0616H | CAN0 transmit buffer register 1BH[Note 2] | TMPTR1L | TMPTR1 | R/W | — | √ | √ | 0000H |
| F0617H | | TMPTR1H | | | — | √ | | |
| F0618H | CAN RAM test register 76[Note 1] | RPGACC76L | RPGACC76 | R/W | — | √ | √ | 0000H |
| F0619H | | RPGACC76H | | | — | √ | | |
| F0618H | CAN0 transmit buffer register 1CL[Note 2] | TMDF01L | TMDF01 | R/W | — | √ | √ | 0000H |
| F0619H | | TMDF01H | | | — | √ | | |
| F061AH | CAN RAM test register 77[Note 1] | RPGACC77L | RPGACC77 | R/W | — | √ | √ | 0000H |
| F061BH | | RPGACC77H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (19/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F061AH | CAN0 transmit buffer register 1CH[Note 2] | TMDF11L | TMDF11 | R/W | — | √ | √ | 0000H |
| F061BH | | TMDF11H | | | — | √ | | |
| F061CH | CAN RAM test register 78[Note 1] | RPGACC78L | RPGACC78 | R/W | — | √ | √ | 0000H |
| F061DH | | RPGACC78H | | | — | √ | | |
| F061CH | CAN0 transmit buffer register 1DL[Note 2] | TMDF21L | TMDF21 | R/W | — | √ | √ | 0000H |
| F061DH | | TMDF21H | | | — | √ | | |
| F061EH | CAN RAM test register 79[Note 1] | RPGACC79L | RPGACC79 | R/W | — | √ | √ | 0000H |
| F061FH | | RPGACC79H | | | — | √ | | |
| F061EH | CAN0 transmit buffer register 1DH[Note 2] | TMDF31L | TMDF31 | R/W | — | √ | √ | 0000H |
| F061FH | | TMDF31H | | | — | √ | | |
| F0620H | CAN RAM test register 80[Note 1] | RPGACC80L | RPGACC80 | R/W | — | √ | √ | 0000H |
| F0621H | | RPGACC80H | | | — | √ | | |
| F0620H | CAN0 transmit buffer register 2AL[Note 2] | TMIDL2L | TMIDL2 | R/W | — | √ | √ | 0000H |
| F0621H | | TMIDL2H | | | — | √ | | |
| F0622H | CAN RAM test register 81[Note 1] | RPGACC81L | RPGACC81 | R/W | — | √ | √ | 0000H |
| F0623H | | RPGACC81H | | | — | √ | | |
| F0622H | CAN0 transmit buffer register 2AH[Note 2] | TMIDH2L | TMIDH2 | R/W | — | √ | √ | 0000H |
| F0623H | | TMIDH2H | | | — | √ | | |
| F0624H | CAN RAM test register 82[Note 1] | RPGACC82L | RPGACC82 | R/W | — | √ | √ | 0000H |
| F0625H | | RPGACC82H | | | — | √ | | |
| F0626H | CAN RAM test register 83[Note 1] | RPGACC83L | RPGACC83 | R/W | — | √ | √ | 0000H |
| F0627H | | RPGACC83H | | | — | √ | | |
| F0626H | CAN0 transmit buffer register 2BH[Note 2] | TMPTR2L | TMPTR2 | R/W | — | √ | √ | 0000H |
| F0627H | | TMPTR2H | | | — | √ | | |
| F0628H | CAN RAM test register 84[Note 1] | RPGACC84L | RPGACC84 | R/W | — | √ | √ | 0000H |
| F0629H | | RPGACC84H | | | — | √ | | |
| F0628H | CAN0 transmit buffer register 2CL[Note 2] | TMDF02L | TMDF02 | R/W | — | √ | √ | 0000H |
| F0629H | | TMDF02H | | | — | √ | | |
| F062AH | CAN RAM test register 85[Note 1] | RPGACC85L | RPGACC85 | R/W | — | √ | √ | 0000H |
| F062BH | | RPGACC85H | | | — | √ | | |
| F062AH | CAN0 transmit buffer register 2CH[Note 2] | TMDF12L | TMDF12 | R/W | — | √ | √ | 0000H |
| F062BH | | TMDF12H | | | — | √ | | |
| F062CH | CAN RAM test register 86[Note 1] | RPGACC86L | RPGACC86 | R/W | — | √ | √ | 0000H |
| F062DH | | RPGACC86H | | | — | √ | | |
| F062CH | CAN0 transmit buffer register 2DL[Note 2] | TMDF22L | TMDF22 | R/W | — | √ | √ | 0000H |
| F062DH | | TMDF22H | | | — | √ | | |
| F062EH | CAN RAM test register 87[Note 1] | RPGACC87L | RPGACC87 | R/W | — | √ | √ | 0000H |
| F062FH | | RPGACC87H | | | — | √ | | |
| F062EH | CAN0 transmit buffer register 2DH[Note 2] | TMDF32L | TMDF32 | R/W | — | √ | √ | 0000H |
| F062FH | | TMDF32H | | | — | √ | | |
| F0630H | CAN RAM test register 88[Note 1] | RPGACC88L | RPGACC88 | R/W | — | √ | √ | 0000H |
| F0631H | | RPGACC88H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (20/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F0630H | CAN0 transmit buffer register 3AL[Note 2] | TMIDL3L | TMIDL3 | R/W | — | √ | √ | 0000H |
| F0631H | | TMIDL3H | | | — | √ | | |
| F0632H | CAN RAM test register 89[Note 1] | RPGACC89L | RPGACC89 | R/W | — | √ | √ | 0000H |
| F0633H | | RPGACC89H | | | — | √ | | |
| F0632H | CAN0 transmit buffer register 3AH[Note 2] | TMIDH3L | TMIDH3 | R/W | — | √ | √ | 0000H |
| F0633H | | TMIDH3H | | | — | √ | | |
| F0634H | CAN RAM test register 90[Note 1] | RPGACC90L | RPGACC90 | R/W | — | √ | √ | 0000H |
| F0635H | | RPGACC90H | | | — | √ | | |
| F0636H | CAN RAM test register 91[Note 1] | RPGACC91L | RPGACC91 | R/W | — | √ | √ | 0000H |
| F0637H | | RPGACC91H | | | — | √ | | |
| F0636H | CAN0 transmit buffer register 3BH[Note 2] | TMPTR3L | TMPTR3 | R/W | — | √ | √ | 0000H |
| F0637H | | TMPTR3H | | | — | √ | | |
| F0638H | CAN RAM test register 92[Note 1] | RPGACC92L | RPGACC92 | R/W | — | √ | √ | 0000H |
| F0639H | | RPGACC92H | | | — | √ | | |
| F0638H | CAN0 transmit buffer register 3CL[Note 2] | TMDF03L | TMDF03 | R/W | — | √ | √ | 0000H |
| F0639H | | TMDF03H | | | — | √ | | |
| F063AH | CAN RAM test register 93[Note 1] | RPGACC93L | RPGACC93 | R/W | — | √ | √ | 0000H |
| F063BH | | RPGACC93H | | | — | √ | | |
| F063AH | CAN0 transmit buffer register 3CH[Note 2] | TMDF13L | TMDF13 | R/W | — | √ | √ | 0000H |
| F063BH | | TMDF13H | | | — | √ | | |
| F063CH | CAN RAM test register 94[Note 1] | RPGACC94L | RPGACC94 | R/W | — | √ | √ | 0000H |
| F063DH | | RPGACC94H | | | — | √ | | |
| F063CH | CAN0 transmit buffer register 3DL[Note 2] | TMDF23L | TMDF23 | R/W | — | √ | √ | 0000H |
| F063DH | | TMDF23H | | | — | √ | | |
| F063EH | CAN RAM test register 95[Note 1] | RPGACC95L | RPGACC95 | R/W | — | √ | √ | 0000H |
| F063FH | | RPGACC95H | | | — | √ | | |
| F063EH | CAN0 transmit buffer register 3DH[Note 2] | TMDF33L | TMDF33 | R/W | — | √ | √ | 0000H |
| F063FH | | TMDF33H | | | — | √ | | |
| F0640H | CAN RAM test register 96[Note 1] | RPGACC96L | RPGACC96 | R/W | — | √ | √ | 0000H |
| F0641H | | RPGACC96H | | | — | √ | | |
| F0642H | CAN RAM test register 97[Note 1] | RPGACC97L | RPGACC97 | R/W | — | √ | √ | 0000H |
| F0643H | | RPGACC97H | | | — | √ | | |
| F0644H | CAN RAM test register 98[Note 1] | RPGACC98L | RPGACC98 | R/W | — | √ | √ | 0000H |
| F0645H | | RPGACC98H | | | — | √ | | |
| F0646H | CAN RAM test register 99[Note 1] | RPGACC99L | RPGACC99 | R/W | — | √ | √ | 0000H |
| F0647H | | RPGACC99H | | | — | √ | | |
| F0648H | CAN RAM test register 100[Note 1] | RPGACC100L | RPGACC100 | R/W | — | √ | √ | 0000H |
| F0649H | | RPGACC100H | | | — | √ | | |
| F064AH | CAN RAM test register 101[Note 1] | RPGACC101L | RPGACC101 | R/W | — | √ | √ | 0000H |
| F064BH | | RPGACC101H | | | — | √ | | |
| F064CH | CAN RAM test register 102[Note 1] | RPGACC102L | RPGACC102 | R/W | — | √ | √ | 0000H |
| F064DH | | RPGACC102H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

**Table 18-3. List of CAN Module Registers (21/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F064EH | CAN RAM test register 103[Note] | RPGACC103L | RPGACC103 | R/W | — | √ | √ | 0000H |
| F064FH | | RPGACC103H | | | — | √ | | |
| F0650H | CAN RAM test register 104[Note] | RPGACC104L | RPGACC104 | R/W | — | √ | √ | 0000H |
| F0651H | | RPGACC104H | | | — | √ | | |
| F0652H | CAN RAM test register 105[Note] | RPGACC105L | RPGACC105 | R/W | — | √ | √ | 0000H |
| F0653H | | RPGACC105H | | | — | √ | | |
| F0654H | CAN RAM test register 106[Note] | RPGACC106L | RPGACC106 | R/W | — | √ | √ | 0000H |
| F0655H | | RPGACC106H | | | — | √ | | |
| F0656H | CAN RAM test register 107[Note] | RPGACC107L | RPGACC107 | R/W | — | √ | √ | 0000H |
| F0657H | | RPGACC107H | | | — | √ | | |
| F0658H | CAN RAM test register 108[Note] | RPGACC108L | RPGACC108 | R/W | — | √ | √ | 0000H |
| F0659H | | RPGACC108H | | | — | √ | | |
| F065AH | CAN RAM test register 109[Note] | RPGACC109L | RPGACC109 | R/W | — | √ | √ | 0000H |
| F065BH | | RPGACC109H | | | — | √ | | |
| F065CH | CAN RAM test register 110[Note] | RPGACC110L | RPGACC110 | R/W | — | √ | √ | 0000H |
| F065DH | | RPGACC110H | | | — | √ | | |
| F065EH | CAN RAM test register 111[Note] | RPGACC111L | RPGACC111 | R/W | — | √ | √ | 0000H |
| F065FH | | RPGACC111H | | | — | √ | | |
| F0660H | CAN RAM test register 112[Note] | RPGACC112L | RPGACC112 | R/W | — | √ | √ | 0000H |
| F0661H | | RPGACC112H | | | — | √ | | |
| F0662H | CAN RAM test register 113[Note] | RPGACC113L | RPGACC113 | R/W | — | √ | √ | 0000H |
| F0663H | | RPGACC113H | | | — | √ | | |
| F0664H | CAN RAM test register 114[Note] | RPGACC114L | RPGACC114 | R/W | — | √ | √ | 0000H |
| F0665H | | RPGACC114H | | | — | √ | | |
| F0666H | CAN RAM test register 115[Note] | RPGACC115L | RPGACC115 | R/W | — | √ | √ | 0000H |
| F0667H | | RPGACC115H | | | — | √ | | |
| F0668H | CAN RAM test register 116[Note] | RPGACC116L | RPGACC116 | R/W | — | √ | √ | 0000H |
| F0669H | | RPGACC116H | | | — | √ | | |
| F066AH | CAN RAM test register 117[Note] | RPGACC117L | RPGACC117 | R/W | — | √ | √ | 0000H |
| F066BH | | RPGACC117H | | | — | √ | | |
| F066CH | CAN RAM test register 118[Note] | RPGACC118L | RPGACC118 | R/W | — | √ | √ | 0000H |
| F066DH | | RPGACC118H | | | — | √ | | |
| F066EH | CAN RAM test register 119[Note] | RPGACC119L | RPGACC119 | R/W | — | √ | √ | 0000H |
| F066FH | | RPGACC119H | | | — | √ | | |
| F0670H | CAN RAM test register 120[Note] | RPGACC120L | RPGACC120 | R/W | — | √ | √ | 0000H |
| F0671H | | RPGACC120H | | | — | √ | | |
| F0672H | CAN RAM test register 121[Note] | RPGACC121L | RPGACC121 | R/W | — | √ | √ | 0000H |
| F0673H | | RPGACC121H | | | — | √ | | |
| F0674H | CAN RAM test register 122[Note] | RPGACC122L | RPGACC122 | R/W | — | √ | √ | 0000H |
| F0675H | | RPGACC122H | | | — | √ | | |
| F0676H | CAN RAM test register 123[Note] | RPGACC123L | RPGACC123 | R/W | — | √ | √ | 0000H |
| F0677H | | RPGACC123H | | | — | √ | | |
| F0678H | CAN RAM test register 124[Note] | RPGACC124L | RPGACC124 | R/W | — | √ | √ | 0000H |
| F0679H | | RPGACC124H | | | — | √ | | |

**Note** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**Table 18-3. List of CAN Module Registers (22/22)**

| Address | Special Function Register (SFR) Name | Symbol | | R/W | Access Size | | | After Reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| F067AH | CAN RAM test register 125[Note 1] | RPGACC125L | RPGACC125 | R/W | — | √ | √ | 0000H |
| F067BH | | RPGACC125H | | | — | √ | | |
| F067CH | CAN RAM test register 126[Note 1] | RPGACC126L | RPGACC126 | R/W | — | √ | √ | 0000H |
| F067DH | | RPGACC126H | | | — | √ | | |
| F067EH | CAN RAM test register 127[Note 1] | RPGACC127L | RPGACC127 | R/W | — | √ | √ | 0000H |
| F067FH | | RPGACC127H | | | — | √ | | |
| F0680H | CAN0 transmit history buffer access register[Note 2] | THLACC0L | THLACC0 | R | — | √ | √ | 0000H |
| F0681H | | THLACC0H | | | — | √ | | |

**Notes 1.** These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

**2.** These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

### 18.3.1 CANi Bit Configuration Register L (CiCFGL) (i = 0)

Address C0CFGLL: F0300H, C0CFGLH: F0301H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | BRP[9:0] | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 10 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 9 to 0 | BRP[9:0] | Prescaler Division Ratio Set | When these bits are set to P (0 to 1023), the baud rate prescaler divides $f_{CAN}$ by P + 1. | R/W |

Modify the CiCFGL register only in channel reset mode or channel halt mode. Set this register in channel reset mode before making a transition to channel communication mode or channel halt mode. For setting bit timing, see **18.10 Initial Settings**.

- BRP[9:0] Bits
  The CANi Tq clock ($f_{CANTQi}$) is obtained by the CAN clock ($f_{CAN}$) and setting the clock division ratio with the BRP[9:0] bits and one clock cycle of the CANi Tq clock is 1 Time Quantum (Tq).

### 18.3.2 CANi Bit Configuration Register H (CiCFGH) (i = 0)

Address C0CFGHL: F0302H, C0CFGHH: F0303H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | SJW[1:0] | | — | TSEG2[2:0] | | | TSEG1[3:0] | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 10 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 9, 8 | SJW[1:0] | Resynchronization Jump Width Control | b9 b8<br>0 0 : 1 Tq<br>0 1 : 2 Tq<br>1 0 : 3 Tq<br>1 1 : 4 Tq | R/W |
| 7 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R |
| 6 to 4 | TSEG2 [2:0] | Time Segment 2 Control | b6 b5 b4<br>0 0 0 : Setting prohibited<br>0 0 1 : 2 Tq<br>0 1 0 : 3 Tq<br>0 1 1 : 4 Tq<br>1 0 0 : 5 Tq<br>1 0 1 : 6 Tq<br>1 1 0 : 7 Tq<br>1 1 1 : 8 Tq | R/W |
| 3 to 0 | TSEG1 [3:0] | Time Segment 1 Control | b3 b2 b1 b0<br>0 0 0 0 : Setting prohibited<br>0 0 0 1 : Setting prohibited<br>0 0 1 0 : Setting prohibited<br>0 0 1 1 : 4 Tq<br>0 1 0 0 : 5 Tq<br>0 1 0 1 : 6 Tq<br>0 1 1 0 : 7 Tq<br>0 1 1 1 : 8 Tq<br>1 0 0 0 : 9 Tq<br>1 0 0 1 : 10 Tq<br>1 0 1 0 : 11 Tq<br>1 0 1 1 : 12 Tq<br>1 1 0 0 : 13 Tq<br>1 1 0 1 : 14 Tq<br>1 1 1 0 : 15 Tq<br>1 1 1 1 : 16 Tq | R/W |

Modify the CiCFGH register only in channel reset mode or channel halt mode. Set this register in channel reset mode before making a transition to channel communication mode or channel halt mode. For setting bit timing, see **18.10 Initial Settings**.

- SJW[1:0] Bits
  These bits are used to specify a Tq value for the resynchronization jump width. A value of 1 Tq to 4 Tq can be set. Set a value equal to or smaller than the value of the TSEG2 bits.

- TSEG2[2:0] Bits
  These bits are used to specify a Tq value for the length of phase buffer segment 2 (PHASE_SEG2).
  A value of 2 Tq to 8 Tq can be set. Set a value smaller than the value of the TSEG1 bits.

- TSEG1[3:0] Bits
  These bits are used to specify a Tq value for the total length of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1). A value of 4 Tq to 16 Tq can be set.

### 18.3.3 CANi Control Register L (CiCTRL) (i = 0)

Address C0CTRLL: F0304H, C0CTRLH: F0305H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ALIE | BLIE | OLIE | BORIE | BOEIE | EPIE | EWIE | BEIE | — | — | — | — | RTBO | CSLPR | CHMDC [1:0] | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 | ALIE | Arbitration Lost Interrupt Enable | 0: Arbitration lost interrupt is disabled.<br>1: Arbitration lost interrupt is enabled. | R/W |
| 14 | BLIE | Bus Lock Interrupt Enable | 0: Bus lock interrupt is disabled.<br>1: Bus lock interrupt is enabled. | R/W |
| 13 | OLIE | Overload Frame Transmit Interrupt Enable | 0: Overload frame transmit interrupt is disabled.<br>1: Overload frame transmit interrupt is enabled. | R/W |
| 12 | BORIE | Bus Off Recovery Interrupt Enable | 0: Bus off recovery interrupt is disabled.<br>1: Bus off recovery interrupt is enabled. | R/W |
| 11 | BOEIE | Bus Off Entry Interrupt Enable | 0: Bus off entry interrupt is disabled.<br>1: Bus off entry interrupt is enabled. | R/W |
| 10 | EPIE | Error Passive Interrupt Enable | 0: Error passive interrupt is disabled.<br>1: Error passive interrupt is enabled. | R/W |
| 9 | EWIE | Error Warning Interrupt Enable | 0: Error warning interrupt is disabled.<br>1: Error warning interrupt is enabled. | R/W |
| 8 | BEIE | Protocol Error Interrupt Enable | 0: Protocol error interrupt is disabled.<br>1: Protocol error interrupt is enabled. | R/W |
| 7 to 4 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 3 | RTBO | Forcible Return from Bus-off | When this bit is set to 1, forcible return from the bus off state is made. This bit is always read as 0. | R/W |
| 2 | CSLPR | Channel Stop Mode | 0: Other than channel stop mode<br>1: Channel stop mode | R/W |
| 1, 0 | CHMDC [1:0] | Mode Select | b1 b0<br>0 0: Channel communication mode<br>0 1: Channel reset mode<br>1 0: Channel halt mode<br>1 1: Setting prohibited | R/W |

- ALIE Bit
  When the ALF flag in the CiERFLL register is set to 1 with the ALIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

- BLIE Bit
  When the BLF flag in the CiERFLL register is set to 1 with the BLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

- OLIE Bit
  When the OVLF flag in the CiERFLL register is set to 1 with the OLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

- BORIE Bit
  When the BORF flag in the CiERFLL register is set to 1 with the BORIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

- BOEIE Bit
  When the BOEF flag in the CiERFLL register is set to 1 with the BOEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

- EPIE Bit
  When the EPF flag in the CiERFLL register is set to 1 with the EPIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

- EWIE Bit
  When the EWF flag in the CiERFLL register is set to 1 with the EWIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

- BEIE Bit
  When the BEF flag in the CiERFLL register is set to 1 with the BEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

- RTBO Bit
  Setting this bit to 1 (forcible return from the bus off state) in the bus off state forcibly returns the state from the bus off state to the error active state. This bit is automatically cleared to 0. Setting this bit to 1 clears the TEC[7:0] and REC[7:0] bits in the CiSTSH register to H'00 and also clears the BOSTS flag in the CiSTSL register to 0 (not in bus off state). The other registers remain unchanged. No bus off recovery interrupt request due to return from the bus off state is generated. Use this bit only when the BOM[1:0] bits in the CiCTRH register are B'00 (ISO11898-1 compliant).
  A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the CAN module transitions to the error active state. Set this bit to 1 in channel communication mode.

- CSLPR Bit
  Setting this bit to 1 places the channel in channel stop mode.
  Clearing this bit to 0 makes the channel leave from channel stop mode.
  Do not modify this bit while the CAN channel is in channel communication mode or channel halt mode.

- CHMDC[1:0] Bits
  These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see **18.4.2 Channel Modes.** Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to B'11. When the CAN module has transitioned to channel halt mode depending on the setting of the BOM[1:0] bits, the CHMDC[1:0] bits automatically becomes B'10.

### 18.3.4 CANi Control Register H (CiCTRH) (i = 0)

Address C0CTRHL: F0306H, C0CTRHH: F0307H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | CTMS[1:0] | | CTME | ERRD | BOM[1:0] | | — | — | — | — | TAIE |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 11 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 10, 9 | CTMS[1:0] | Communication Test Mode Select | b10 b9<br>0 0: Standard test mode<br>0 1: Listen-only mode<br>1 0: Self-test mode 0 (external loopback mode)<br>1 1: Self-test mode 1 (internal loopback mode) | R/W |
| 8 | CTME | Communication Test Mode Enable | 0: Communication test mode is disabled.<br>1: Communication test mode is enabled. | R/W |
| 7 | ERRD | Error Display Mode Select | 0: Only the first error is indicated after bits 14 to 8 in the CiERFLL register have all been cleared.<br>1: The error flags of all errors are indicated. | R/W |
| 6, 5 | BOM[1:0] | Bus Off Recovery Mode Select | b6 b5<br>0 0: ISO11898-1 compliant<br>0 1: Entry to channel halt mode at bus-off entry<br>1 0: Entry to channel halt mode at bus-off end<br>1 1: Entry to channel halt mode (in the bus off state) by a program request | R/W |
| 4 to 1 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 0 | TAIE | Transmit Abort Interrupt Enable | 0: Transmit abort interrupt is disabled.<br>1: Transmit abort interrupt is enabled. | R/W |

- CTMS[1:0] Bits
  These bits are used to select a communication test mode. Modify these bits only in channel halt mode. These bits are set to 0 in channel reset mode.

- CTME Bit
  Setting this bit to 1 enables communication test mode. Modify this bit only in channel halt mode. This bit is set to 0 in channel reset mode.

- ERRD Bit
  This bit is used to control display mode of bits 14 to 8 in the CiERFLL register.
  When this bit is clear to 0, only the flags of the first error are set to 1. If two or more errors occur first, all the flags of detected errors are set to 1.
  When this bit is set to 1, all the flags of errors that have occurred are set to 1 regardless of the error occurrence order. Modify this bit only in channel reset mode or channel halt mode.

- BOM[1:0] Bits
  These bits are used to select a bus off recovery mode of the CAN module.
  When the BOM[1:0] bits are set to B'00, return to the error active state from the bus off state is compliant with the CAN specifications. That is, the CAN module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CHMDC[1:0] bits are set to B'10 (channel halt mode) before recessive bits are detected 128 times, the CAN module does not transition to channel halt mode until recessive bits are detected 128 times.
  When the CAN module reaches the bus off state while the BOM[1:0] bits are set to B'01, the CHMDC[1:0] bits in the CiCTRL register are set to B'10 and the CAN module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits in the CiSTSH register are cleared to H'00.
  When the CAN module reaches the bus off state when the BOM[1:0] bits are set to B'10, the CHMDC[1:0] bits are set to B'10 and the CAN module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to H'00.
  When the BOM[1:0] bits are set to B'11 and the CHMDC[1:0] bits are set to B'10 while the CAN module is in the bus off state, the CAN module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to H'00. However, if 11 consecutive recessive bits are detected 128 times and the CAN module has recovered to the error active state from the bus off state before the CHMDC[1:0] bits are set to B'10, a bus off recovery interrupt request is generated.
  If the CPU requests transition to channel reset mode at the same time when the CAN module transitions to channel halt mode (at bus off entry when the BOM[1:0] bits are B'01 or at bus off end when the BOM[1:0] bits are B'10), the CPU's request takes precedence. Modify these bits only in channel reset mode.

- TAIE Bit
  When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

### 18.3.5 CANi Status Register L (CiSTSL) (i = 0)

Address C0STSLL: F0308H, C0STSLH: F0309H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | COMSTS | RECSTS | TRMSTS | BOSTS | EPSTS | CSLPSTS | CHLTSTS | CRSTSTS |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | — | Reserved | These bits are always read as 0. | R |
| 7 | COMSTS | Communication Status Flag | 0: Communication is not ready.<br>1: Communication is ready. | R |
| 6 | RECSTS | Receive Status Flag | 0: Bus idle, in transmission or bus off state<br>1: In reception | R |
| 5 | TRMSTS | Transmit Status Flag | 0: Bus idle or in reception<br>1: In transmission or bus off state | R |
| 4 | BOSTS | Bus Off Status Flag | 0: Not in bus off state<br>1: In bus off state | R |
| 3 | EPSTS | Error Passive Status Flag | 0: Not in error passive state<br>1: In error passive state | R |
| 2 | CSLPSTS | Channel Stop Status Flag | 0: Not in channel stop mode<br>1: In channel stop mode | R |
| 1 | CHLTSTS | Channel Halt Status Flag | 0: Not in channel halt mode<br>1: In channel halt mode | R |
| 0 | CRSTSTS | Channel Reset Status Flag | 0: Not in channel reset mode<br>1: In channel reset mode | R |

- COMSTS Flag
  This bit indicates that communication is ready.
  This flag becomes 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag is cleared to 0 in channel reset mode or channel halt mode.

- RECSTS Flag
  This flag is set to 1 when reception has started, and is cleared to 0 when the bus has become idle or transmission has started.

- TRMSTS Flag
  This flag is set to 1 when transmission has started, and is cleared to 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

- BOSTS Flag
  This flag is set to 1 when the CAN module has entered the bus off state (TEC[7:0] value > 255), and is cleared to 0 when the CAN module has exited the bus off state.

- EPSTS Flag
  This flag is set to 1 when the CAN module has entered the error passive state (128 ≤ TEC[7:0] value ≤ 255 or 128 ≤ REC[7:0] value), and is cleared to 0 when the CAN module has exited the error passive state or has entered channel reset mode.

- CSLPSTS Flag
  This flag is set to 1 when the CAN module has transitioned to channel stop mode, and is cleared to 0 when the CAN module has returned from channel stop mode.

- CHLTSTS Flag
  This flag is set to 1 when the CAN module has transitioned to channel halt mode, and is cleared to 0 when the CAN module has exited channel halt mode.

- CRSTSTS Flag
  This flag is set to 1 when the CAN module has transitioned to channel reset mode, and is cleared to 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 even if the CAN module transitions from channel reset mode to channel stop mode.

### 18.3.6 CANi Status Register H (CiSTSH) (i = 0)

Address C0STSHL: F030AH, C0STSHH: F030BH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TEC[7:0] | | | | | | | | REC[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description | Counter Value | R/W |
|---|---|---|---|---|
| 15 to 8 | TEC[7:0] | The transmit error counter (TEC) can be read. | — | R |
| 7 to 0 | REC[7:0] | The receive error counter (REC) can be read. | — | R |

- TEC[7:0] Bits
  These bits indicate the transmit error counter value. For transmit error counter increment/decrement conditions, see the CAN specifications (ISO11898-1).
  These bits are cleared to 0 in channel reset mode.

- REC[7:0] Bits
  These bits indicate the receive error counter value. For receive error counter increment/decrement conditions, see the CAN specifications (ISO11898-1).
  These bits are cleared to 0 in channel reset mode.

### 18.3.7 CANi Error Flag Register L (CiERFLL) (i = 0)

Address C0ERFLLL: F030CH, C0ERFLLH: F030DH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | ADERR | B0ERR | B1ERR | CERR | AERR | FERR | SERR | ALF | BLF | OVLF | BORF | BOEF | EPF | EWF | BEF |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R |
| 14 | ADERR | ACK Delimiter Error Flag | 0: No ACK delimiter error is detected.<br>1: ACK delimiter error is detected. | R/(W) Note |
| 13 | B0ERR | Dominant Bit Error Flag | 0: No dominant bit error is detected.<br>1: Dominant bit error is detected. | R/(W) Note |
| 12 | B1ERR | Recessive Bit Error Flag | 0: No recessive bit error is detected.<br>1: Recessive bit error is detected. | R/(W) Note |
| 11 | CERR | CRC Error Flag | 0: No CRC error is detected.<br>1: CRC error is detected. | R/(W) Note |
| 10 | AERR | ACK Error Flag | 0: No ACK error is detected.<br>1: ACK error is detected. | R/(W) Note |
| 9 | FERR | Form Error Flag | 0: No form error is detected.<br>1: Form error is detected. | R/(W) Note |
| 8 | SERR | Stuff Error Flag | 0: No stuff error is detected.<br>1: Stuff error is detected. | R/(W) Note |
| 7 | ALF | Arbitration Lost Flag | 0: No arbitration lost is detected.<br>1: Arbitration lost is detected. | R/(W) Note |
| 6 | BLF | Bus Lock Flag | 0: No channel bus lock is detected.<br>1: Channel bus lock is detected. | R/(W) Note |
| 5 | OVLF | Overload Flag | 0: No overload is detected.<br>1: Overload is detected. | R/(W) Note |
| 4 | BORF | Bus Off Recovery Flag | 0: No bus off recovery is detected.<br>1: Bus off recovery is detected. | R/(W) Note |
| 3 | BOEF | Bus Off Entry Flag | 0: No bus off entry is detected.<br>1: Bus off entry is detected. | R/(W) Note |
| 2 | EPF | Error Passive Flag | 0: No error passive is detected.<br>1: Error passive is detected. | R/(W) Note |
| 1 | EWF | Error Warning Flag | 0: No error warning is detected.<br>1: Error warning is detected. | R/(W) Note |
| 0 | BEF | Bus Error Flag | 0: No channel bus error is detected.<br>1: Channel bus error is detected. | R/(W) Note |

**Note** The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state. To write 0 to this flag bit, write by using an 8-bit data transfer instruction or a 16-bit data transfer instruction.

<R>

See the CAN specifications (ISO11898-1) if you want to check error occurrence conditions. To clear each flag of this register, write 0 by the program. These flags cannot be set to 1 by the program. If any of these flags is set to 1 at the timing when the program writes 0 to the flag, the flag is set to 1. Each flag is cleared to 0 in channel reset mode.

With respect to bits 14 to 8 in the CiERFLL register, if an error is detected with all flags of bits 14 to 8 set to 0 when the ERRD bit in the CiCTRH register is set to 0 (only the first error information is displayed), the corresponding flag is set to 1.

- ADERR Flag
  This flag is set to 1 when a form error has been detected in the ACK delimiter during transmission.

- B0ERR Flag
  This flag is set to 1 when a recessive bit has been detected though a dominant bit was transmitted.

- B1ERR Flag
  This flag is set to 1 when a dominant bit has been detected though a recessive bit was transmitted.

- CERR Flag
  This flag is set to 1 when a CRC error has been detected.

- AERR Flag
  This flag is set to 1 when an ACK error has been detected.

- FERR Flag
  This flag is set to 1 when a form error has been detected.

- SERR Flag
  This flag is set to 1 when a stuff error has been detected.

- ALF Flag
  This flag is set to 1 when an arbitration lost has been detected.

- BLF Flag
  This flag is set to 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, detection of the bus lock becomes possible again if either of the following conditions is met.
  - A recessive bit is detected after the BLF bit has been modified from 1 to 0.
  - The CAN module transitions to channel reset mode and returns to channel communication mode after the BLF bit has been modified from 1 to 0.

- OVLF Flag
  This flag is set to 1 when the overload frame transmit condition has been detected when performing reception or transmission.

- BORF Flag
  This flag is set to 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus off state. However, this flag is not set to 1 if the CAN module returns from the bus off state in either of the following ways before 11 consecutive recessive bits are detected 128 times.
  - The CHMDC[1:0] bits in the CiCTRL register are set to B'01 (channel reset mode).
  - The RTBO bit in the CiCTRL register is set to 1 (forcible return from the bus off state is made).
  - The BOM[1:0] bits in the CiCTRH register are set to B'01 (transition to channel halt mode at bus off entry).
  - The CHMDC[1:0] bits in the CiCTRL register are set to B'10 (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the BOM[1:0] bits set to B'11 (transition to channel halt mode upon a request from the program during bus off).

- BOEF Flag
  This flag is set to 1 when the state becomes bus off state (TEC[7:0] value > 255). This flag is also set to 1 when the state becomes bus off state with the BOM[1:0] bits in the CiCTRH register set to B'01 (transition to channel halt mode at bus off entry).

- EPF Flag
  This flag becomes 1 when the CAN module becomes error passive state (REC[7:0] or TEC[7:0] value > 127). This flag becomes 1 only when the REC[7:0] or TEC[7:0] value exceeds 127 for the first time. Therefore, if the program writes 0 to this flag with the REC[7:0] or TEC[7:0] value remaining over 127, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 127 or less and then the REC[7:0] or TEC[7:0] value exceeds 127 again.

- EWF Flag
  This flag is set to 1 only when the REC[7:0] or TEC[7:0] value exceeds 95 for the first time. Therefore, if the program writes 0 to this flag with the REC[7:0] or TEC[7:0] value remaining over 95, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 95 or less and then the REC[7:0] or TEC[7:0] value exceeds 95 again.

- BEF Flag
  This flag is set to 1 when any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags in the CiERFLL register is set to 1.

### 18.3.8 CANi Error Flag Register H (CiERFLH) (i = 0)

Address C0ERFLHL: F030EH, C0ERFLHH: F030FH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | CRCREG[14:0] | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R |
| 14 to 0 | CRCREG [14:0] | CRC Calculation Data | A CRC value calculated based on the transmit message or receive message is indicated. | R |

- CRCREG[14:0] Bits
  When the CTME bit in the CiCTRH register is set to 1 (communication test mode is enabled), the CRC value calculated based on the transmit or receive message can be read. When the CTME bit is set to 0 (communication test mode is disabled), these bits are always read as 0.

### 18.3.9 CAN Global Configuration Register L (GCFGL)

Address GCFGLL: F0322H, GCFGLH: F0323H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | TSSS | TSP[3:0] | | | | — | — | — | DCS | MME | DRE | DCE | TPRI |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 13 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 12 | TSSS | Timestamp Clock Source Select | 0: Clock obtained by frequency-dividing $f_{CLK}$ by 2 ($f_{CLK}/2$)<br>1: CANi bit time clock | R/W |
| 11 to 8 | TSP[3:0] | Timestamp Clock Source Division | b11 b10 b9 b8<br>0 0 0 0 : Not divided<br>0 0 0 1 : Divided by 2<br>0 0 1 0 : Divided by 4<br>0 0 1 1 : Divided by 8<br>0 1 0 0 : Divided by 16<br>0 1 0 1 : Divided by 32<br>0 1 1 0 : Divided by 64<br>0 1 1 1 : Divided by 128<br>1 0 0 0 : Divided by 256<br>1 0 0 1 : Divided by 512<br>1 0 1 0 : Divided by 1024<br>1 0 1 1 : Divided by 2048<br>1 1 0 0 : Divided by 4096<br>1 1 0 1 : Divided by 8192<br>1 1 1 0 : Divided by 16384<br>1 1 1 1 : Divided by 32768 | R/W |
| 7 to 5 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 4 | DCS | CAN Clock Source Select | 0: Clock obtained by frequency-dividing $f_{CLK}$ by 2 ($f_{CLK}/2$)<br>1: X1 clock (fx) | R/W |
| 3 | MME | Mirror Function Enable | 0: Mirror function is disabled.<br>1: Mirror function is enabled. | R/W |
| 2 | DRE | DLC Replacement Enable | 0: DLC replacement is disabled.<br>1: DLC replacement is enabled. | R/W |
| 1 | DCE | DLC Check Enable | 0: DLC check is disabled.<br>1: DLC check is enabled. | R/W |
| 0 | TPRI | Transmit Priority Select | 0: ID priority<br>1: Transmit buffer number priority | R/W |

Modify the GCFGL register only in global reset mode.

- TSSS Bit
  This bit is used to select a clock source of the timestamp counter.

- TSP[3:0] Bits
  The clock obtained by dividing the clock source (selected by the TSSS bit) by the TSP[3:0] value is the count source of the timestamp counter.

- DCS Bit
  When this bit is set to 0, the clock obtained by frequency-dividing $f_{CLK}$ by 2 ($f_{CLK}/2$) is used as the clock source of the CAN clock ($f_{CAN}$).
  When this bit is set to 1, the X1 clock (fx) is used as the clock source of the CAN clock. If the X1 clock (fx) is selected, make the X1 clock (fx) into the value below half[Note1, 2] of $f_{CLK}$.

  Notes 1. When the $f_{CLK}$ clock source is high-speed on-chip oscillator or PLL clock sourced high-speed on-chip oscillator, the frequency must be set as "$f_X < f_{CLK}/2$".
  2. If the high-speed system clock is to be selected as $f_{CLK}$, do not select $f_X$ as $f_{CAN}$.

- MME Bit
  Setting this bit to 1 makes the mirror function available.

- DRE Bit
  When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of H'00 is stored in the data byte that exceeds the DLC value of the receive rule.
  When the DCE bit is set to 1 (DLC check is enabled), the DLC replacement function is available.

- DCE Bit
  Setting this bit to 1 makes the DLC check function available. Set the GAFLDLC[3:0] bits in the GAFLPHj register to B'0000 before clearing the DCE bit in the GCFGL register to 0.

- TPRI Bit
  This bit is used to set the transmit priority.
  When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO11898-1 specifications). When this bit is set to 1, transmit buffer number priority is selected and the minimum number of transmit buffer specified for transmission takes precedence.

### 18.3.10 CAN Global Configuration Register H (GCFGH)

Address GCFGHL: F0324H, GCFGHH: F0325H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ITRCP[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15-0 | ITRCP[15:0] | Interval Timer Prescaler Set | If the set value is M, $f_{CLK}$/2 is frequency-divided by M. Setting H'0000 is prohibited, when the interval timer is in use. | R/W |

Modify the GCFGH register only in global reset mode.

- ITRCP[15:0] Bits
  These bits are used to set a clock source division value of the interval timer for FIFO buffers. For details, see **18.6.3 (1) Interval Transmission Function**.

### 18.3.11 CAN Global Control Register L (GCTRL)

Address GCTRLL: F0326H, GCTRLH: F0327H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | THLEIE | MEIE | DEIE | — | — | — | — | — | GSLPR | GMDC[1:0] | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 11 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 10 | THLEIE | Transmit History Buffer Overflow Interrupt Enable | 0: Transmit history buffer overflow interrupt is disabled.<br>1: Transmit history buffer overflow interrupt is enabled. | R/W |
| 9 | MEIE | FIFO Message Lost Interrupt Enable | 0: FIFO message lost interrupt is disabled.<br>1: FIFO message lost interrupt is enabled. | R/W |
| 8 | DEIE | DLC Error Interrupt Enable | 0: DLC error interrupt is disabled.<br>1: DLC error interrupt is enabled. | R/W |
| 7 to 3 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 2 | GSLPR | Global Stop Mode | 0: Other than global stop mode<br>1: Global stop mode | R/W |
| 1, 0 | GMDC[1:0] | Global Mode Select | b1 b0<br>0 0 : Global operating mode<br>0 1 : Global reset mode<br>1 0 : Global test mode<br>1 1 : Setting prohibited | R/W |

- THLEIE Bit
  When the THLEIE bit is set to 1 and the THLES flag in the GERFLL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

- MEIE Bit
  When the MEIE bit is set to 1 and the MES flag in the GERFLL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

- DEIE Bit
  When the DEIE bit is set to 1 and the DEF flag in the GERFLL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

- GSLPR Bit
  Setting this bit to 1 places the CAN module in global stop mode.
  Clearing this bit to 0 makes the CAN module leave from global stop mode.
  Do not modify this bit while the CAN channel is in global operating mode or global test mode.

- GMDC[1:0] Bits
  These bits are used to select the mode of entire CAN module (global operating mode, global reset mode, or global test mode). For details, see **18.4.1 Global Modes**. Setting the GSLPR bit to 1 in global reset mode places the CAN module in global stop mode.

### 18.3.12 CAN Global Control Register H (GCTRH)

Address GCTRHL: F0328H, GCTRHH: F0329H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TSRST |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 1 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 0 | TSRST | Timestamp Counter Reset | Setting the TSRST bit to 1 resets the timestamp counter. This bit is always read as 0. | R/W |

- TSRST Bit
  This bit is used to reset the timestamp counter. When this bit is set to 1, the GTSC register is cleared to H'0000.

### 18.3.13 CAN Global Status Register (GSTS)

Address GSTSL: F032AH, GSTSH: F032BH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | GRAMINIT | GSLPSTS | GHLTSTS | GRSTSTS |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 4 | — | Reserved | These bits are always read as 0. | R |
| 3 | GRAMINIT | CAN RAM Initialization Status Flag | 0: CAN RAM initialization is completed.<br>1: CAN RAM initialization is ongoing. | R |
| 2 | GSLPSTS | Global Stop Status Flag | 0: Not in global stop mode<br>1: In global stop mode | R |
| 1 | GHLTSTS | Global Test Status Flag | 0: Not in global test mode<br>1: In global test mode | R |
| 0 | GRSTSTS | Global Reset Status Flag | 0: Not in global reset mode<br>1: In global reset mode | R |

- GRAMINIT Flag
  This flag indicates the initialization status of the CAN RAM.
  This flag is set to 1 after the CAN module is enabled, and is cleared to 0 when CAN RAM initialization is completed.

- GSLPSTS Flag
  This flag is set to 1 when the CAN module has transitioned to global stop mode, and is cleared to 0 when the CAN module has returned from global stop mode.

- GHLTSTS Flag
  This flag is set to 1 when the CAN module has transitioned to global test mode, and is cleared to 0 when the CAN module has exited global test mode.

- GRSTSTS Flag
  This flag is set to 1 when the CAN module has transitioned to global reset mode, and is cleared to 0 when the CAN module has exited global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

### 18.3.14 CAN Global Error Flag Register (GERFLL)

Address GERFLL: F032CH

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | THLES | MES | DEF |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 3 | — | Reserved | The read value is undefined. The write value should always be 0. | R |
| 2 | THLES | Transmit History Buffer Overflow Status Flag | 0: No transmit history buffer overflow is present.<br>1: A transmit history buffer overflow is present. | R |
| 1 | MES | FIFO Message Lost Status Flag | 0: No FIFO message lost error is present.<br>1: A FIFO message lost error is present. | R |
| 0 | DEF | DLC Error Flag | 0: No DLC error is present.<br>1: A DLC error is present. | R/(W) Note |

<R> **Note** The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state. To write 0 to this flag bit, write by using an 8-bit data transfer instruction.

All flags in the GERFLL register are cleared to 0 in global reset mode.

- THLES Flag
  The THLES flag is set to 1 when the THLELT flag in the THLSTSi register is set to 1.
  This flag is cleared to 0 when the THLELT flag is set to 0.

- MES Flag
  The MES flag is set to 1 when any one of the RFMLT flags in the RFSTSm register (m = 0, 1) or the CFMLT flag in the CFSTSk register is set to 1.
  This flag is cleared to 0 when all RFMLT flags and the CFMLT flag are set to 0.

- DEF Flag
  The DEF flag is set to 1 when an error has been detected during the DLC check. This flag can be cleared to 0 by writing 0 by the program.

### 18.3.15 CAN Global Transmit Interrupt Status Register (GTINTSTS)

Address GTINTSTSL: F0388H, GTINTSTSH: F0389H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | THIF0 | CFTIF0 | TAIF0 | TSIF0 |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 4 | — | Reserved | The read value is undefined. The write value should always be 0. | R |
| 3 | THIF0 | CANi Transmit History Interrupt Status Flag | 0: No transmit history interrupt request is present.<br>1: A transmit history interrupt request is present. | R |
| 2 | CFTIF0 | CANi Transmit/Receive FIFO Interrupt Status Flag | 0: No transmit/receive FIFO transmit interrupt request is present.<br>1: A transmit/receive FIFO transmit interrupt request is present. | R |
| 1 | TAIF0 | CANi Transmit Buffer Abort Interrupt Status Flag | 0: No transmit buffer abort interrupt request is present.<br>1: A transmit buffer abort interrupt request is present. | R |
| 0 | TSIF0 | CANi Transmit Buffer Interrupt Status Flag | 0: No transmit buffer transmit complete interrupt request is present.<br>1: A transmit buffer transmit complete interrupt request is present. | R |

All flags in the GTINTSTS register are cleared to 0 in global reset or channel reset mode.

- THIF0 Flag
  The THIF0 flag is set to 1 when the THLIE bit in the THLCCi register is set to 1 (enabling interrupts) and the THLIF flag in the THLSTSi register is set to 1 (interrupt request present).
  This flag is cleared to 0 when the THLIF flag is set to 0. This flag is also cleared to 0 when the THLIE bit is set to 0.

- CFTIF0 Flag
  The CFTIF0 flag is set to 1 when the CFTXIE bit in the CFCCLk register is set to 1 (enabling interrupts) and the CFTXIF flag in the CFSTSk register is set to 1 (interrupt request present).
  This flag is cleared to 0 when the CFTXIF flag is set to 0. This flag is also cleared to 0 when the CFTXIE bit is set to 0.

- TAIF0 Flag
  The TAIF0 flag is set to 1 when the TAIE bit in the CiCTRH register is set to 1 (enabling interrupts) and the TMTRF[1:0] flag in the TMSTSp register is set to B'01 (transmit abort has been completed).
  This flag is cleared to 0 when the TMTRF[1:0] flag, which indicates that the abortion of transmission has been completed, is set to B'00.

- TSIF0 Flag
  The TSIF0 flag is set to 1 when the TMIEp bit in the TMIEC register is set to 1 (enabling interrupts) and the TMTRF[1:0] flag in the corresponding TMSTSp register is set to B’10 (transmission has been completed (without transmit abort request) ) or B’11 (transmission has been completed (with transmit abort request)).
  This flag is cleared to 0 when all TMTRF[1:0] flags that satisfy a condition for setting the TSIF0 flag to 1 are set to B’00. This flag is also cleared to 0 when the TMIEp bit is set to 0.

### 18.3.16 CAN Timestamp Register (GTSC)

Address GTSC: F032EH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TS[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description | Counter Value | R/W |
|---|---|---|---|---|
| 15 to 0 | TS[15:0] | The timestamp counter value can be read. | H'0000 to H'FFFF | R |

- TS[15:0] Bits
  When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. The TS[15:0] value is captured when the SOF is detected and then stored in the receive buffer or the FIFO buffer. The timestamp counter is initialized in global reset mode.
  The timestamp counter start timing and stop timing depend on the count source.
  - When the TSSS value in the GCFGL register is 0 (the clock obtained by frequency-dividing $f_{CLK}$ by 2 ($f_{CLK}/2$) is selected):
    The timestamp counter starts counting when the CAN module has transitioned to global operating mode.
    This counter stops counting when the CAN module has transitioned to global stop mode or global test mode.
  - When the TSSS value in the GCFGL register is 1 (CANi bit time clock is selected):
    The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.
    This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

### 18.3.17 CAN Receive Rule Number Configuration Register (GAFLCFG)

Address GAFLCFGL: F0330H, GAFLCFGH: F0331H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | RNC0[4:0] | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 5 | — | Reserved | These bits are always read as 0. | R |
| 4 to 0 | RNC0[4:0] | CANi Receive Rule Number Set | Set the number of receive rules of channel 0.<br>Set these bits to a value within a range of H'00 to H'10. | R/W |

Modify the GAFLCFG register only in global reset mode.
Up to 16 rules can be registered in the receive rule table.

- RNC0[4:0] Bits
  These bits are used to set the number of rules to be registered in the channel 0 receive rule table.
  Set these bits to a value within a range of H'00 to H'10.

### 18.3.18 CAN Receive Rule Entry Register jAL (GAFLIDLj) (j = 0 to 15)

Address GAFLIDL0L: F03A0H, GAFLIDL0H: F03A1H GAFLIDL1L: F03ACH, GAFLIDL1H: F03ADH
GAFLIDL2L: F03B8H, GAFLIDL2H: F03B9H GAFLIDL3L: F03C4H, GAFLIDL3H: F03C5H
GAFLIDL4L: F03D0H, GAFLIDL4H: F03D1H GAFLIDL5L: F03DCH, GAFLIDL5H: F03DDH
GAFLIDL6L: F03E8H, GAFLIDL6H: F03E9H GAFLIDL7L: F03F4H, GAFLIDL7H: F03F5H
GAFLIDL8L: F0400H, GAFLIDL8H: F0401H GAFLIDL9L: F040CH, GAFLIDL9H: F040DH
GAFLIDL10L: F0418H, GAFLIDL10H: F0419H GAFLIDL11L: F0424H, GAFLIDL11H: F0425H
GAFLIDL12L: F0430H, GAFLIDL12H: F0431H GAFLIDL13L: F043CH, GAFLIDL13H: F043DH
GAFLIDL14L: F0448H, GAFLIDL14H: F0449H GAFLIDL15L: F0454H, GAFLIDL15H: F0455H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GAFLID[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 0 | GAFLID[15:0] | ID Set L | Set the ID of the receive rule.<br>For the standard ID, set the ID in bits 10 to 0 and set bits 15 to 11 to 0. | R/W |

Modify the GAFLIDLj register only when the RPAGE bit in the GRWCR register is set to 0 in global reset mode.

- GAFLID[15:0] Bits
  These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID in the received message during the acceptance filter processing.

### 18.3.19 CAN Receive Rule Entry Register jAH (GAFLIDHj) (j = 0 to 15)

Address GAFLIDH0L: F03A2H, GAFLIDH0H: F03A3H GAFLIDH1L: F03AEH, GAFLIDH1H: F03AFH
GAFLIDH2L: F03BAH, GAFLIDH2H: F03BBH GAFLIDH3L: F03C6H, GAFLIDH3H: F03C7H
GAFLIDH4L: F03D2H, GAFLIDH4H: F03D3H GAFLIDH5L: F03DEH, GAFLIDH5H: F03DFH
GAFLIDH6L: F03EAH, GAFLIDH6H: F03EBH GAFLIDH7L: F03F6H, GAFLIDH7H: F03F7H
GAFLIDH8L: F0402H, GAFLIDH8H: F0403H GAFLIDH9L: F040EH, GAFLIDH9H: F040FH
GAFLIDH10L: F041AH, GAFLIDH10H: F041BH GAFLIDH11L: F0426H, GAFLIDH11H: F0427H
GAFLIDH12L: F0432H, GAFLIDH12H: F0433H GAFLIDH13L: F043EH, GAFLIDH13H: F043FH
GAFLIDH14L: F044AH, GAFLIDH14H: F044BH GAFLIDH15L: F0456H, GAFLIDH15H: F0457H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GAFLIDE | GAFLRTR | GAFLLB | GAFLID[28:16] | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 | GAFLIDE | IDE Select | 0: Standard ID<br>1: Extended ID | R/W |
| 14 | GAFLRTR | RTR Select | 0: Data frame<br>1: Remote frame | R/W |
| 13 | GAFLLB | Receive Rule Target Message Select | 0: When a message transmitted from another CAN node is received<br>1: When a message transmitted from own node is received | R/W |
| 12 to 0 | GAFLID[28:16] | ID Set H | Set the ID of the receive rule.<br>For the standard ID, set these bits to 0. | R/W |

Modify the GAFLIDHj register only when the RPAGE bit in the GRWCR register is set to 0 in global reset mode.

- GAFLIDE Bit
  This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

- GAFLRTR Bit
  This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

- GAFLLB Bit
  When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.
  When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when receiving messages transmitted from the own CAN node.

- GAFLID[28:16] Bits
  These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID in the received message during the acceptance filter processing.

### 18.3.20 CAN Receive Rule Entry Register jBL (GAFLMLj) (j = 0 to 15)

Address GAFLML0L: F03A4H, GAFLML0H: F03A5H GAFLML1L: F03B0H, GAFLML1H: F03B1H
GAFLML2L: F03BCH, GAFLML2H: F03BDH GAFLML3L: F03C8H, GAFLML3H: F03C9H
GAFLML4L: F03D4H, GAFLML4H: F03D5H GAFLML5L: F03E0H, GAFLML5H: F03E1H
GAFLML6L: F03ECH, GAFLML6H: F03EDH GAFLML7L: F03F8H, GAFLML7H: F03F9H
GAFLML8L: F0404H, GAFLML8H: F0405H GAFLML9L: F0410H, GAFLML9H: F0411H
GAFLML10L: F041CH, GAFLML10H: F041DH GAFLML11L: F0428H, GAFLML11H: F0429H
GAFLML12L: F0434H, GAFLML12H: F0435H GAFLML13L: F0440H, GAFLML13H: F0441H
GAFLML14L: F044CH, GAFLML14H: F044DH GAFLML15L: F0458H, GAFLML15H: F0459H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GAFLIDM[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 0 | GAFLIDM [15:0] | ID Mask L | 0: The corresponding ID bit is not compared.<br>1: The corresponding ID bit is compared. | R/W |

Modify the GAFLMLj register only when the RPAGE bit in the GRWCR register is set to 0 in global reset mode.

- GAFLIDM[15:0] Bits
  These bits are used to mask the corresponding ID bit of the receive rule.

### 18.3.21 CAN Receive Rule Entry Register jBH (GAFLMHj) (j = 0 to 15)

Address GAFLMH0L: F03A6H, GAFLMH0H: F03A7H GAFLMH1L: F03B2H, GAFLMH1H: F03B3H
GAFLMH2L: F03BEH, GAFLMH2H: F03BFH GAFLMH3L: F03CAH, GAFLMH3H: F03CBH
GAFLMH4L: F03D6H, GAFLMH4H: F03D7H GAFLMH5L: F03E2H, GAFLMH5H: F03E3H
GAFLMH6L: F03EEH, GAFLMH6H: F03EFH GAFLMH7L: F03FAH, GAFLMH7H: F03FBH
GAFLMH8L: F0406H, GAFLMH8H: F0407H GAFLMH9L: F0412H, GAFLMH9H: F0413H
GAFLMH10L: F041EH, GAFLMH10H: F041FH GAFLMH11L: F042AH, GAFLMH11H: F042BH
GAFLMH12L: F0436H, GAFLMH12H: F0437H GAFLMH13L: F0442H, GAFLMH13H: F0443H
GAFLMH14L: F044EH, GAFLMH14H: F044FH GAFLMH15L: F045AH, GAFLMH15H: F045BH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GAFLIDEM | GAFLRTRM | — | GAFLIDM[28:16] | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 | GAFLIDEM | IDE Mask | 0: The IDE bit is not compared.<br>1: The IDE bit is compared. | R/W |
| 14 | GAFLRTRM | RTR Mask | 0: The RTR bit is not compared.<br>1: The RTR bit is compared | R/W |
| 13 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R |
| 12 to 0 | GAFLIDM [28:16] | ID Mask H | 0: The corresponding ID bit is not compared.<br>1: The corresponding ID bit is compared. | R/W |

Modify the GAFLMHj register only when the RPAGE bit in the GRWCR register is set to 0 in global reset mode.

- GAFLIDEM Bit
  When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDE bit in the GAFLIDHj register.
  When this bit is set to 0, it is regarded that all received messages have matched the specified ID format. To set the GAFLIDEM bit to 0, set the GAFLIDM[28:16] bits in the GAFLMHj register and the GAFLIDM[15:0] bits in the GAFLMLj register to all 0 at the same time.

- GAFLRTRM Bit
  This bit is used to mask the RTR bit of the receive rule.

- GAFLIDM[28:16] Bits
  These bits are used to mask the corresponding ID bit of the receive rule.

### 18.3.22 CAN Receive Rule Entry Register jCL (GAFLPLj) (j = 0 to 15)

Address GAFLPL0L: F03A8H, GAFLPL0H: F03A9H GAFLPL1L: F03B4H, GAFLPL1H: F03B5H
GAFLPL2L: F03C0H, GAFLPL2H: F03C1H GAFLPL3L: F03CCH, GAFLPL3H: F03CDH
GAFLPL4L: F03D8H, GAFLPL4H: F03D9H GAFLPL5L: F03E4H, GAFLPL5H: F03E5H
GAFLPL6L: F03F0H, GAFLPL6H: F03F1H GAFLPL7L: F03FCH, GAFLPL7H: F03FDH
GAFLPL8L: F0408H, GAFLPL8H: F0409H GAFLPL9L: F0414H, GAFLPL9H: F0415H
GAFLPL10L: F0420H, GAFLPL10H: F0421H GAFLPL11L: F042CH, GAFLPL11H: F042DH
GAFLPL12L: F0438H, GAFLPL12H: F0439H GAFLPL13L: F0444H, GAFLPL13H: F0445H
GAFLPL14L: F0450H, GAFLPL14H: F0451H GAFLPL15L: F045CH, GAFLPL15H: F045DH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GAFL RMV | GAFLRMDP[6:0] | | | | | | | — | — | — | GAFL FDP4 | — | — | GAFL FDP1 | GAFL FDP0 |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 | GAFLRMV | Receive Buffer Enable | 0: No receive buffer is used.<br>1: A receive buffer is used. | R/W |
| 14 to 8 | GAFLRMDP [6:0] | Receive Buffer Number Select | Set the receive buffer number to store receive messages. | R/W |
| 7 to 5 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 4 | GAFLFDP4 | CAN0 Transmit/Receive FIFO Buffer Select 0 | 0: Not select a CAN0 transmit/receive FIFO buffer 0<br>1: Select a CAN0 transmit/receive FIFO buffer 0 | R/W |
| 3, 2 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 1 | GAFLFDP1 | Receive FIFO Buffer Select 1 | 0: Not select a receive FIFO buffer 1<br>1: Select a receive FIFO buffer 1 | R/W |
| 0 | GAFLFDP0 | Receive FIFO Buffer Select 0 | 0: Not select a receive FIFO buffer 0<br>1: Select a receive FIFO buffer 0 | R/W |

Modify the GAFLPLj register only when the RPAGE bit in the GRWCR register is set to 0 in global reset mode.

- GAFLRMV Bit
  When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

- GAFLRMDP[6:0] Bits
  These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the NRXMB[4:0] bits in the RMNB register.

- GAFLFDP4, GAFLFDP1, and GAFLFDP0 Bits
  These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to two FIFO buffers are selectable. However, when the GAFLRMV bit in the GAFLPLj register is set to 1 (a receive buffer is used), up to one FIFO buffer is selectable. Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFM[1:0] bits in the CFCCHk register are set to B'00 (receive mode) are selectable.

### 18.3.23 CAN Receive Rule Entry Register jCH (GAFLPHj) (j = 0 to 15)

Address GAFLPH0L: F03AAH, GAFLPH0H: F03ABH GAFLPH1L: F03B6H, GAFLPH1H: F03B7H
GAFLPH2L: F03C2H, GAFLPH2H: F03C3H GAFLPH3L: F03CEH, GAFLPH3H: F03CFH
GAFLPH4L: F03DAH, GAFLPH4H: F03DBH GAFLPH5L: F03E6H, GAFLPH5H: F03E7H
GAFLPH6L: F03F2H, GAFLPH6H: F03F3H GAFLPH7L: F03FEH, GAFLPH7H: F03FFH
GAFLPH8L: F040AH, GAFLPH8H: F040BH GAFLPH9L: F0416H, GAFLPH9H: F0417H
GAFLPH10L: F0422H, GAFLPH10H: F0423H GAFLPH11L: F042EH, GAFLPH11H: F042FH
GAFLPH12L: F043AH, GAFLPH12H: F043BH GAFLPH13L: F0446H, GAFLPH13H: F0447H
GAFLPH14L: F0452H, GAFLPH14H: F0453H GAFLPH15L: F045EH, GAFLPH15H: F045FH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | GAFLDLC[3:0] | | | | GAFLPTR[11:0] | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 12 | GAFLDLC [3:0] | Receive Rule DLC | b15 b14 b13 b12<br>0 0 0 0 : DLC check is disabled.<br>0 0 0 1 : 1 data byte<br>0 0 1 0 : 2 data bytes<br>0 0 1 1 : 3 data bytes<br>0 1 0 0 : 4 data bytes<br>0 1 0 1 : 5 data bytes<br>0 1 1 0 : 6 data bytes<br>0 1 1 1 : 7 data bytes<br>1 X X X : 8 data bytes | R/W |
| 11 to 0 | GAFLPTR [11:0] | Receive Rule Label | Set the 12-bit label information. | R/W |

Modify the GAFLPHj register only when the RPAGE bit in the GRWCR register is set to 0 in global reset mode.

- GAFLDLC[3:0] Bits
  These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to B'0000 disables the DLC check function allowing messages with any data length to pass the DLC check.

- GAFLPTR [11:0] Bits
  These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

### 18.3.24 CAN Receive Buffer Number Configuration Register (RMNB)

Address RMNBL: F0332H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | NRXMB[4:0] | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 5 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 4 to 0 | NRXMB[4:0] | Receive Buffer Number Configuration | Set the number of receive buffers.<br>Set a value of 0 to 16. | R/W |

Modify the RMNB register only in global reset mode.

- NRXMB[4:0] Bits
  These bits are used to set the total number of receive buffers of the CAN module.
  The maximum value is 16.
  Setting these bits to all 0 makes receive buffers unavailable.

### 18.3.25 CAN Receive Buffer Receive Complete Flag Register (RMND0)

Address RMND0L: F0334H, RMND0H: F0335H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMNS[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 0 | RMNS[15:0] | Receive Buffer Receive Complete Flag n | 0: Receive buffer n contains no new message (n = 0 to 15).<br>1: Receive buffer n contains a new message. | R/W |

Write 0 to the RMND0 register in global operating mode or global test mode.

- RMNS[15:0] Flags
  Each RMNS flag is set to 1 when the processing for storing a message in the corresponding receive buffer starts.
  To clear these flags to 0, write 0 to the corresponding flag by the program. In this case, write 0 to bits to be cleared and write 1 to other bits by using an 8-bit or 16-bit data transfer instruction. These bits cannot be set to 0 while a message is being stored. It takes time of ten clock cycles of $f_{CLK}$ for storing a message.
  These flags are cleared to 0 in global reset mode.

### 18.3.26 CAN Receive Buffer Register nAL (RMIDLn) (n = 0 to 15)

Address RMIDL0L: F03A0H, RMIDL0H: F03A1H RMIDL1L: F03B0H, RMIDL1H: F03B1H
RMIDL2L: F03C0H, RMIDL2H: F03C1H RMIDL3L: F03D0H, RMIDL3H: F03D1H
RMIDL4L: F03E0H, RMIDL4H: F03E1H RMIDL5L: F03F0H, RMIDL5H: F03F1H
RMIDL6L: F0400H, RMIDL6H: F0401H RMIDL7L: F0410H, RMIDL7H: F0411H
RMIDL8L: F0420H, RMIDL8H: F0421H RMIDL9L: F0430H, RMIDL9H: F0431H
RMIDL10L: F0440H, RMIDL10H: F0441H RMIDL11L: F0450H, RMIDL11H: F0451H
RMIDL12L: F0460H, RMIDL12H: F0461H RMIDL13L: F0470H, RMIDL13H: F0471H
RMIDL14L: F0480H, RMIDL14H: F0481H RMIDL15L: F0490H, RMIDL15H: F0491H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMID[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 0 | RMID[15:0] | Receive Buffer ID Data L | The standard ID or extended ID of received message can be read.<br>Read bits 10 to 0 for standard ID. Bits 15 to 11 are read as 0. | R |

This register can be read when the RPAGE bit in the GRWCR register is 1.

- RMID[15:0] Bits
  These bits indicate the ID of the message stored in the receive buffer.

### 18.3.27 CAN Receive Buffer Register nAH (RMIDHn) (n = 0 to 15)

Address RMIDH0L: F03A2H, RMIDH0H: F03A3H RMIDH1L: F03B2H, RMIDH1H: F03B3H
RMIDH2L: F03C2H, RMIDH2H: F03C3H RMIDH3L: F03D2H, RMIDH3H: F03D3H
RMIDH4L: F03E2H, RMIDH4H: F03E3H RMIDH5L: F03F2H, RMIDH5H: F03F3H
RMIDH6L: F0402H, RMIDH6H: F0403H RMIDH7L: F0412H, RMIDH7H: F0413H
RMIDH8L: F0422H, RMIDH8H: F0423H RMIDH9L: F0432H, RMIDH9H: F0433H
RMIDH10L: F0442H, RMIDH10H: F0443H RMIDH11L: F0452H, RMIDH11H: F0453H
RMIDH12L: F0462H, RMIDH12H: F0463H RMIDH13L: F0472H, RMIDH13H: F0473H
RMIDH14L: F0482H, RMIDH14H: F0483H RMIDH15L: F0492H, RMIDH15H: F0493H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMIDE | RMRTR | — | RMID[28:16] | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 | RMIDE | Receive Buffer IDE | 0: Standard ID<br>1: Extended ID | R |
| 14 | RMRTR | Receive Buffer RTR | 0: Data frame<br>1: Remote frame | R |
| 13 | — | Reserved | This bit is always read as 0. | R |
| 12 to 0 | RMID[28:16] | Receive Buffer ID Data H | The standard ID or extended ID of received message can be read.<br>For standard ID, these bits are read as 0. | R |

This register can be read when the RPAGE bit in the GRWCR register is 1.

- RMIDE Bit
  This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

- RMRTR Bit
  This bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer.

- RMID[28:16] Bits
  These bits indicate the ID of the message stored in the receive buffer.

### 18.3.28 CAN Receive Buffer Register nBL (RMTSn) (n = 0 to 15)

Address RMTS0L: F03A4H, RMTS0H: F03A5H RMTS1L: F03B4H, RMTS1H: F03B5H
RMTS2L: F03C4H, RMTS2H: F03C5H RMTS3L: F03D4H, RMTS3H: F03D5H
RMTS4L: F03E4H, RMTS4H: F03E5H RMTS5L: F03F4H, RMTS5H: F03F5H
RMTS6L: F0404H, RMTS6H: F0405H RMTS7L: F0414H, RMTS7H: F0415H
RMTS8L: F0424H, RMTS8H: F0425H RMTS9L: F0434H, RMTS9H: F0435H
RMTS10L: F0444H, RMTS10H: F0445H RMTS11L: F0454H, RMTS11H: F0455H
RMTS12L: F0464H, RMTS12H: F0465H RMTS13L: F0474H, RMTS13H: F0475H
RMTS14L: F0484H, RMTS14H: F0485H RMTS15L: F0494H, RMTS15H: F0495H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMTS[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 0 | RMTS[15:0] | Receive Buffer Timestamp Data | Timestamp value of the received message can be read. | R |

This register can be read when the RPAGE bit in the GRWCR register is 1.

- RMTS[15:0] Bits
  These bits indicate the timestamp value of the message stored in the receive buffer.

### 18.3.29 CAN Receive Buffer Register nBH (RMPTRn) (n = 0 to 15)

Address RMPTR0L: F03A6H, RMPTR0H: F03A7H RMPTR1L: F03B6H, RMPTR1H: F03B7H
RMPTR2L: F03C6H, RMPTR2H: F03C7H RMPTR3L: F03D6H, RMPTR3H: F03D7H
RMPTR4L: F03E6H, RMPTR4H: F03E7H RMPTR5L: F03F6H, RMPTR5H: F03F7H
RMPTR6L: F0406H, RMPTR6H: F0407H RMPTR7L: F0416H, RMPTR7H: F0417H
RMPTR8L: F0426H, RMPTR8H: F0427H RMPTR9L: F0436H, RMPTR9H: F0437H
RMPTR10L: F0446H, RMPTR10H: F0447H RMPTR11L: F0456H, RMPTR11H: F0457H
RMPTR12L: F0466H, RMPTR12H: F0467H RMPTR13L: F0476H, RMPTR13H: F0477H
RMPTR14L: F0486H, RMPTR14H: F0487H RMPTR15L: F0496H, RMPTR15H: F0497H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMDLC[3:0] | | | | RMPTR[11:0] | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 12 | RMDLC[3:0] | Receive Buffer DLC Data | b15 b14 b13 b12<br>0 0 0 0 : 0 data bytes<br>0 0 0 1 : 1 data byte<br>0 0 1 0 : 2 data bytes<br>0 0 1 1 : 3 data bytes<br>0 1 0 0 : 4 data bytes<br>0 1 0 1 : 5 data bytes<br>0 1 1 0 : 6 data bytes<br>0 1 1 1 : 7 data bytes<br>1 X X X : 8 data bytes | R |
| 11 to 0 | RMPTR [11:0] | Receive Buffer Label Data | Label information of the received message can be read. | R |

This register can be read when the RPAGE bit in the GRWCR register is 1.

- RMDLC[3:0] Bits
  These bits indicate the data length of the message stored in the receive buffer.

- RMPTR[11:0] Bits
  These bits indicate the label information of the message stored in the receive buffer.

### 18.3.30 CAN Receive Buffer Register nCL (RMDF0n) (n = 0 to 15)

Address RMDF00L: F03A8H, RMDF00H: F03A9H RMDF01L: F03B8H, RMDF01H: F03B9H
RMDF02L: F03C8H, RMDF02H: F03C9H RMDF03L: F03D8H, RMDF03H: F03D9H
RMDF04L: F03E8H, RMDF04H: F03E9H RMDF05L: F03F8H, RMDF05H: F03F9H
RMDF06L: F0408H, RMDF06H: F0409H RMDF07L: F0418H, RMDF07H: F0419H
RMDF08L: F0428H, RMDF08H: F0429H RMDF09L: F0438H, RMDF09H: F0439H
RMDF010L: F0448H, RMDF010H: F0449H RMDF011L: F0458H, RMDF011H: F0459H
RMDF012L: F0468H, RMDF012H: F0469H RMDF013L: F0478H, RMDF013H: F0479H
RMDF014L: F0488H, RMDF014H: F0489H RMDF015L: F0498H, RMDF015H: F0499H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMDB1[7:0] | | | | | | | | RMDB0[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | RMDB1[7:0] | Receive Buffer Data Byte 1 | Data in the message stored in the receive buffer can be read. | R |
| 7 to 0 | RMDB0[7:0] | Receive Buffer Data Byte 0 | | R |

When the RMDLC[3:0] value in the RMPTRn register is smaller than B'1000, data bytes for which no data is set are read as H'00.

This register can be read when the RPAGE bit in the GRWCR register is 1.

### 18.3.31 CAN Receive Buffer Register nCH (RMDF1n) (n = 0 to 15)

Address RMDF10L: F03AAH, RMDF10H: F03ABH RMDF11L: F03BAH, RMDF11H: F03BBH
RMDF12L: F03CAH, RMDF12H: F03CBH RMDF13L: F03DAH, RMDF13H: F03DBH
RMDF14L: F03EAH, RMDF14H: F03EBH RMDF15L: F03FAH, RMDF15H: F03FBH
RMDF16L: F040AH, RMDF16H: F040BH RMDF17L: F041AH, RMDF17H: F041BH
RMDF18L: F042AH, RMDF18H: F042BH RMDF19L: F043AH, RMDF19H: F043BH
RMDF110L: F044AH, RMDF110H: F044BH RMDF111L: F045AH, RMDF111H: F045BH
RMDF112L: F046AH, RMDF112H: F046BH RMDF113L: F047AH, RMDF113H: F047BH
RMDF114L: F048AH, RMDF114H: F048BH RMDF115L: F049AH, RMDF115H: F049BH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMDB3[7:0] | | | | | | | | RMDB2[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | RMDB3[7:0] | Receive Buffer Data Byte 3 | Data in the message stored in the receive buffer can be read. | R |
| 7 to 0 | RMDB2[7:0] | Receive Buffer Data Byte 2 | | R |

When the RMDLC[3:0] value in the RMTPRn register is smaller than B'1000, data bytes for which no data is set are read as H'00.

This register can be read when the RPAGE bit in the GRWCR register is 1.

### 18.3.32 CAN Receive Buffer Register nDL (RMDF2n) (n = 0 to 15)

Address RMDF20L: F03ACH, RMDF20H: F03ADH RMDF21L: F03BCH, RMDF21H: F03BDH
RMDF22L: F03CCH, RMDF22H: F03CDH RMDF23L: F03DCH, RMDF23H: F03DDH
RMDF24L: F03ECH, RMDF24H: F03EDH RMDF25L: F03FCH, RMDF25H: F03FDH
RMDF26L: F040CH, RMDF26H: F040DH RMDF27L: F041CH, RMDF27H: F041DH
RMDF28L: F042CH, RMDF28H: F042DH RMDF29L: F043CH, RMDF29H: F043DH
RMDF210L: F044CH, RMDF210H: F044DH RMDF211L: F045CH, RMDF211H: F045DH
RMDF212L: F046CH, RMDF212H: F046DH RMDF213L: F047CH, RMDF213H: F047DH
RMDF214L: F048CH, RMDF214H: F048DH RMDF215L: F049CH, RMDF215H: F049DH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMDB5[7:0] | | | | | | | | RMDB4[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | RMDB5[7:0] | Receive Buffer Data Byte 5 | Data in the message stored in the receive buffer can be read. | R |
| 7 to 0 | RMDB4[7:0] | Receive Buffer Data Byte 4 | | R |

When the RMDLC[3:0] value in the RMPTRn register is smaller than B'1000, data bytes for which no data is set are read as H'00.

This register can be read when the RPAGE bit in the GRWCR register is 1.

### 18.3.33 CAN Receive Buffer Register nDH (RMDF3n) (n = 0 to 15)

Address RMDF30L: F03AEH, RMDF30H: F03AFH RMDF31L: F03BEH, RMDF31H: F03BFH
RMDF32L: F03CEH, RMDF32H: F03CFH RMDF33L: F03DEH, RMDF33H: F03DFH
RMDF34L: F03EEH, RMDF34H: F03EFH RMDF35L: F03FEH, RMDF35H: F03FFH
RMDF36L: F040EH, RMDF36H: F040FH RMDF37L: F041EH, RMDF37H: F041FH
RMDF38L: F042EH, RMDF38H: F042FH RMDF39L: F043EH, RMDF39H: F043FH
RMDF310L: F044EH, RMDF310H: F044FH RMDF311L: F045EH, RMDF311H: F045FH
RMDF312L: F046EH, RMDF312H: F046FH RMDF313L: F047EH, RMDF313H: F047FH
RMDF314L: F048EH, RMDF314H: F048FH RMDF315L: F049EH, RMDF315H: F049FH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RMDB7[7:0] | | | | | | | | RMDB6[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | RMDB7[7:0] | Receive Buffer Data Byte 7 | Data in the message stored in the receive buffer can be read. | R |
| 7 to 0 | RMDB6[7:0] | Receive Buffer Data Byte 6 | | R |

When the RMDLC[3:0] value in the RMPTRn register is smaller than B'1000, data bytes for which no data is set are read as H'00.

This register can be read when the RPAGE bit in the GRWCR register is 1.

### 18.3.34 CAN Receive FIFO Control Register m (RFCCm) (m = 0, 1)

Address RFCC0L: F0338H, RFCC0H: F0339H
RFCC1L: F033AH, RFCC1H: F033BH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFIGCV[2:0] | | | RFIM | — | RFDC[2:0] | | | — | — | — | — | — | — | RFIE | RFE |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 13 | RFIGCV[2:0] | Receive FIFO Interrupt Request Timing Select | b15 b14 b13<br>0 0 0 : When FIFO is 1/8 full.<br>0 0 1 : When FIFO is 2/8 full.<br>0 1 0 : When FIFO is 3/8 full.<br>0 1 1 : When FIFO is 4/8 full.<br>1 0 0 : When FIFO is 5/8 full.<br>1 0 1 : When FIFO is 6/8 full.<br>1 1 0 : When FIFO is 7/8 full.<br>1 1 1 : When FIFO is full. | R/W |
| 12 | RFIM | Receive FIFO Interrupt Source Select | 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met.<br>1: An interrupt occurs each time a message has been received. | R/W |
| 11 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R |
| 10 to 8 | RFDC[2:0] | Receive FIFO Buffer Depth Configuration | b10 b9 b8<br>0 0 0 : 0 messages<br>0 0 1 : 4 messages<br>0 1 0 : 8 messages<br>0 1 1 : 16 messages<br>1 0 0 : Setting prohibited<br>1 0 1 : Setting prohibited<br>1 1 0 : Setting prohibited<br>1 1 1 : Setting prohibited | R/W |
| 7 to 2 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 1 | RFIE | Receive FIFO Interrupt Enable | 0: Receive FIFO interrupt is disabled.<br>1: Receive FIFO interrupt is enabled. | R/W |
| 0 | RFE | Receive FIFO Buffer Enable | 0: No receive FIFO buffer is used.<br>1: Receive FIFO buffers are used. | R/W |

- RFIGCV[2:0] Bits
  These bits are used to specify the fraction of the transmit/receive FIFO buffer (the number of messages is selected by the setting of the RFDC[2:0] bits) that must be filled for the FIFO buffer to generate a receive interrupt request when the RFIM bit is set to 0.
  When the RFDC[2:0] bits are set to B'001 (4 messages), set the RFIGCV[2:0] bits to B'001, B'011, B'101, or B'111. Modify these bits only in global reset mode.

- RFIM Bit
  This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

- RFDC[2:0] Bits
  These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. If these bits are set to B'000, do not use any receive FIFO buffer. Modify these bits only in global reset mode.

- RFIE Bit
  Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit is set to 0 (no receive FIFO buffer is used).

- RFE Bit
  Setting the RFE bit to 1 makes receive FIFO buffers available. Clearing this bit to 0 sets the RFEMP flag in the RFSTSm register to 1 (the receive FIFO buffer contains no unread message (buffer empty)). Modify this bit only in global operating mode or global test mode.

### 18.3.35 CAN Receive FIFO Status Register m (RFSTSm) (m = 0, 1)

Address RFSTS0L: F0340H, RFSTS0H: F0341H
RFSTS1L: F0342H, RFSTS1H: F0343H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | RFMC[5:0] | | | | | | — | — | — | — | RFIF | RFMLT | RFFLL | RFEMP |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15, 14 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 13 to 8 | RFMC[5:0] | Receive FIFO Unread Message Counter | The number of unread messages stored in the receive FIFO buffer is displayed. | R |
| 7 to 4 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 3 | RFIF | Receive FIFO Interrupt Request Flag | 0: No receive FIFO interrupt request is present.<br>1: A receive FIFO interrupt request is present. | R/(W)Note |
| 2 | RFMLT | Receive FIFO Message Lost Flag | 0: No receive FIFO message is lost.<br>1: A receive FIFO message is lost. | R/(W)Note |
| 1 | RFFLL | Receive FIFO Buffer Full Status Flag | 0: The receive FIFO buffer is not full.<br>1: The receive FIFO buffer is full. | R |
| 0 | RFEMP | Receive FIFO Buffer Empty Status Flag | 0: The receive FIFO buffer contains unread messages.<br>1: The receive FIFO buffer contains no unread message (buffer empty). | R |

<R> **Note** The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state. To write 0 to this flag bit, write by using an 8-bit data transfer instruction or a 16-bit data transfer instruction.

- RFMC[5:0] Flag
  This flag indicates the number of unread messages in the receive FIFO buffer. This flag becomes H'00 when the RFE bit in the RFCCm register is set to 0.

- RFIF Flag
  This flag is set to 1 when the receive FIFO interrupt request generation conditions set by the RFIGCV[2:0] bits and the RFIM bit in the RFCCm register are met. This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit only in global operating mode or global test mode.

- RFMLT Flag
  This flag is set to 1 when it is attempted to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.
  This flag is cleared to 0 in global reset mode or by writing 0 to this flag.
  Modify this bit only in global operating mode or global test mode.

- RFFLL Flag
  This flag is set to 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFDC[2:0] bits in the RFCCm register.
  If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFDC[2:0] bits, this flag is cleared to 0. This flag is also cleared to 0 when the RFE bit in the RFCCm register is set to 0 (no receive FIFO buffer is used) or in global reset mode.

- RFEMP Flag
  This flag is set to 1 when all messages in the receive FIFO buffer have been read. This flag is also set to 1 when the RFE bit in the RFCCm register is 0 or in global reset mode.
  This flag is cleared to 0 when even a single received message has been stored in the receive FIFO buffer.

### 18.3.36 CAN Receive FIFO Pointer Control Register m (RFPCTRm) (m = 0, 1)

Address RFPCTR0L: F0348H, RFPCTR0H: F0349H
RFPCTR1L: F034AH, RFPCTR1H: F034BH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | RFPC[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | — | Reserved | The write value should always be 0. | R |
| 7 to 0 | RFPC[7:0] | Receive FIFO Pointer | When these bits are set to H'FF, the read pointer moves to the next unread message in the receive FIFO buffer. The setting for these bits must be H'FF. | W |

- RFPC[7:0] Bits
  When the RFPC[7:0] bits are set to H'FF, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFMC[5:0] (receive FIFO unread message counter) value in the RFSTSm register is decremented. Read the RFIDLm, RFIDHm, RFTSm, RFPTRm, and RFDF0m to RFDF3m registers to read messages in the receive FIFO buffer, and then write H'FF to the RFPC[7:0] bits.
  Write H'FF to these bits when the RFE bit in the RFCCm register is set to 1 (receive FIFO buffers are used) and the RFEMP flag in the RFSTSm register is 0 (the receive FIFO buffer contains unread messages).

### 18.3.37 CAN Receive FIFO Access Register mAL (RFIDLm) (m = 0, 1)

Address RFIDL0L: F05A0H, RFIDL0H: F05A1H
RFIDL1L: F05B0H, RFIDL1H: F05B1H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFID[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 0 | RFID[15:0] | Receive FIFO Buffer ID Data L | The standard ID or extended ID of received message can be read.<br>Read bits 10 to 0 for standard ID. Bits 15 to 11 are read as 0. | R |

This register can be read when the RPAGE bit in the GRWCR register is 1.

- RFID[15:0] Bits
  These bits indicate the ID of the message stored in the receive FIFO buffer.

### 18.3.38 CAN Receive FIFO Access Register mAH (RFIDHm) (m = 0, 1)

Address RFIDH0L: F05A2H, RFIDH0H: F05A3H
RFIDH1L: F05B2H, RFIDH1H: F05B3H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFIDE | RFRTR | — | RFID[28:16] | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 | RFIDE | Receive FIFO Buffer IDE | 0: Standard ID<br>1: Extended ID | R |
| 14 | RFRTR | Receive FIFO Buffer RTR | 0: Data frame<br>1: Remote frame | R |
| 13 | — | Reserved | This bit is always read as 0. | R |
| 12 to 0 | RFID[28:16] | Receive FIFO Buffer ID Data H | The standard ID or extended ID of received message can be read.<br>For standard ID, these bits are read as 0. | R |

This register can be read when the RPAGE bit in the GRWCR register is 1.

- RFIDE Bit
  This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

- RFRTR Bit
  This bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer.

- RFID[28:16] Bits
  These bits indicate the ID of the message stored in the receive FIFO buffer.

### 18.3.39 CAN Receive FIFO Access Register mBL (RFTSm) (m = 0, 1)

Address RFTS0L: F05A4H, RFTS0H: F05A5H
RFTS1L: F05B4H, RFTS1H: F05B5H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFTS[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 0 | RFTS[15:0] | Receive FIFO Buffer Timestamp Data | Timestamp value of the received message can be read. | R |

This register can be read when the RPAGE bit in the GRWCR register is 1.

- RFTS[15:0]
  These bits indicate the timestamp value of the message stored in the receive FIFO buffer.

### 18.3.40 CAN Receive FIFO Access Register mBH (RFPTRm) (m = 0, 1)

Address RFPTR0L: F05A6H, RFPTR0H: F05A7H
RFPTR1L: F05B6H, RFPTR1H: F05B7H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFDLC[3:0] | | | | RFPTR[11:0] | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 12 | RFDLC[3:0] | Receive FIFO Buffer DLC Data | b15 b14 b13 b12<br>0 0 0 0 : 0 data bytes<br>0 0 0 1 : 1 data byte<br>0 0 1 0 : 2 data bytes<br>0 0 1 1 : 3 data bytes<br>0 1 0 0 : 4 data bytes<br>0 1 0 1 : 5 data bytes<br>0 1 1 0 : 6 data bytes<br>0 1 1 1 : 7 data bytes<br>1 X X X : 8 data bytes | R |
| 11 to 0 | RFPTR[11:0] | Receive FIFO Buffer Label Data | Label information of the received message can be read. | R |

This register can be read when the RPAGE bit in the GRWCR register is 1.

- RFDLC[3:0] Bits
  These bits indicate the data length of the message stored in the receive FIFO buffer.

- RFPTR[11:0] Bits
  These bits indicate the label information of the message stored in the receive FIFO buffer.

### 18.3.41 CAN Receive FIFO Access Register mCL (RFDF0m) (m = 0, 1)

Address RFDF00L: F05A8H, RFDF00H: F05A9H
RFDF01L: F05B8H, RFDF01H: F05B9H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFDB1[7:0] | | | | | | | | RFDB0[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | RFDB1[7:0] | Receive FIFO Buffer Data Byte 1 | Data in the message stored in the receive FIFO buffer can be read. | R |
| 7 to 0 | RFDB0[7:0] | Receive FIFO Buffer Data Byte 0 | | R |

When the RFDLC[3:0] value in the RFPTRm register is smaller than B'1000, data bytes for which no data is set are read as H'00.

This register can be read when the RPAGE bit in the GRWCR register is 1.

### 18.3.42 CAN Receive FIFO Access Register mCH (RFDF1m) (m = 0, 1)

Address RFDF10L: F05AAH, RFDF10H: F05ABH
RFDF11L: F05BAH, RFDF11H: F05BBH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFDB3[7:0] | | | | | | | | RFDB2[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | RFDB3[7:0] | Receive FIFO Buffer Data Byte 3 | Data in the message stored in the receive FIFO buffer can be read. | R |
| 7 to 0 | RFDB2[7:0] | Receive FIFO Buffer Data Byte 2 | | R |

When the RFDLC[3:0] value in the RFPTRm register is smaller than B'1000, data bytes for which no data is set are read as H'00.

This register can be read when the RPAGE bit in the GRWCR register is 1.

### 18.3.43 CAN Receive FIFO Access Register mDL (RFDF2m) (m = 0, 1)

Address RFDF20L: F05ACH, RFDF20H: F05ADH
RFDF21L: F05BCH, RFDF21H: F05BDH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFDB5[7:0] | | | | | | | | RFDB4[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | RFDB5[7:0] | Receive FIFO Buffer Data Byte 5 | Data in the message stored in the receive FIFO buffer can be read. | R |
| 7 to 0 | RFDB4[7:0] | Receive FIFO Buffer Data Byte 4 | | R |

When the RFDLC[3:0] value in the RFPTRm register is smaller than B'1000, data bytes for which no data is set are read as H'00.

This register can be read when the RPAGE bit in the GRWCR register is 1.

### 18.3.44 CAN Receive FIFO Access Register mDH (RFDF3m) (m = 0, 1)

Address RFDF30L: F05AEH, RFDF30H: F05AFH
RFDF31L: F05BEH, RFDF31H: F05BFH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RFDB7[7:0] | | | | | | | | RFDB6[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | RFDB7[7:0] | Receive FIFO Buffer Data Byte 7 | Data in the message stored in the receive FIFO buffer can be read. | R |
| 7 to 0 | RFDB6[7:0] | Receive FIFO Buffer Data Byte 6 | | R |

When the RFDLC[3:0] value in the RFPTRm register is smaller than B'1000, data bytes for which no data is set are read as H'00.

This register can be read when the RPAGE bit in the GRWCR register is 1.

### 18.3.45 CANi Transmit/Receive FIFO Control Register kL (CFCCLk) (i = 0) (k = 0)

Address CFCCL0L: F0350H, CFCCL0H: F0351H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFIGCV[2:0] | | | CFIM | — | CFDC[2:0] | | | — | — | — | — | — | CFTXIE | CFRXIE | CFE |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 13 | CFIGCV[2:0] | Transmit/Receive FIFO Receive Interrupt Request Timing Select | b15 b14 b13<br>0 0 0 : When FIFO is 1/8 full.<br>0 0 1 : When FIFO is 2/8 full.<br>0 1 0 : When FIFO is 3/8 full.<br>0 1 1 : When FIFO is 4/8 full.<br>1 0 0 : When FIFO is 5/8 full.<br>1 0 1 : When FIFO is 6/8 full.<br>1 1 0 : When FIFO is 7/8 full.<br>1 1 1 : When FIFO is full. | R/W |
| 12 | CFIM | Transmit/Receive FIFO Interrupt Source Select | 0:<br>• Receive mode<br>When the number of received messages has met the condition set by the CFIGCV[2:0] bits, a FIFO receive interrupt request is generated.<br>• Transmit mode<br>When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated.<br>1:<br>• Receive mode<br>A FIFO receive interrupt request is generated each time a message has been received.<br>• Transmit mode<br>A FIFO transmit interrupt request is generated each time a message has been transmitted. | R/W |
| 11 | — | Reserved | This bit is always read as 0. The write value should always be 0. | R |
| 10 to 8 | CFDC[2:0] | Transmit/Receive FIFO Buffer Depth Configuration | b10 b9 b8<br>0 0 0 : 0 messages<br>0 0 1 : 4 messages<br>0 1 0 : 8 messages<br>0 1 1 : 16 messages<br>1 0 0 : Setting prohibited<br>1 0 1 : Setting prohibited<br>1 1 0 : Setting prohibited<br>1 1 1 : Setting prohibited | R/W |
| 7 to 3 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 2 | CFTXIE | Transmit/Receive FIFO Transmit Interrupt Enable | 0: Transmit/receive FIFO transmit interrupt is disabled.<br>1: Transmit/receive FIFO transmit interrupt is enabled. | R/W |
| 1 | CFRXIE | Transmit/Receive FIFO Receive Interrupt Enable | 0: Transmit/receive FIFO receive interrupt is disabled.<br>1: Transmit/receive FIFO receive interrupt is enabled. | R/W |
| 0 | CFE | Transmit/Receive FIFO Buffer Enable | 0: No transmit/receive FIFO buffer is used.<br>1: Transmit/receive FIFO buffers are used. | R/W |

- CFIGCV[2:0] Bits
  These bits are used to specify the fraction of the transmit/receive FIFO buffer (the number of messages is selected by the setting of the CFDC[2:0] bits) that must be filled for the FIFO buffer to generate a receive interrupt request when the CFM[1:0] bits are set to B'00 (reception mode) and the CFIM bit is set to 0.
  When the CFDC[2:0] bits are set to B'001 (4 messages), set the CFIGCV[2:0] bits to B'001, B'011, B'101, or B'111.
  Modify these bits only in global reset mode.

- CFIM Bit
  This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

- CFDC[2:0] Bits
  These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer. If these bits are set to B'000, do not use any receive FIFO buffer. Modify these bits only in global reset mode.

- CFTXIE Bit
  When this bit is set to 1 and the CFTXIF flag in the CFSTSk register is set to 1, a transmit/receive FIFO transmit interrupt request is generated.
  Modify this bit with the CFE bit set to 0 (no transmit/receive FIFO buffer is used).

- CFRXIE Bit
  When this bit is set to 1 and the CFRXIF flag in the CFSTSk register is set to 1, a transmit/receive FIFO receive interrupt request is generated.
  Modify this bit with the CFE bit set to 0.

- CFE Bit
  Setting this bit to 1 makes transmit/receive FIFO buffers available.
  When this bit is set to 0 in transmit mode, if a message in the transmit/receive FIFO buffer is being transmitted or to be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission, CAN bus error detection, or arbitration lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

  This bit is cleared to 0 when the following conditions are met.
  - Receive mode: Global reset mode
  - Transmit mode: Channel reset mode

  Modify this bit only in the following mode.
  - Receive mode: Global operating mode or global test mode
  - Transmit mode: Channel communication mode or channel halt mode

### 18.3.46 CANi Transmit/Receive FIFO Control Register kH (CFCCHk) (i = 0) (k = 0)

Address CFCCH0L: F0352H, CFCCH0H: F0353H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFITT[7:0] | | | | | | | | — | — | CFTML[1:0] | | CF ITR | CF ITSS | CFM[1:0] | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | CFITT[7:0] | Message Transmission Interval Configuration | Set a message transmission interval.<br>Set these bits to a value within a range of H'00 to H'FF. | R/W |
| 7, 6 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 5, 4 | CFTML[1:0] | Transmit Buffer Link Configuration | Set the transmit buffer number to be linked to the transmit/receive FIFO buffer. | R/W |
| 3 | CFITR | Transmit/Receive FIFO Interval Timer Resolution | 0: Clock obtained by frequency-dividing $f_{CLK}$/2 by the ITRCP[15:0] value<br>1: Clock obtained by frequency-dividing $f_{CLK}$/2 by the ITRCP[15:0] value × 10 | R/W |
| 2 | CFITSS | Interval Timer Clock Source Select | 0: Clock selected by the CFITR bit<br>1: CANi bit time clock | R/W |
| 1, 0 | CFM[1:0] | Transmit/Receive FIFO Mode Select | b1 b0<br>0 0 : Receive mode<br>0 1 : Transmit mode<br>1 0 : Setting prohibited<br>1 1 : Setting prohibited | R/W |

- CFITT[7:0] Bits
  These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to B'01 (transmit mode).
  Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) and then modify the CFITT[7:0] bits.

- CFTML[1:0] Bits
  These bits are used to set the number of transmit buffer to be linked to the transmit/receive FIFO buffer when the CFM[1:0] bits are set to B'01 (transmit mode).
  Setting the CFDC[2:0] bits to B'001 or more enables the setting of the CFTML[1:0] bits.
  Modify these bits only in global reset mode.

- CFITR Bit
  This bit is valid when the setting of the CFITSS bit is 0.
  Setting this bit to 0 selects the clock obtained by frequency-dividing $f_{CLK}$/2 by the ITRCP[15:0] value.
  Setting this bit to 1 selects the clock obtained by frequency-dividing $f_{CLK}$/2 by the ITRCP[15:0] value × 10.
  Modify the CFITR bit with the CFE bit in theCFCCLk register set to 0 (no transmit/receive FIFO buffer is used).

- CFITSS Bit
  Setting this bit to 0 selects the clock selected by the CFITR bit as the clock source for counting by the interval timer.
  Setting this bit to 1 selects the CANi bit time clock as the clock source for counting by the interval timer. Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITSS bit.

- CFM[1:0] Bits
  These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

### 18.3.47 CANi Transmit/Receive FIFO Status Register k (CFSTSk) (i = 0) (k = 0)

Address CFSTS0L: F0358H, CFSTS0H: F0359H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | CFMC[5:0] | | | | | | — | — | — | CFTXIF | CFRXIF | CFMLT | CFFLL | CFEMP |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15, 14 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 13 to 8 | CFMC[5:0] | Transmit/Receive FIFO Message Counter | The number of messages stored in the transmit/receive FIFO buffer is indicated. | R |
| 7 to 5 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 4 | CFTXIF | Transmit/Receive FIFO Transmit Interrupt Request Flag | 0: No transmit/receive FIFO transmit interrupt request is present.<br>1: A transmit/receive FIFO transmit interrupt request is present. | R/(W)Note |
| 3 | CFRXIF | Transmit/Receive FIFO Receive Interrupt Request Flag | 0: No transmit/receive FIFO receive interrupt request is present.<br>1: A transmit/receive FIFO receive interrupt request is present. | R/(W)Note |
| 2 | CFMLT | Transmit/Receive FIFO Message Lost Flag | 0: No transmit/receive FIFO message is lost.<br>1: A transmit/receive FIFO message is lost. | R/(W)Note |
| 1 | CFFLL | Transmit/Receive FIFO Buffer Full Status Flag | 0: The transmit/receive FIFO buffer is not full.<br>1: The transmit/receive FIFO buffer is full. | R |
| 0 | CFEMP | Transmit/Receive FIFO Buffer Empty Status Flag | 0: The transmit/receive FIFO buffer contains messages.<br>1: The transmit/receive FIFO buffer contains no message (buffer empty). | R |

<R> **Note** The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state. To write 0 to this flag bit, write by using an 8-bit data transfer instruction or a 16-bit data transfer instruction.

- CFMC[5:0] Bits

  The CFMC[5:0] bits indicate the following values that depend on the setting of the CFM[1:0] bits in the CFCCHk register.

  • When CFM[1:0] value is B'01 (transmit mode): Number of untransmitted messages in the buffer

  • When CFM[1:0] value is B'00 (receive mode): Number of unread received messages in the buffer

  These bits are cleared to 0 when any of the following conditions is met.

  • When CFM[1:0] value is B'00: In global reset mode

  • When CFM[1:0] value is B'01: In channel reset mode

- CFTXIF Flag

  The CFTXIF flag is set to 1 when the following condition is met.

  • When CFM[1:0] value is B'01 and interrupt source setting the CFIM bit in the CFCCLk register is generated.

  The CFTXIF flag is cleared to 0 when any of the following conditions is met.

  • Write 0 to the CFTXIF flag

  • When CFM[1:0] value is B'00: In global reset mode

  • When CFM[1:0] value is B'01: In channel reset mode

  Clear this flag to 0 in global operating mode or global test mode.

- CFRXIF Flag
  The CFRXIF flag is set to 1 when the following condition is met.
  - When CFM[1:0] value is B'00 and interrupt source setting the CFIM bit in the CFCCLk register is generated.

  The CFRXIF flag is cleared to 0 when any of the following conditions is met.
  - Write 0 to the CFRXIF flag
  - When CFM[1:0] value is B'00: In global reset mode
  - When CFM[1:0] value is B'01: In channel reset mode

  Clear this flag to 0 in global operating mode or global test mode.

- CFMLT Flag
  The CFMLT flag is set to 1 when the following condition is met.
  - When it is attempted to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

  The CFMLT flag is cleared to 0 when any of the following conditions is met.
  - Write 0 to the CFMLT flag
  - When CFM[1:0] value is B'00: In global reset mode
  - When CFM[1:0] value is B'01: In channel reset mode

  Clear this flag to 0 in global operating mode or global test mode.

- CFFLL Flag
  The CFFLL flag is set to 1 when the following condition is met.
  - When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFDC[2:0] bits in the CFCCLk register.

  The CFFLL flag is cleared to 0 when any of the following conditions is met.
  - When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFDC[2:0] bits.
  - When the CFE value in the CFCCLk register is 0 (no transmit/receive FIFO buffer is used).
    Note that this flag is set to 0 after transmission completion, CAN bus error detection, or arbitration lost when the message in the transmit/receive FIFO buffer is being transmitted or to be transmitted next.
  - When CFM[1:0] value is B'00: In global reset mode
  - When CFM[1:0] value is B'01: In channel reset mode

- CFEMP Flag
  The CFEMP flag is set to 1 when any of the following conditions is met.
  - When the CFM[1:0] value is B'00: All messages have been read, or global reset mode.
  - When the CFM[1:0] value is B'01: All messages have been transmitted, or channel reset mode.
  - When the CFE value in the CFCCLk register is 0 (no transmit/receive FIFO buffer is used).
    Note that this flag is set to 1 after transmission completion, CAN bus error detection, or arbitration lost when the message in the transmit/receive FIFO buffer is being transmitted or to be transmitted next.

  The CFEMP flag is cleared to 0 when any of the following conditions is met.
  - When the CFM[1:0] value is B'00: Any one of received messages has been stored in the transmit/receive FIFO buffer.
  - When the CFM[1:0] value is B'01: A value H'FF has been written to the CFPCTRk register after data was written to the CFIDLk, CFIDHk, CFPTRk, and CFDF0k to CFDF3k registers.

### 18.3.48 CANi Transmit/Receive FIFO Pointer Control Register k (CFPCTRk) (i = 0) (k = 0)

Address CFPCTR0L: F035CH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | CFPC[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | — | Reserved | The write value should always be 0. | R |
| 7 to 0 | CFPC[7:0] | CANi Transmit/Receive FIFO Pointer | Receive mode:<br>Writing H'FF to these bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer.<br>Transmit mode:<br>Writing H'FF to these bits moves the write pointer to the next stage of the transmit/receive FIFO buffer. | W |

- CFPC[7:0] Bits

  Receive mode (CFM [1:0] value in the CFCCHk register is B'00):

  Writing H'FF to the CFPC[7:0] bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFMC[5:0] value (transmit/receive FIFO message counter) in the CFSTSk register is decremented. Read the CFIDLk, CFIDHk, CFTSk, CFPTRk, and CFDF0k to CFDF3k registers to read messages in the transmit/receive FIFO buffer, and then write H'FF to the CFPC[7:0] bits.

  Write H'FF to these bits when the CFE bit in the CFCCLk register is set to 1 (transmit/receive FIFO buffers are used) and the CFEMP flag in the CFSTSk register is cleared to 0 (the transmit/receive FIFO buffer contains messages).

  Transmit mode (CFM [1:0] value in the CFCCHk register is B'01):

  Writing H'FF to the CFPC[7:0] bits stores the data written to the CFIDLk, CFIDHk, CFPTRk, and CFDF0k to CFDF3k registers in the transmit/receive FIFO buffer and moves the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFMC[5:0] value is incremented. Write transmit messages to the CFIDLk, CFIDHk, CFPTRk, and CFDF0k to CFDF3k registers and then write H'FF to the CFPC[7:0] bits.

  Write H'FF to these bits when the CFE bit in the CFCCLk register is set to 1 and the CFFLL flag in the CFSTSk register is cleared to 0 (the transmit/receive FIFO buffer is not full).

### 18.3.49 CANi Transmit/Receive FIFO Access Register kAL (CFIDLk) (i = 0) (k = 0)

Address CFIDL0L: F05E0H, CFIDL0H: F05E1H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFID[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 0 | CFID[15:0] | Transmit/Receive FIFO Buffer ID Data L | When CFM[1:0] value is B'01 (transmit mode):<br>Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 15 to 11.<br>When CFM[1:0] value is B'00 (receive mode):<br>Standard ID or extended ID in the received message can be read. For standard ID, read bits 10 to 0. Bits 15 to 11 are read as 0. | R/W |

Modify these bits only when the CFM[1:0] value in the CFCCHk register is B'01 (transmit mode). This register is readable only when the CFM[1:0] value is B'00 (receive mode).

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

- CFID[15:0] Bits
  These bits indicate the ID of the received message stored in the transmit/receive FIFO buffer when the CFM [1:0] value is B'00.
  When the CFM [1:0] value is B'01, set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

### 18.3.50 CANi Transmit/Receive FIFO Access Register kAH (CFIDHk) (i = 0) (k = 0)

Address CFIDH0L: F05E2H, CFIDH0H: F05E3H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFIDE | CFRTR | THLEN | CFID[28:16] | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 | CFIDE | Transmit/Receive FIFO Buffer IDE | 0: Standard ID<br>1: Extended ID | R/W |
| 14 | CFRTR | Transmit/Receive FIFO Buffer RTR | 0: Data frame<br>1: Remote frame | R/W |
| 13 | THLEN | Transmit History Data Store Enable | This bit is valid only when the CFM[1:0] value is B'01 (transmit mode).<br>0: Transmit history data is not stored in the buffer.<br>1: Transmit history data is stored in the buffer. | R/W |
| 12 to 0 | CFID[28:16] | Transmit/Receive FIFO Buffer ID Data H | When CFM[1:0] value is B'01 (transmit mode):<br>Set standard ID or extended ID. For standard ID, write 0 to these bits.<br>When CFM[1:0] value is B'00 (receive mode):<br>Standard ID or extended ID in the received message can be read. For standard ID, these bits are read as 0. | R/W |

Modify these bits only when the CFM[1:0] value in the CFCCHk register is B'01 (transmit mode). This register is readable only when the CFM[1:0] value is B'00 (receive mode).

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

- CFIDE Bit
  This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is B'00. When the CFM [1:0] value is B'01, set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

- CFRTR Bit
  This bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFM [1:0] value is B'00. When the CFM [1:0] value is B'01, set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

- THLEN Bit
  When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmit messages is stored in the transmit history buffer after transmission is completed.
  This bit is enabled when the CFM [1:0] value is B'01 (transmit mode).

- CFID[28:16] Bits
  These bits indicate the ID of the received message stored in the transmit/receive FIFO buffer when the CFM [1:0] value is B'00.
  When the CFM[1:0] value is B'01, set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

### 18.3.51 CANi Transmit/Receive FIFO Access Register kBL (CFTSk) (i = 0) (k = 0)

Address CFTS0L: F05E4H, CFTS0H: F05E5H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFTS[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 0 | CFTS[15:0] | Transmit/Receive FIFO Buffer Timestamp Data | These bits are valid only when the CFM[1:0] value is B'00 (receive mode).<br>The timestamp value of the received message can be read. | R |

This register can be read when the RPAGE bit in the GRWCR register is 1.

- CFTS[15:0] Bits
  These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.
  These bits are valid when the CFM[1:0] value is B'00.

### 18.3.52 CANi Transmit/Receive FIFO Access Register kBH (CFPTRk) (i = 0) (k = 0)

Address CFPTR0L: F05E6H, CFPTR0H: F05E7H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFDLC[3:0] | | | | CFPTR[11:0] | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 12 | CFDLC[3:0] | Transmit/Receive FIFO Buffer DLC Data | b15 b14 b13 b12<br>0 0 0 0: 0 data bytes<br>0 0 0 1: 1 data byte<br>0 0 1 0: 2 data bytes<br>0 0 1 1: 3 data bytes<br>0 1 0 0: 4 data bytes<br>0 1 0 1: 5 data bytes<br>0 1 1 0: 6 data bytes<br>0 1 1 1: 7 data bytes<br>1 X X X: 8 data bytes | R/W |
| 11 to 0 | CFPTR[11:0] | Transmit/Receive FIFO Buffer Label Data | When CFM[1:0] value is B'01 (transmit mode):<br>Set the label information to be stored in the transmit history buffer.<br>Only bits CFPTR[7:0] are valid.<br>When CFM[1:0] value is B'00 (receive mode):<br>The label information of the received message can be read. | R/W |

Modify these bits only when the CFM[1:0] value in the CFCCHk register is B'01 (transmit mode). This register is readable only when the CFM[1:0] value is B'00 (receive mode).

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

- CFDLC[3:0] Bits
  These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFM [1:0] value is B'00. When the CFM [1:0] value is B'01, set the data length of the message to be transmitted from the transmit/receive FIFO buffer. If 9-byte or more data length is set, 8 bytes of data is actually transmitted.

- CFPTR[11:0] Bits
  These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFM [1:0] value is B'00. When the CFM [1:0] value is B'01, the CFPTR [7:0] value is stored in the transmit history buffer when message transmission has been completed.

### 18.3.53 CANi Transmit/Receive FIFO Access Register kCL (CFDF0k) (i = 0) (k = 0)

Address CFDF00L: F05E8H, CFDF00H: F05E9H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFDB1[7:0] | | | | | | | | CFDB0[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | CFDB1[7:0] | Transmit/Receive FIFO Buffer Data Byte 1 | When CFM[1:0] value is B'01 (transmit mode):<br>Set the transmit/receive FIFO buffer data.<br>When CFM[1:0] value is B'00 (receive mode):<br>The message data stored in the transmit/receive FIFO buffer can be read. | R/W |
| 7 to 0 | CFDB0[7:0] | Transmit/Receive FIFO Buffer Data Byte 0 | | R/W |

Modify these bits only when the CFM[1:0] value in the CFCCHk register is B'01.

This register is readable only when the CFM[1:0] value is B'00. When the CFDLC [3:0] value in the CFPTRk register is smaller than B'1000, data bytes for which no data is set are read as H'00.

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

### 18.3.54 CANi Transmit/Receive FIFO Access Register kCH (CFDF1k) (i = 0) (k = 0)

Address CFDF10L: F05EAH, CFDF10H: F05EBH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFDB3[7:0] | | | | | | | | CFDB2[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | CFDB3[7:0] | Transmit/Receive FIFO Buffer Data Byte 3 | When CFM[1:0] value is B'01 (transmit mode):<br>Set the transmit/receive FIFO buffer data.<br>When CFM[1:0] value is B'00 (receive mode):<br>The message data stored in the transmit/receive FIFO buffer can be read. | R/W |
| 7 to 0 | CFDB2[7:0] | Transmit/Receive FIFO Buffer Data Byte 2 | | R/W |

Modify these bits only when the CFM[1:0] value in the CFCCHk register is B'01.

This register is readable only when the CFM[1:0] value is B'00. When the CFDLC [3:0] value in the CFPTRk register is smaller than B'1000, data bytes for which no data is set are read as H'00.

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

### 18.3.55 CANi Transmit/Receive FIFO Access Register kDL (CFDF2k) (i = 0) (k = 0)

Address CFDF20L: F05ECH, CFDF20H: F05EDH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFDB5[7:0] | | | | | | | | CFDB4[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | CFDB5[7:0] | Transmit/Receive FIFO Buffer Data Byte 5 | When CFM[1:0] value is B'01 (transmit mode): Set the transmit/receive FIFO buffer data. When CFM[1:0] value is B'00 (receive mode): The message data stored in the transmit/receive FIFO buffer can be read. | R/W |
| 7 to 0 | CFDB4[7:0] | Transmit/Receive FIFO Buffer Data Byte 4 | | R/W |

Modify these bits only when the CFM[1:0] value in the CFCCHk register is B'01.

This register is readable only when the CFM[1:0] value is B'00. When the CFDLC [3:0] value in the CFPTRk register is smaller than B'1000, data bytes for which no data is set are read as H'00.

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

### 18.3.56 CANi Transmit/Receive FIFO Access Register kDH (CFDF3k) (i = 0) (k = 0)

Address CFDF30L: F05EEH, CFDF30H: F05EFH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CFDB7[7:0] | | | | | | | | CFDB6[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | CFDB7[7:0] | Transmit/Receive FIFO Buffer Data Byte 7 | When CFM[1:0] value is B'01 (transmit mode): Set the transmit/receive FIFO buffer data. When CFM[1:0] value is B'00 (receive mode): The message data stored in the transmit/receive FIFO buffer can be read. | R/W |
| 7 to 0 | CFDB6[7:0] | Transmit/Receive FIFO Buffer Data Byte 6 | | R/W |

Modify these bits only when the CFM[1:0] value in the CFCCHk register is B'01.

This register is readable only when the CFM[1:0] value is B'00. When the CFDLC [3:0] value in the CFPTRk register is smaller than B'1000, data bytes for which no data is set are read as H'00.

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

### 18.3.57 Receive FIFO Message Lost Status Register (RFMSTS)

Address RFMSTS: F0360H

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | RF1MLT | RF0MLT |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 2 | — | Reserved | These bits are always read as 0. | R |
| 1 | RF1MLT | Receive FIFO Buffer 1 Message Lost Status Flag | 0: No receive FIFO buffer m message is lost (m = 0, 1).<br>1: A receive FIFO buffer m message is lost. | R |
| 0 | RF0MLT | Receive FIFO Buffer 0 Message Lost Status Flag | | R |

The RFMSTS register is cleared to H'00 in global reset mode.

- RFmMLT Flag
  The RFmMLT flag is set to 1 when the RFMLT flag in the RFSTSm register is set to 1 (a receive FIFO message is lost). When the RFMLT flag is cleared to 0, the RFmMLT flag is cleared to 0.

### 18.3.58 CANi Transmit/Receive FIFO Message Lost Status Register (CFMSTS) (i = 0)

Address CFMSTS: F0361H

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | CF0MLT |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7-1 | — | Reserved | These bits are always read as 0. | R |
| 0 | CF0MLT | CANi Transmit/Receive FIFO Buffer k Message Lost Status Flag | 0: No CANi transmit/receive FIFO buffer k message is lost.<br>1: A CANi transmit/receive FIFO buffer k message is lost. | R |

The CFMSTS register is cleared to H'00 in global reset mode.

- CF0MLT Flag
  The CF0MLT flag is set to 1 when the CFMLT flag in the CFSTSk register is set to 1 (a transmit/receive FIFO message is lost). When the CFMLT flag is cleared to 0, the CF0MLT flag is cleared to 0.

### 18.3.59 CAN Receive FIFO Interrupt Status Register (RFISTS)

Address RFISTS: F0362H

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | RF1IF | RF0IF |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 2 | — | Reserved | These bits are always read as 0. | R |
| 1 | RF1IF | Receive FIFO Buffer 1 Interrupt Request Status Flag | 0: No receive FIFO buffer m interrupt request is present (m = 0, 1).<br>1: A receive FIFO buffer m interrupt request is present. | R |
| 0 | RF0IF | Receive FIFO Buffer 0 Interrupt Request Status Flag | | R |

The RFISTS register is cleared to H'00 in global reset mode.

- RFmIF Flag
  The RFmIF flag is set to 1 when the RFIF flag in the RFSTSm register is set to 1 (a receive FIFO interrupt request is present). When the RFIF flag is cleared to 0, the RFmIF flag is cleared to 0.

### 18.3.60 CAN Transmit/Receive FIFO Receive Interrupt Status Register (CFISTS)

Address CFISTS: F0363H

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | CF0IF |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 1 | — | Reserved | These bits are always read as 0. | R |
| 0 | CF0IF | CANi Transmit/Receive FIFO Buffer k Receive Interrupt Request Status Flag | 0: No CANi transmit/receive FIFO buffer k receive interrupt request is present.<br>1: A CANi transmit/receive FIFO buffer k receive interrupt request is present. | R |

The CFISTS register is cleared to H'00 in global reset mode.

- CF0IF Flag
  The CF0IF flag is set to 1 when the CFRXIF flag in the CFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). When the CFRXIF flag is cleared to 0, the CF0IF flag is cleared to 0.

### 18.3.61 CANi Transmit Buffer Control Register p (TMCp) (i = 0) (p = 0 to 3)

Address TMC0: F0364H, TMC1: F0365H
TMC2: F0366H, TMC3: F0367H

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | TMOM | TMTAR | TMTR |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 3 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 2 | TMOM | One-Shot Transmission Enable | 0: One-shot transmission is disabled.<br>1: One-shot transmission is enabled. | R/W |
| 1 | TMTAR | Transmit Abort Request | 0: Transmit abort is not requested.<br>1: Transmit abort is requested. | R/(W)Note |
| 0 | TMTR | Transmit Request | 0: Transmission is not requested.<br>1: Transmission is requested. | R/(W)Note |

**Note** The only effective value for writing to this bit is 1, which sets the bit. Otherwise writing to the bit results in retention of its state.

When the TMCp register meets any of the following conditions, set it to H'00.

- The TMCp register corresponds to the transmit buffer number selected by the CFTML[1:0] bits in the CFCCHk register.

Bits in the TMCp register are cleared to all 0 in channel reset mode. Modify the TMCp register only in channel communication mode or channel halt mode.

- TMOM Bit
  Setting this bit to 1 enables one-shot transmission. When transmission fails, retransmission defined in the CAN protocol is not performed.
  Modify the TMOM bit when the TMTRM flag in the TMSTSp register is set to 0. To set the TMOM bit to 1, also set the TMTR bit together.

- TMTAR Bit
  Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or to be transmitted next cannot be aborted.
  When the TMTR bit is set to 1, the TMTAR bit can be set to 1.
  The TMTAR bit is cleared to 0 when any of the following conditions is met, but is not cleared by writing 0 by the program.
  - Transmission has been completed.
  - Transmit abort has been completed.
  - An error or arbitration lost has been detected.

  If this bit becomes 0 at the timing when the program writes 1 to this bit, this bit becomes 0.

- TMTR Bit
  Setting this bit to 1 transmits the message stored in the transmit buffer.
  The TMTR bit is cleared to 0 when any of the following conditions is met, but is not cleared by writing 0 by the program.
  - Transmission has been completed.
  - Transmit abort has been completed by setting the TMTAR bit to 1.
  - An error or arbitration lost has been detected with the TMOM bit set to 1.

  Set the TMTR bit to 1 when the TMTRF[1:0] value in the TMSTSp register is B'00.

### 18.3.62 CANi Transmit Buffer Status Register p (TMSTSp) (i = 0) (p = 0 to 3)

Address TMSTS0: F036CH, TMSTS1: F036DH
TMSTS2: F036EH, TMSTS3: F036FH

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | TMTARM | TMTRM | TMTRF[1:0] | | TMTSTS |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 5 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 4 | TMTARM | Transmit Buffer Transmit Abort Request Status Flag | 0: No transmit abort request is present.<br>1: A transmit abort request is present. | R |
| 3 | TMTRM | Transmit Buffer Transmit Request Status Flag | 0: No transmit request is present.<br>1: A transmit request is present. | R |
| 2, 1 | TMTRF[1:0] | Transmit Buffer Transmit Result Flag | b2 b1<br>0 0 : Transmission is in progress or no transmit request is present.<br>0 1 : Transmit abort has been completed.<br>1 0 : Transmission has been completed (without transmit abort request).<br>1 1 : Transmission has been completed (with transmit abort request). | R/W |
| 0 | TMTSTS | Transmit Buffer Transmit Status Flag | 0: Transmission is not in progress.<br>1: Transmission is in progress. | R |

The TMSTSp register is cleared to all 0 in channel reset mode.

- TMTARM Flag
  The TMTARM flag is set to 1 when the TMTAR bit in the TMCp register is set to 1, and is cleared to 0 when the TMTAR bit is set to 0.

- TMTRM Flag
  The TMTRM flag is set to 1 when the TMTR bit in the TMCp register is set to 1, and is cleared to 0 when the TMTR bit is set to 0.

- TMTRF[1:0] Flag
  This flag indicates the result of transmission from the transmit buffer.
  B'00: Transmission is in progress or no transmit request is present.
  B'01: Transmission from the transmit buffer was aborted.
  B'10: Transmission has been completed with the TMTAR bit in the TMCp register set to 0 (transmit abort is not requested).
  B'11: Transmission has been completed with the TMTAR bit in the TMCp register set to 1 (transmit abort is requested).
  Write B'00 to the TMTRF[1:0] flag in channel communication mode or channel halt mode. Do not write any value other than B'00 to this flag.

- TMTSTS Flag
  This flag is set to 1 when transmission from the transmit buffer starts, and is cleared to 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

### 18.3.63 CANi Transmit Buffer Transmit Request Status Register (TMTRSTS) (i = 0)

Address TMTRSTSL: F0374H, TMTRSTSH: F0375H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | TMTRSTS3 | TMTRSTS2 | TMTRSTS1 | TMTRSTS0 |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 4 | — | Reserved | These bits are always read as 0. | R |
| 3 | TMTRSTS3 | CANi Transmit Buffer 3 Transmit Request Status Flag | 0: No transmit request is present.<br>1: A transmit request is present. | R |
| 2 | TMTRSTS2 | CANi Transmit Buffer 2 Transmit Request Status Flag | | R |
| 1 | TMTRSTS1 | CANi Transmit Buffer 1 Transmit Request Status Flag | | R |
| 0 | TMTRSTS0 | CANi Transmit Buffer 0 Transmit Request Status Flag | | R |

- TMTRSTSp Flags (p = 0 to 3)
  These flags indicate the status of the TMTR bit in the TMCp register.
  When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag is set to 1.
  The corresponding TMTRSTSp flag is cleared to 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

### 18.3.64 CANi Transmit Buffer Transmit Complete Status Register (TMTCSTS) (i = 0)

Address TMTCSTSL: F0376H, TMTCSTSH: F0377H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | TMTCSTS3 | TMTCSTS2 | TMTCSTS1 | TMTCSTS0 |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 4 | — | Reserved | These bits are always read as 0. | R |
| 3 | TMTCSTS3 | CANi Transmit Buffer 3 Transmit Complete Status Flag | 0: Transmission has not been completed.<br>1: Transmission has been completed | R |
| 2 | TMTCSTS2 | CANi Transmit Buffer 2 Transmit Complete Status Flag | | R |
| 1 | TMTCSTS1 | CANi Transmit Buffer 1 Transmit Complete Status Flag | | R |
| 0 | TMTCSTS0 | CANi Transmit Buffer 0 Transmit Complete Status Flag | | R |

- TMTCSTSp Flags (p = 0 to 3)
  When the TMTRF[1:0] flag in the TMSTSp register is set to B'10 (transmission has been completed (without transmit abort request)) or B'11 (transmission has been completed (with transmit abort request)), the corresponding TMTCSTSp flag is set to 1.
  These flags are cleared to 0 when the corresponding TMTRF [1:0] flag is set to B'00 or in channel reset mode.

### 18.3.65 CANi Transmit Buffer Transmit Abort Status Register (TMTASTS) (i = 0)

Address TMTASTSL: F0378H, TMTASTSH: F0379H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | TMTASTS3 | TMTASTS2 | TMTASTS1 | TMTASTS0 |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 4 | — | Reserved | These bits are always read as 0. | R |
| 3 | TMTASTS3 | CANi Transmit Buffer 3 Transmit Abort Status Flag | 0: Transmission is not aborted.<br>1: Transmission is aborted. | R |
| 2 | TMTASTS2 | CANi Transmit Buffer 2 Transmit Abort Status Flag | | R |
| 1 | TMTASTS1 | CANi Transmit Buffer 1 Transmit Abort Status Flag | | R |
| 0 | TMTASTS0 | CANi Transmit Buffer 0 Transmit Abort Status Flag | | R |

- TMTASTSp Flags (p = 0 to 3)
  When the TMTRF[1:0] flag in the TMSTSp register is set to B'01 (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.
  These flags are cleared to 0 when the corresponding TMTRF[1:0] flag is set to B'00 or in channel reset mode.

### 18.3.66 CANi Transmit Buffer Interrupt Enable Register (TMIEC) (i = 0)

Address TMIECL: F037AH, TMIECH: F037BH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | TMIE3 | TMIE2 | TMIE1 | TMIE0 |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 4 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 3 | TMIE3 | CANi Transmit Buffer 3 Interrupt Enable | 0: Transmit buffer interrupt is disabled.<br>1: Transmit buffer interrupt is enabled. | R/W |
| 2 | TMIE2 | CANi Transmit Buffer 2 Interrupt Enable | | R/W |
| 1 | TMIE1 | CANi Transmit Buffer 1 Interrupt Enable | | R/W |
| 0 | TMIE0 | CANi Transmit Buffer 0 Interrupt Enable | | R/W |

- TMIEp Bits (p = 0 to 3)
  When any of these bits is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.
  Modify these bits when the TMTRM flag in the corresponding TMSTSp register is 0 (no transmit request is present).
  Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers.

### 18.3.67 CANi Transmit Buffer Register pAL (TMIDLp) (i = 0) (p = 0 to 3)

Address TMIDL0L: F0600H, TMIDL0H: F0601H TMIDL1L: F0610H, TMIDL1H: F0611H
TMIDL2L: F0620H, TMIDL2H: F0621H TMIDL3L: F0630H, TMIDL3H: F0631H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMID[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 0 | TMID[15:0] | Transmit Buffer ID Data L | Set standard ID or extended ID.<br>For standard ID, write an ID to bits 10 to 0 and write 0 to bits 15 to 11. | R/W |

Modify this register when the TMTRM bit in the corresponding TMSTSp register is set to 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

- TMID[15:0] Bits
  These bits are used to set the ID of the message to be transmitted from the transmit buffer.

### 18.3.68 CANi Transmit Buffer Register pAH (TMIDHp) (i = 0) (p = 0 to 3)

Address TMIDH0L: F0602H, TMIDH0H: F0603H TMIDH1L: F0612H, TMIDH1H: F0613H
TMIDH2L: F0622H, TMIDH2H: F0623H TMIDH3L: F0632H, TMIDH3H: F0633H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMIDE | TMRTR | THLEN | TMID[28:16] | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 | TMIDE | Transmit Buffer IDE | 0: Standard ID<br>1: Extended ID | R/W |
| 14 | TMRTR | Transmit Buffer RTR | 0: Data frame<br>1: Remote frame | R/W |
| 13 | THLEN | Transmit History Data Store Enable | 0: Transmit history data is not stored in the buffer.<br>1: Transmit history data is stored in the buffer. | R/W |
| 12 to 0 | TMID[28:16] | Transmit Buffer ID Data H | Set standard ID or extended ID.<br>For standard ID, write 0 to these bits. | R/W |

Modify this register when the TMTRM bit in the corresponding TMSTSp register is set to 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

- TMIDE Bit
  This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

- TMRTR Bit
  This bit is used to set the data format of the message to be transmitted from the transmit buffer.

- THLEN Bit
  When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmit messages is stored in the transmit history buffer after transmission is completed.

- TMID[28:16] Bits
  These bits are used to set the ID of the message to be transmitted from the transmit buffer.

### 18.3.69 CANi Transmit Buffer Register pBH (TMPTRp) (i = 0) (p = 0 to 3)

Address TMPTR0L: F0606H, TMPTR0H: F0607H TMPTR1L: F0616H, TMPTR1H: F0617H
TMPTR2L: F0626H, TMPTR2H: F0627H TMPTR3L: F0636H, TMPTR3H: F0637H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMDLC[3:0] | | | | — | — | — | — | TMPTR[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 12 | TMDLC[3:0] | Transmit Buffer DLC Data | b15 b14 b13 b12<br>0 0 0 0 : 0 data bytes<br>0 0 0 1 : 1 data byte<br>0 0 1 0 : 2 data bytes<br>0 0 1 1 : 3 data bytes<br>0 1 0 0 : 4 data bytes<br>0 1 0 1 : 5 data bytes<br>0 1 1 0 : 6 data bytes<br>0 1 1 1 : 7 data bytes<br>1 X X X : 8 data bytes | R/W |
| 11 to 8 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 7 to 0 | TMPTR[7:0] | Transmit Buffer Label Data | Set the label information to be stored in the transmit history buffer. | R/W |

Modify this register when the TMTRM bit in the corresponding TMSTSp register is set to 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

- TMDLC[3:0] Bits
  These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMRTR bit in the TMIDHp register is set to 0 (data frame). If a 9-byte (or more) data length is set, 8 bytes of data is actually transmitted.
  When the TMRTR bit is set to 1 (remote frame), set the data length of messages to be requested.

- TMPTR[7:0] Bits
  When message transmission has been completed, the TMPTR[7:0] value is stored in the transmit history buffer.

### 18.3.70 CANi Transmit Buffer Register pCL (TMDF0p) (i = 0) (p = 0 to 3)

Address TMDF00L: F0608H, TMDF00H: F0609H TMDF01L: F0618H, TMDF01H: F0619H
TMDF02L: F0628H, TMDF02H: F0629H TMDF03L: F0638H, TMDF03H: F0639H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMDB1[7:0] | | | | | | | | TMDB0[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | TMDB1[7:0] | Transmit Buffer Data Byte 1 | Set transmit buffer data. | R/W |
| 7 to 0 | TMDB0[7:0] | Transmit Buffer Data Byte 0 | | R/W |

Modify this register when the TMTRM bit in the corresponding TMSTSp register is set to 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

### 18.3.71 CANi Transmit Buffer Register pCH (TMDF1p) (i = 0) (p = 0 to 3)

Address TMDF10L: F060AH, TMDF10H: F060BH TMDF11L: F061AH, TMDF11H: F061BH
TMDF12L: F062AH, TMDF12H: F062BH TMDF13L: F063AH, TMDF13H: F063BH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMDB3[7:0] | | | | | | | | TMDB2[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | TMDB3[7:0] | Transmit Buffer Data Byte 3 | Set transmit buffer data. | R/W |
| 7 to 0 | TMDB2[7:0] | Transmit Buffer Data Byte 2 | | R/W |

Modify this register when the TMTRM bit in the corresponding TMSTSp register is set to 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

### 18.3.72 CANi Transmit Buffer Register pDL (TMDF2p) (i = 0) (p = 0 to 3)

Address TMDF20L: F060CH, TMDF20H: F060DH TMDF21L: F061CH, TMDF21H: F061DH
TMDF22L: F062CH, TMDF22H: F062DH TMDF23L: F063CH, TMDF23H: F063DH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMDB5[7:0] | | | | | | | | TMDB4[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | TMDB5[7:0] | Transmit Buffer Data Byte 5 | Set transmit buffer data. | R/W |
| 7 to 0 | TMDB4[7:0] | Transmit Buffer Data Byte 4 | | R/W |

Modify this register when the TMTRM bit in the corresponding TMSTSp register is set to 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

### 18.3.73 CANi Transmit Buffer Register pDH (TMDF3p) (i = 0) (p = 0 to 3)

Address TMDF30L: F060EH, TMDF30H: F060FH TMDF31L: F061EH, TMDF31H: F061FH
TMDF32L: F062EH, TMDF32H: F062FH TMDF33L: F063EH, TMDF33H: F063FH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TMDB7[7:0] | | | | | | | | TMDB6[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | TMDB7[7:0] | Transmit Buffer Data Byte 7 | Set transmit buffer data. | R/W |
| 7 to 0 | TMDB6[7:0] | Transmit Buffer Data Byte 6 | | R/W |

Modify this register when the TMTRM bit in the corresponding TMSTSp register is set to 0 (no transmit request is present). If this register is linked to any transmit/receive FIFO buffer, do not write data to this register.

This register can be read/written when the RPAGE bit in the GRWCR register is 1.

### 18.3.74 CANi Transmit History Buffer Control Register (THLCCi) (i = 0)

Address THLCC0L: F037CH, THLCC0H: F037DH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | THLDTE | THLIM | THLIE | — | — | — | — | — | — | — | THLE |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 11 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 10 | THLDTE | Transmit History Target Buffer Select | 0: Entry from transmit/receive FIFO buffers<br>1: Entry from transmit buffers, transmit/receive FIFO buffers | R/W |
| 9 | THLIM | Transmit History Interrupt Source Select | 0: When 6 sets of data have been stored in the transmit history buffer<br>1: When a single set of transmit history data has been stored | R/W |
| 8 | THLIE | Transmit History Interrupt Enable | 0: Transmit history interrupt is disabled.<br>1: Transmit history interrupt is enabled. | R/W |
| 7 to 1 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 0 | THLE | Transmit History Buffer Enable | 0: Transmit history buffer is not used.<br>1: Transmit history buffer is used. | R/W |

- THLDTE Bit
  When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers and transmit/receive FIFO buffers is stored in the transmit history buffer.
  Modify this bit only in channel reset mode.

- THLIM Bit
  This bit is used to select a transmit history interrupt source.
  Modify this bit only in channel reset mode.

- THLIE Bit
  When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit with the THLE bit set to 0.

- THLE Bit
  Setting this bit to 1 makes the transmit history buffer available. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer.
  Modify this bit only in channel communication mode or channel halt mode.

### 18.3.75 CANi Transmit History Buffer Status Register (THLSTSi) (i = 0)

Address THLSTS0L: F0380H, THLSTS0H: F0381H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | THLMC[3:0] | | | | — | — | — | — | THLIF | THLELT | THLFLL | THLEMP |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 12 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 11 to 8 | THLMC[3:0] | Transmit History Buffer Unread Data Counter | These bits indicate the number of unread data sets stored in the transmit history buffer. | R |
| 7 to 4 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 3 | THLIF | Transmit History Interrupt Request Flag | 0: No transmit history interrupt request is present.<br>1: A transmit history interrupt request is present. | R/(W)Note |
| 2 | THLELT | Transmit History Buffer Overflow Flag | 0: Transmit history buffer overflow has not occurred.<br>1: Transmit history buffer overflow has occurred. | R/(W)Note |
| 1 | THLFLL | Transmit History Buffer Full Status Flag | 0: Transmit history buffer is not full.<br>1: Transmit history buffer is full. | R |
| 0 | THLEMP | Transmit History Buffer Empty Status Flag | 0: Transmit history buffer contains unread data.<br>1: Transmit history buffer contains no unread data (buffer empty). | R |

<R> **Note** The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state. To write 0 to this flag bit, write by using an 8-bit data transfer instruction or a 16-bit data transfer instruction.

- THLMC[3:0] Bits
  These bits indicate the number of unread data sets stored in the transmit history buffer.

- THLIF Flag
  The THLIF flag is set to 1 when the interrupt source set by the THLIM bit in the THLCCi register has occurred. This flag is cleared to 0 in channel reset mode or by writing 0 to this flag by the program.

- THLELT Flag
  The THLELT flag is set to 1 when it is attempted to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded. This flag becomes 0 in channel reset mode or by writing 0 to this flag by the program.

- THLFLL Flag
  The THLFLL flag is set to 1 when 8 data sets have been stored in the transmit history buffer, and is cleared to 0 when the number of data sets stored in the transmit history buffer has decreased to less than 8. This bit is also cleared to 0 in channel reset mode or when the THLE bit in the THLCCi register is set to 0 (transmit history buffer is not used).

- THLEMP Flag
  The THLEMP flag is cleared to 0 when even a single set of transmit history data has been stored in the transmit history buffer.
  This flag is set to 1 when all the data in the transmit history buffer has been read. This flag is also set to 1 in channel reset mode or when the THLE bit in the THLCCi register is set to 0 (transmit history buffer is not used).

### 18.3.76 CANi Transmit History Buffer Access Register (THLACCi) (i = 0)

Address THLACC0L: F0680H, THLACC0H: F0681H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TID[7:0] | | | | | | | | — | — | — | BN[1:0] | | — | BT[1:0] | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | TID[7:0] | Label Data | The label information of stored data can be read. | R |
| 7 to 5 | — | Reserved | These bits are always read as 0. | R |
| 4, 3 | BN[1:0] | Buffer Number Data | The buffer number of transmit source (transmit buffer or transmit/receive FIFO) can be read. | R |
| 2 | — | Reserved | This bit is always read as 0. | R |
| 1, 0 | BT[1:0] | Buffer Type Data | b1 b0<br>0 1 : Transmit buffer<br>1 0 : Transmit FIFO buffer | R |

This register can be read when the RPAGE bit in the GRWCR register is 1.

- TID [7:0] Bits
  These bits indicate the label information of transmit history data stored in the transmit history buffer.

- BN[1:0] Bits
  These bits indicate the transmit source buffer number of transmit history data stored in the transmit history buffer.

- BT[1:0] Bits
  These bits indicate the transmit source buffer type of transmit history data stored in the transmit history buffer.

### 18.3.77 CANi Transmit History Buffer Pointer Control Register (THLPCTRi) (i = 0)

Address THLPCTR0L: F0384H, THLPCTR0H: F0385H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | THLPC[7:0] | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 8 | — | Reserved | The write value should always be 0. | R |
| 7 to 0 | THLPC[7:0] | Transmit History Buffer Pointer | Writing H'FF to these bits moves the read pointer to the next unread data in the transmit history buffer. | W |

- THLPC [7:0] Bits
  When the THLPC [7:0] bits are set to H'FF, the read pointer moves to the next data in the transmit history buffer. At this time, the THLMC[3:0] (transmit history buffer unread data counter) value in the THLSTSi register is decremented. After reading the THLACCi register, write H'FF to the THLPC [7:0] bits.
  Write H'FF to the THLPC[7:0] bits when the THLE bit in the THLCCi register is set to 1 (transmit history buffer is used) and the THLEMP flag in the THLSTSi register is 0.

### 18.3.78 CAN Global RAM Window Control Register (GRWCR)

Address GRWCRL: F038AH, GRWCRH: F038BH

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RPAGE |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 1 | — | Reserved | The write value should always be 0. | R |
| 0 | RPAGE | RAM Window Select | 0: Selects window 0 (receive rule entry registers, CAN RAM test registers)<br>1: Selects window 1 (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, transmit history data access register) | R/W |

- RPAGE Bit

  This bit is used to select a window for the switching of registers that are allocated to addresses from H'F03A0 to H'F0681.

  [Registers allocated when the RPAGE bit is set to 0 (window 0 selected)]
  - CAN receive rule entry registers: GAFLIDLj, GAFLIDHj, GAFLMLj, GAFLMHj, GAFLPLj, GAFLPHj
  - CAN RAM test registers: RPGACCr

  [Registers allocated when the RPAGE bit is set to 1 (window 1 selected)]
  - CAN receive buffer registers: RMIDLn, RMIDHn, RMTSn, RMPTRn, RMDF0n to RMDF3n
  - CAN receive FIFO access registers: RFIDLm, RFIDHm, RFTSm, RFPTRm, RFDF0m to RFDF3m
  - CANi transmit/receive FIFO access registers: CFIDLk, CFIDHk, CFTSk, CFPTRk, CFDF0k to CFDF3k
  - CANi transmit buffer registers: TMIDLp, TMIDHp, TMPTRp, TMDF0p to TMDF3p
  - CANi transmit history buffer access register: THLACCi

### 18.3.79 CAN Global Test Configuration Register (GTSTCFG)

Address GTSTCFGL: F038CH, GTSTCFGH: F038DH

| | b15 | B14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | RTMPS[2:0] | | | — | — | — | — | — | — | — | — |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 11 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 10 to 8 | RTMPS[2:0] | RAM Test Page Configuration | Set a value within a range of page 0 (H'00) to page 2 (H'02). | R/W |
| 7 to 0 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |

Modify the GTSTCFG register only in global test mode.

- RTMPS[2:0] Bits
  These bits are used to set the RAM test target page number for RAM test. Set a value from H'00 to H'02.

### 18.3.80 CAN Global Test Control Register (GTSTCTRL)

Address GTSTCTRL: F038EH

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | RTME | — | — |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 7 to 3 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |
| 2 | RTME | RAM Test Enable | 0: RAM test is disabled.<br>1: RAM test is enabled. | R/W |
| 1, 0 | — | Reserved | These bits are always read as 0. The write value should always be 0. | R |

- RTME Bit

  Setting this bit to 1 enables RAM test. Modify this bit only in global test mode.

  (1) Set the GMDC[1:0] bits in the GCTRL register to B'10 (global test mode).

  (2) Release protection by successively writing H'7575 and H'8A8A to the GLOCKK register.

  (3) Set the RTME bit to 1.

  (4) Check that the RTME bit is set to 1.

### 18.3.81 CAN Global Test Protection Unlock Register (GLOCKK)

Address GLOCKK: F0394H

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LOCK[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Symbol | Bit Name | Description | R/W |
|---|---|---|---|---|
| 15 to 0 | LOCK[15:0] | Protection Unlock Data | Write protection unlock data to use test functions. These bits are always read as 0. | W |

Modify the GLOCKK register only in global test mode.

- LOCK[15:0] Bits
  Write the protection unlock data shown in Table 18-4 to the LOCK[15:0] bits in succession to allow writing 1 to the target bit.

**Table 18-4. Protection Unlock Data for Test Functions**

| Test Function | Protection Unlock Data 1 | Protection Unlock Data 2 | Target Bit |
|---|---|---|---|
| RAM test | H'7575 | H'8A8A | RTME bit in the GTSTCTRL register |

Writing data to the CAN's SFR area (H'F0300 to H'F039F) except the RAM area after protection is unlocked enables protection again.

Protection is not enabled even by reading data from the CAN's SFR area or reading/writing data from/to other areas.

### 18.3.82 CAN RAM Test Register r (RPGACCr) (r = 0 to 127)

| Address | | | |
|---|---|---|---|
| | RPGACC0L: F0580H, RPGACC0H: F0581H | RPGACC1L: F0582H, RPGACC1H: F0583H | RPGACC2L: F0584H, RPGACC2H: F0585H |
| | RPGACC3L: F0586H, RPGACC3H: F0587H | RPGACC4L: F0588H, RPGACC4H: F0589H | RPGACC5L: F058AH, RPGACC5H: F058BH |
| | RPGACC6L: F058CH, RPGACC6H: F058DH | RPGACC7L: F058EH, RPGACC7H: F058FH | RPGACC8L: F0590H, RPGACC8H: F0591H |
| | RPGACC9L: F0592H, RPGACC9H: F0593H | RPGACC10L: F0594H, RPGACC10H: F0595H | RPGACC11L: F0596H, RPGACC11H: F0597H |
| | RPGACC12L: F0598H, RPGACC12H: F0599H | RPGACC13L: F059AH, RPGACC13H: F059BH | RPGACC14L: F059CH, RPGACC14H: F059DH |
| | RPGACC15L: F059EH, RPGACC15H: F059FH | RPGACC16L: F05A0H, RPGACC16H: F05A1H | RPGACC17L: F05A2H, RPGACC17H: F05A3H |
| | RPGACC18L: F05A4H, RPGACC18H: F05A5H | RPGACC19L: F05A6H, RPGACC19H: F05A7H | RPGACC20L: F05A8H, RPGACC20H: F05A9H |
| | RPGACC21L: F05AAH, RPGACC21H: F05ABH | RPGACC22L: F05ACH, RPGACC22H: F05ADH | RPGACC23L: F05AEH, RPGACC23H: F05AFH |
| | RPGACC24L: F05B0H, RPGACC24H: F05B1H | RPGACC25L: F05B2H, RPGACC25H: F05B3H | RPGACC26L: F05B4H, RPGACC26H: F05B5H |
| | RPGACC27L: F05B6H, RPGACC27H: F05B7H | RPGACC28L: F05B8H, RPGACC28H: F05B9H | RPGACC29L: F05BAH, RPGACC29H: F05BBH |
| | RPGACC30L: F05BCH, RPGACC30H: F05BDH | RPGACC31L: F05BEH, RPGACC31H: F05BFH | RPGACC32L: F05C0H, RPGACC32H: F05C1H |
| | RPGACC33L: F05C2H, RPGACC33H: F05C3H | RPGACC34L: F05C4H, RPGACC34H: F05C5H | RPGACC35L: F05C6H, RPGACC35H: F05C7H |
| | RPGACC36L: F05C8H, RPGACC36H: F05C9H | RPGACC37L: F05CAH, RPGACC37H: F05CBH | RPGACC38L: F05CCH, RPGACC38H: F05CDH |
| | RPGACC39L: F05CEH, RPGACC39H: F05CFH | RPGACC40L: F05D0H, RPGACC40H: F05D1H | RPGACC41L: F05D2H, RPGACC41H: F05D3H |
| | RPGACC42L: F05A4H, RPGACC42H: F05D5H | RPGACC43L: F05D6H, RPGACC43H: F05D7H | RPGACC44L: F05D8H, RPGACC44H: F05D9H |
| | RPGACC45L: F05DAH, RPGACC45H: F05DBH | RPGACC46L: F05DCH, RPGACC46H: F05DDH | RPGACC47L: F05DEH, RPGACC47H: F05DFH |
| | RPGACC48L: F05E0H, RPGACC48H: F05E1H | RPGACC49L: F05E2H, RPGACC49H: F05E3H | RPGACC50L: F05E4H, RPGACC50H: F05E5H |
| | RPGACC51L: F05E6H, RPGACC51H: F05E7H | RPGACC52L: F05E8H, RPGACC52H: F05E9H | RPGACC53L: F05EAH, RPGACC53H: F05EBH |
| | RPGACC54L: F05ECH, RPGACC54H: F05EDH | RPGACC55L: F05EEH, RPGACC55H: F05EFH | RPGACC56L: F05F0H, RPGACC56H: F05F1H |
| | RPGACC57L: F05F2H, RPGACC57H: F05F3H | RPGACC58L: F05F4H, RPGACC58H: F05F5H | RPGACC59L: F05F6H, RPGACC59H: F05F7H |
| | RPGACC60L: F05F8H, RPGACC60H: F05F9H | RPGACC61L: F05FAH, RPGACC61H: F05FBH | RPGACC62L: F05FCH, RPGACC62H: F05FDH |
| | RPGACC63L: F05FEH, RPGACC63H: F05FFH | RPGACC64L: F0600H, RPGACC64H: F0601H | RPGACC65L: F0602H, RPGACC65H: F0603H |
| | RPGACC66L: F0604H, RPGACC66H: F0605H | RPGACC67L: F0606H, RPGACC67H: F0607H | RPGACC68L: F0608H, RPGACC68H: F0609H |
| | RPGACC69L: F060AH, RPGACC69H: F060BH | RPGACC70L: F060CH, RPGACC70H: F060DH | RPGACC71L: F060EH, RPGACC71H: F060FH |
| | RPGACC72L: F0610H, RPGACC72H: F0611H | RPGACC73L: F0612H, RPGACC73H: F0613H | RPGACC74L: F0614H, RPGACC74H: F0615H |
| | RPGACC75L: F0616H, RPGACC75H: F0617H | RPGACC76L: F0618H, RPGACC76H: F0619H | RPGACC77L: F061AH, RPGACC77H: F061BH |
| | RPGACC78L: F061CH, RPGACC78H: F061DH | RPGACC79L: F061EH, RPGACC79H: F061FH | RPGACC80L: F0620H, RPGACC80H: F0621H |
| | RPGACC81L: F0622H, RPGACC81H: F0623H | RPGACC82L: F0624H, RPGACC82H: F0625H | RPGACC83L: F0626H, RPGACC83H: F0627H |
| | RPGACC84L: F0628H, RPGACC84H: F0629H | RPGACC85L: F062AH, RPGACC85H: F062BH | RPGACC86L: F062CH, RPGACC86H: F062DH |
| | RPGACC87L: F062EH, RPGACC87H: F062FH | RPGACC88L: F0630H, RPGACC88H: F0631H | RPGACC89L: F0632H, RPGACC89H: F0633H |
| | RPGACC90L: F0634H, RPGACC90H: F0635H | RPGACC91L: F0636H, RPGACC91H: F0637H | RPGACC92L: F0638H, RPGACC92H: F0639H |
| | RPGACC93L: F063AH, RPGACC93H: F063BH | RPGACC94L: F063CH, RPGACC94H: F063DH | RPGACC95L: F063EH, RPGACC95H: F063FH |
| | RPGACC96L: F0640H, RPGACC96H: F0641H | RPGACC97L: F0642H, RPGACC97H: F0643H | RPGACC98L: F0644H, RPGACC98H: F0645H |
| | RPGACC99L: F0646H, RPGACC99H: F0647H | RPGACC100L: F0648H, RPGACC100H: F0649H | RPGACC101L: F064AH, RPGACC101H: F064BH |
| | RPGACC102L: F064CH, RPGACC102H: F064DH | RPGACC103L: F064EH, RPGACC103H: F064FH | RPGACC104L: F0650H, RPGACC104H: F0651H |
| | RPGACC105L: F0652H, RPGACC105H: F0653H | RPGACC106L: F0654H, RPGACC106H: F0655H | RPGACC107L: F0656H, RPGACC107H: F0657H |
| | RPGACC108L: F0658H, RPGACC108H: F0659H | RPGACC109L: F065AH, RPGACC109H: F065BH | RPGACC110L: F065CH, RPGACC110H: F065DH |
| | RPGACC111L: F065EH, RPGACC111H: F065FH | RPGACC112L: F0660H, RPGACC112H: F0661H | RPGACC113L: F0662H, RPGACC113H: F0663H |
| | RPGACC114L: F0664H, RPGACC114H: F0665H | RPGACC115L: F0666H, RPGACC115H: F0667H | RPGACC116L: F0668H, RPGACC116H: F0669H |
| | RPGACC117L: F066AH, RPGACC117H: F066BH | RPGACC118L: F066CH, RPGACC118H: F066DH | RPGACC119L: F066EH, RPGACC119H: F066FH |
| | RPGACC120L: F0670H, RPGACC120H: F0671H | RPGACC121L: F0672H, RPGACC121H: F0673H | RPGACC122L: F0674H, RPGACC122H: F0675H |
| | RPGACC123L: F0676H, RPGACC123H: F0677H | RPGACC124L: F0678H, RPGACC124H: F0679H | RPGACC125L: F067AH, RPGACC125H: F067BH |
| | RPGACC126L: F067CH, RPGACC126H: F067DH | RPGACC127L: F067EH, RPGACC127H: F067FH | |

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RDTA[15:0] | | | | | | | | | | | | | | | |
| After Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Description | R/W |
|---|---|
| Data can be read and written in CAN RAM. | R/W |

Modify the RPGACCr register in global test mode with the RTME bit in the GTSTCTRL register set to 1 (RAM test is enabled). The RPGACCr register is readable and writable when the RTME bit is set to 1.

This register can be read/written when the RPAGE bit in the GRWCR register is 0.

## 18.4 CAN Modes

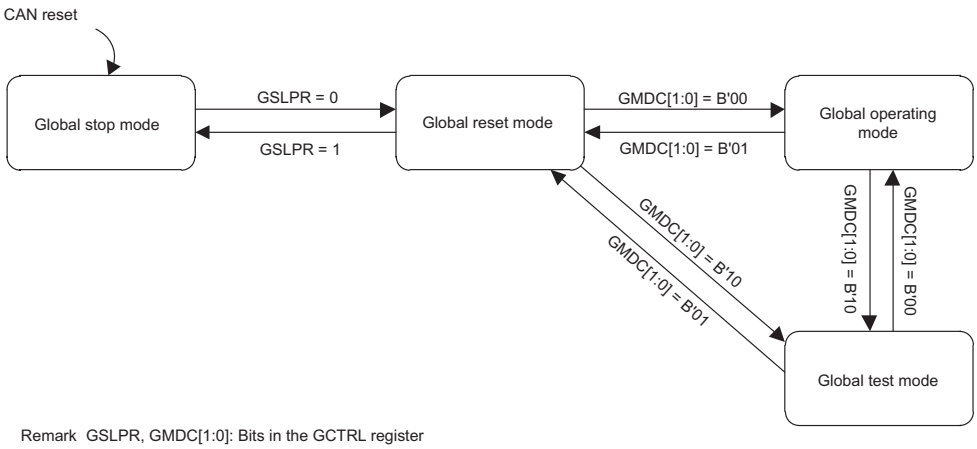
The CAN module has four global modes to control entire CAN module status and four channel modes to control individual channel status. Details of global modes are described in 18.4.1, and details of channel modes are described in 18.4.2.

- Global stop mode : Stops clocks of entire module to achieve low power consumption.
- Global reset mode : Performs initial settings for entire module.
- Global test mode : Performs test settings and performs RAM test.
- Global operating mode : Makes entire module operable.
- Channel stop mode : Stops channel clock.
- Channel reset mode : Performs initial settings for channels.
- Channel halt mode : Stops CAN communication and enables channel test.
- Channel communication mode : Performs CAN communication.

### 18.4.1 Global Modes

Figure 18-2 shows the transitions of global modes.

**Figure 18-2. Transitions of Global Modes**




Remark GSLPR, GMDC[1:0]: Bits in the GCTRL register

Channel modes transition in some cases with transitions of global modes. Table 18-5 shows the transitions of channel modes depending on the global mode setting by the GMDC[1:0] bits and the GSLPR bit.

**Table 18-5. Transitions of Channel Modes Depending on Global Mode Setting (GMDC[1:0] and GSLPR Bits)**

| Channel Mode before Setting | Channel Mode after Setting[Note] | | | |
|---|---|---|---|---|
| | GMDC[1:0] = B'00<br>GSLPR = 0<br>(Global Operation) | GMDC[1:0] = B'10<br>GSLPR = 0<br>(Global Test) | GMDC[1:0] = B'01<br>GSLPR = 0<br>(Global Reset) | GMDC[1:0] = B'01<br>GSLPR = 1<br>(Global Stop) |
| Channel communication | Channel communication | Channel halt | Channel reset | Transition prohibited |
| Channel halt | Channel halt | Channel halt | Channel reset | Transition prohibited |
| Channel reset | Channel reset | Channel reset | Channel reset | Channel stop |
| Channel stop | Channel stop | Channel stop | Channel stop | Channel stop |

**Note** GMDC[1:0], GSLPR: Bits in the GCTRL register.

Table 18-6 shows the global mode transition time.

**Table 18-6. Global Mode Transition Time**

| Mode before Transition | Mode after Transition | Maximum Transition Time |
|---|---|---|
| Global stop | Global reset | 3 $f_{CLK}$ cycles |
| Global reset | Global stop | 3 $f_{CLK}$ cycles |
| Global reset | Global test | 10 $f_{CLK}$ cycles |
| Global reset | Global operating | 10 $f_{CLK}$ cycles |
| Global test | Global reset | 3 $f_{CLK}$ cycles |
| Global test | Global operating | 3 $f_{CLK}$ cycles |
| Global operating | Global reset | 3 $f_{CLK}$ cycles |
| Global operating | Global test | Two CAN frames |

**(1) Global Stop Mode**

In global stop mode, clocks of the CAN do not run and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained.

After the operation of the CAN module is enabled, the CAN module transitions to global stop mode. Setting the GSLPR bit in the GCTRL register to 1 (in global stop mode) in global reset mode sets the CSLPR bit in each of the CiCTRL register to 1 (channel stop mode). If all channels are forcibly caused to transition to channel stop mode, the CAN module transitions to global stop mode. The GSLPR bit should not be modified in global operating mode and global test mode.

**(2) Global Reset Mode**

In global reset mode, CAN module settings are performed. When the CAN module transitions to global reset mode, some registers are initialized. Table 18-9 and Table 18-10 list the registers to be initialized.

Setting the GMDC[1:0] bits in the GCTRL register to B'01 sets the CHMDC[1:0] bits in each of the CiCTRL register to B'01 (channel reset mode). If all channels are forcibly caused to transition to channel reset mode, the CAN module transitions to global reset mode. Channels that are already in channel reset mode or channel stop mode do not transition (because the CHMDC[1:0] bits have already been set to B'01).

**(3) Global Test Mode**

In global test mode, settings for test-related registers are performed. When the CAN module transitions to global test mode, all CAN communications are disabled.

Setting the GMDC[1:0] bits in the GCTRL register to B'10 sets the CHMDC[1:0] bits in each of the CiCTRL register to B'10 (channel halt mode). If all channels are forcibly caused to transition to channel halt mode, the CAN module transitions to global test mode. Channels that are in channel stop mode, channel reset mode, or channel halt mode do not transition.

**(4) Global Operating Mode**

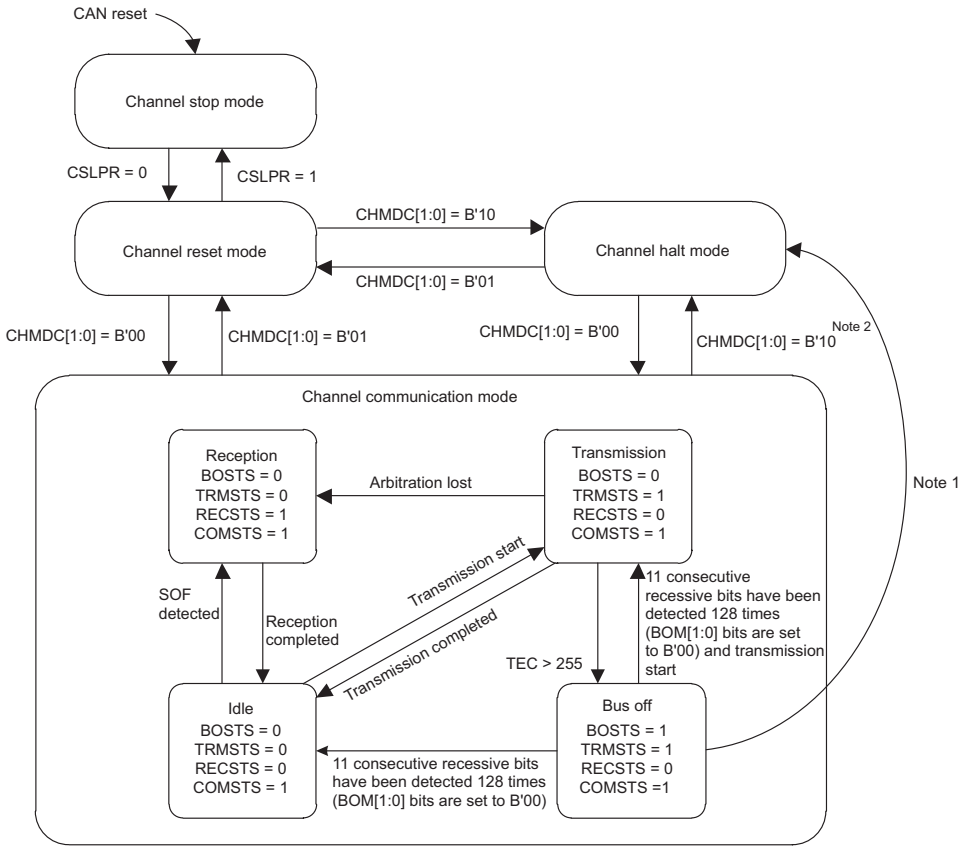
In global operating mode, entire CAN module operates.

When the GMDC[1:0] bits in the GCTRL register are set to B'00, the CAN module transitions to global operating mode.

### 18.4.2 Channel Modes

Figure 18-3 shows a channel mode state transition chart (i = 0). Table 18-7 shows the channel mode transition time (i = 0).

**Figure 18-3. Channel Mode State Transition Chart (i = 0)**




Notes1. Timing of transition from bus off state to channel halt mode
When BOM[1:0] = B'01: Transition to channel halt mode when TEC exceeds 255
When BOM[1:0] = B'10: Transition to channel halt mode when 11 consecutive recessive bits have been detected 128 times
When BOM[1:0] = B'11: Transition to channel halt mode when the CHMDC[1:0] bits are set to B'10
2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode.

Remark CHMDC[1:0], CSLPR: Bits in the CiCTRL register (i = 0)
BOM[1:0]: Bits in the CiCTRH register (i = 0)
BOSTS, TRMSTS, RECSTS, COMSTS: Bits in the CiSTSL register (i = 0)

**Table 18-7. Channel Mode Transition Time (i = 0)**

| Mode before Transition | Mode after Transition | Maximum Transition Time |
|---|---|---|
| Channel stop | Channel reset | 3 $f_{CLK}$ cycles |
| Channel reset | Channel stop | 3 $f_{CLK}$ cycles |
| Channel reset | Channel halt | 3 CANi bit times |
| Channel reset | Channel communication | 2 CANi bit times |
| Channel halt | Channel reset | 3 $f_{CLK}$ cycles |
| Channel halt | Channel communication | 3 CANi bit times |
| Channel communication | Channel reset | 3 $f_{CLK}$ cycles |
| Channel communication | Channel halt | 2 CANi frames |

**(1) Channel Stop Mode**

In channel stop mode, clocks are not supplied to channels and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained.

Each channel enters channel stop mode after the operation of the CAN module is enabled. The channel transitions to channel stop mode when the CSLPR bit in the CiCTRL register is set to 1 (channel stop mode) in channel reset mode. The CSLPR bit should not be modified in channel communication mode and channel halt mode.

**(2) Channel Reset Mode**

In channel reset mode, channel settings are performed. When a channel transitions to channel reset mode, some channel-related registers are initialized. Table 18-9 lists the registers to be initialized.

When the CHMDC[1:0] bits in the CiCTRL register are set to B'01 (channel reset mode) during CAN communication, communication is terminated before it is completed and the channel transitions to channel reset mode. Table 18-8 shows the operation when the CHMDC[1:0] bits are set to B'01 (channel reset mode) during CAN communication.

### (3) Channel Halt Mode

In channel halt mode, settings for test-related registers of channels are performed. When a channel transitions to channel halt mode, CAN communication of the channel stops.

Table 18-8 shows operation when the CHMDC[1:0] bits are set to B'10 (channel halt mode) during CAN communication.

**Table 18-8. Operation when a Channel Transitions to Channel Reset Mode/Channel Halt Mode**

| Mode | During Reception | During Transmission | Bus Off State |
|---|---|---|---|
| Channel reset (CHMDC[1:0] = B'01) | Transitions to channel reset mode before reception is completed. Note 1 | Transitions to channel reset mode before transmission is completed. Note 1 | Transitions to channel reset mode before bus off recovery. |
| Channel halt Note 3 (CHMDC[1:0] = B'10) | Transitions to channel halt mode after reception is completed. Note 2 | Transitions to channel halt mode after transmission is completed. Note 2 | [When BOM[1:0] = B'00]<br>Transitions to channel halt mode (CHMDC[1:0] = B'10) only after bus off recovery.<br>[When BOM[1:0] = B'01]<br>Transitions to channel halt mode automatically when the condition for transition to bus off state is met.<br>[When BOM[1:0] = B'10]<br>Transitions to channel halt mode automatically after bus off recovery.<br>[When BOM[1:0] = B'11]<br>Transitions to channel halt mode immediately after the CHMDC[1:0] bits are set to B'10 before bus off recovery. |

**Notes 1.** To allow transition to channel reset mode after communication is completed, set the CHMDC[1:0] bits to B'10 and confirm that communication has been completed and transition to channel halt mode has been made, and then set the CHMDC[1:0] bits to B'01.

**2.** While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode. The CAN bus status can be confirmed with the BLF flag of the CiERFLL register that becomes 1 when dominant lock is detected.

**3.** In case of a transition from channel reset mode to channel halt mode, transition to channel halt mode after setting the CiCFGL and CiCFGH registers in channel reset mode.

**(4) Channel Communication Mode**

In channel communication mode, CAN communication is performed. Each channel has the following communication states during CAN communication.

- Idle : Neither reception nor transmission is in progress.
- Reception : Receiving a message sent from another node.
- Transmission : Transmitting a message.
- Bus off : Isolated from CAN communication.

When the CHMDC[1:0] bits in the CiCTRL register are set to B'00, the channel transitions to channel communication mode. After that, when 11 consecutive recessive bits have been detected, the COMSTS flag in the CiSTSL register is set to 1 (communication is ready) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

**(5) Bus Off State**

A channel transitions to the bus off state according to the transmit/receive error counter increment/decrement rules of the CAN specifications.

How to return from the bus off state is set by the BOM[1:0] bits in the CiCTRH register.

- When BOM[1:0] = B'00:
  Bus off recovery is compliant with the CAN specifications. After 11 consecutive recessive bits have been detected 128 times, a channel returns from the bus off state to the CAN communication ready state (error active state). At that time, the TEC[7:0] and REC[7:0] bits in the CiSTSH register are initialized to H'00 and the BORF flag in the CiERFLL register is set to 1 (bus off recovery is detected). When the CHMDC[1:0] bits in the CiCTRL register are set to B'10 (channel halt mode) in the bus off state, the channel transitions to channel halt mode after bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times).
- When BOM[1:0] = B'01:
  When a channel transitions to the bus off state, the CHMDC[1:0] bits are set to B'10 and the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to H'00 but the BORF flag is not set to 1.
- When BOM[1:0] = B'10:
  When a channel has transitioned to the bus off state, the CHMDC[1:0] bits are set to B'10. After bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times), the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to H'00 and the BORF flag is set to 1.
- When BOM[1:0] = B'11:
  When the CHMDC[1:0] bits are set to B'10 in the bus off state, the channel transitions to channel halt mode before bus off recovery is completed. At that time, the TEC[7:0] and REC[7:0] bits are initialized to H'00 but the BORF flag is not set to 1.
  However, the BORF flag becomes 1 if a CAN module transitions to error active state (by detecting 128 times of 11 consecutive recessive bits) before CHMDC[1:0] bits are set to B'10.

If the channel transitions to channel halt mode simultaneously when the program writes a value to the CHMDC[1:0] bits, writing by the program takes precedence. An automatic transition to channel halt mode when the BOM[1:0] bits are set to B'01 or B'10 is made only when the CHMDC[1:0] bits are B'00 (channel communication mode). Furthermore, setting the RTBO bit in the CiCTRL register to 1 allows forcible return from the bus off state. As soon as the RTBO bit is set to 1, the state changes to the error active state. After 11 consecutive recessive bits have been detected, the condition of CAN module becomes ready for communication. In this case, the BORF flag is not set to 1 and the TEC[7:0] and REC[7:0] bits are initialized to H'00. Write 1 to the RTBO bit when the BOM[1:0] value is B'00.

**Table 18-9. Registers Initialized in Global Reset Mode or Channel Reset Mode**

| Register | Bit / Flag |
|---|---|
| CiCTRL register | CHMDC[1:0] |
| CiCTRH register | CTMS[1:0], CTME |
| CiSTSL register | CHLTSTS, EPSTS, BOSTS, TRMSTS, RECSTS, COMSTS |
| CiSTSH register | REC[7:0], TEC[7:0] |
| CiERFLL register | ADERR, B0ERR, B1ERR, CERR, AERR, FERR, SERR, ALF, BLF, OVLF, BORF, BOEF, EPF, EWF, BEF |
| CiERFLH register | CRCREG[14:0] |
| CFCCLk register | When transmit/receive FIFO buffer is in transmit mode: CFE |
| CFSTSk register | When transmit/receive FIFO buffer is in transmit mode: CFMC[5:0], CFTXIF, CFRXIF, CFMLT, CFFLL, CFEMP |
| TMCp register | TMOM, TMTAR, TMTR |
| TMSTSp register | TMTARM, TMTRM, TMTRF[1:0], TMTSTS |
| TMTRSTS register | TMTRSTSp |
| TMTCSTS register | TMTCSTSp |
| TMTASTS register | TMTASTSp |
| THLCCi register | THLE |
| THLSTSi register | THLMC[3:0], THLIF, THLELT, THLFLL, THLEMP |
| GTINTSTS register | THIFi, CFTIFi, TAIFi, TSIFi |

**Remark** i = 0, k = 0, p = 0 to 3

**Table 18-10. Registers Initialized Only in Global Reset Mode**

| Register | Bit / Flag |
|---|---|
| GSTS register | GHLTSTS |
| GERFLL register | THLES, MES, DEF |
| GTSC register | TS[15:0] |
| RMND0 register | RMNSn |
| RFCCm register | RFE |
| RFSTSm register | RFMC[5:0], RFIF, RFMLT, RFFLL, RFEMP |
| CFCCLk register | When transmit/receive FIFO buffer is in receive mode: CFE |
| CFSTSk register | When transmit/receive FIFO buffer is in receive mode: CFMC[5:0], CFTXIF, CFRXIF, CFMLT, CFFLL, CFEMP |
| RFMSTS register | RFmMLT |
| CFMSTS register | CFkMLT |
| RFISTS register | RFmIF |
| CFISTS register | CFkIF |
| GTSTCFG register | RTMPS[2:0] |
| GTSTCTRL register | RTME |

**Remark** k = 0, m = 0, 1, n = 0 to 15

## 18.5 Reception Function

There are two reception types.

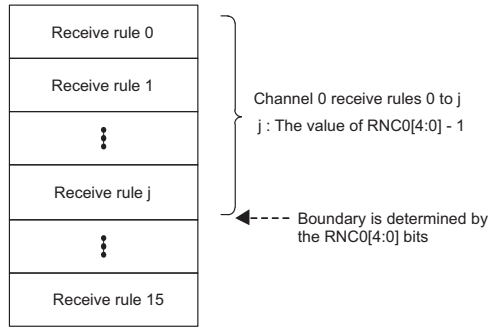
- Reception by receive buffers:
  Zero to 16 receive buffers can be shared by all channels. Since messages stored in receive buffers are overwritten at each reception, the latest receive data can always be read.
- Reception by receive FIFO buffers and transmit/receive FIFO buffers (receive mode):
  Two receive FIFO buffers can be shared by all channels and one dedicated transmit/receive FIFO buffer is provided for each channel. The FIFO buffers can hold the number of received messages set by the RFDC[2:0] bits and CFDC[2:0] bits, and massages can be read sequentially from the oldest.

### 18.5.1 Data Processing Using the Receive Rule Table

Data processing using the receive rule table allows selected messages to be stored in the specified buffer. Data processing includes acceptance filter processing, DLC filter processing, routing processing, label addition processing, and mirror function processing.

Up to 16 receive rules can be registered per channel. If receive rules are not set, no message can be received. Figure 18-4 illustrates how receive rules are registered.

**Figure 18-4. Entry of Receive Rules**




Remark RNC0[4:0]: Bits in the GAFLCFG register

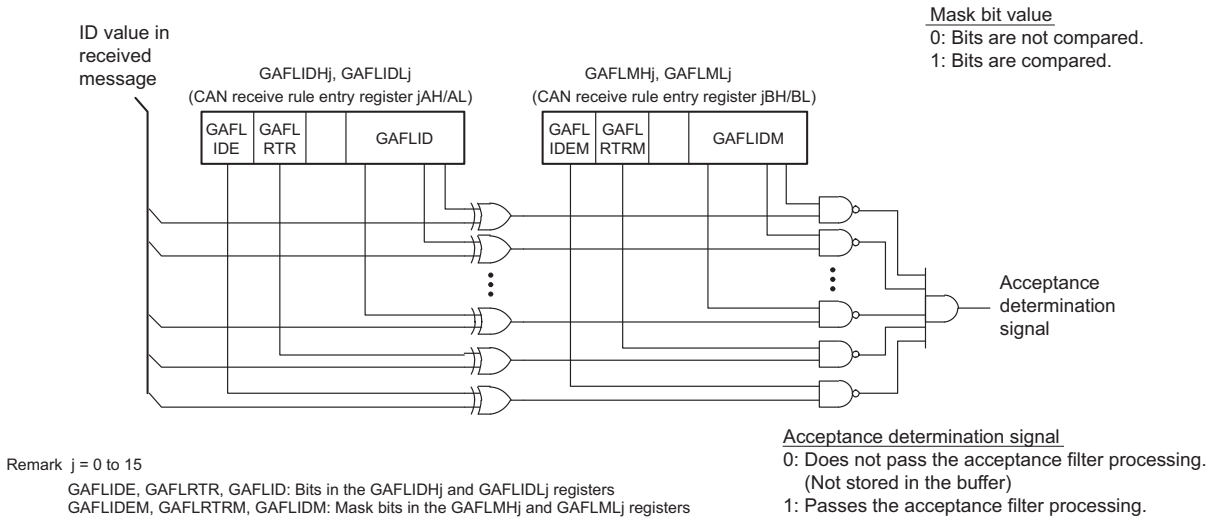
Each receive rule consists of 12 bytes in the GAFLIDLj, GAFLIDHj, GAFLMLj, GAFLMHj, GAFLPLj, and GAFLPHj registers (j = 0 to 15). The GAFLIDLj and GAFLIDHj registers (j = 0 to 15) are used to set ID, IDE bit, RTR bit, and the mirror function, the GAFLMLj and GAFLMHj registers are used to set mask, the GAFLPLj and GAFLPHj registers are used to set label information to be added, DLC value, and storage receive buffer, and storage FIFO buffer.

#### (1) Acceptance Filter Processing

In the acceptance filter processing, the ID data, IDE bit, and RTR bit in a received message are compared with the ID data, IDE bit, and RTR bit set in the receive rule of the corresponding channel. When all these bits match, the message passes through the acceptance filter processing. The ID data, IDE bit, and RTR bit in a received message which correspond to bits that are set to 0 (bits are not compared) in the GAFLMLj and GAFLMHj registers are not compared and are regarded as matched.

Check begins with the receive rule with the smallest rule number of the corresponding channel. When all the bits to be compared in a received message match the bits set in the receive rule or when all the receive rules are compared without any match, filter processing stops. If there is no matching receive rule, the received message is not stored in the receive buffer or FIFO buffer.

**Figure 18-5. Acceptance Filter Function**




#### (2) DLC Filter Processing

When the DCE bit in the GCFGL register is set to 1 (DLC check is enabled), DLC filter processing is added to messages that passed through the acceptance filter processing. When the DLC value in a message is equal to or larger than the DLC value set in the receive rule, the message passes through the DLC filter processing.

When a message has passed through the DLC filter processing with the DRE bit in the GCFGL register set to 0 (DLC replacement is disabled), the DLC value in the received message is stored in the buffer. In this case, all the data bytes in the received message are stored in the buffer.

When a message has passed through the DLC filter processing with the DRE bit in the GCFGL register set to 1 (DLC replacement is enabled), the DLC value in the receive rule is stored in the buffer instead of the DLC value in the received message. In this case, a value of H'00 is written to data bytes that are larger than the DLC value in the receive rule.

When the DLC value in the received message is smaller than that in the receive rule, the message does not pass through the DLC filter processing. In this case, the message is not stored in the receive buffer or the FIFO buffer and the DEF flag in the GERFLL register is set to 1 (a DLC error is present).

#### (3) Routing Processing

Messages that passed through the acceptance filter processing and the DLC filter processing are stored in receive buffers, receive FIFO buffers, or transmit/receive FIFO buffers (set to receive mode). Message storage destination is set by the GAFLRMV, GAFLRMDP[6:0], GAFLFDP4, and GAFLFDP[1:0] bits in the GAFLPLj register (j = 0 to 15). Messages that passed through the acceptance filter processing and the DLC filter processing can be stored in up to two buffers.

#### (4) Label Addition Processing

It is possible to add 12-bit label information to messages that passed through the filter processing and store them in buffers. This label information is set in the GAFLPTR[11:0] bits in the GAFLPHj register.

#### (5) Mirror Function Processing

The mirror function allows reception of messages transmitted from the own CAN node. The mirror function is made available by setting the MME bit in the GCFGL register to 1 (mirror function is enabled).

When the mirror function is in use, receive rules for which the GAFLLB bit in the GAFLIDHj register is set to 0 are used for data processing when receiving messages transmitted from other CAN nodes. When receiving messages transmitted from the own CAN node, receive rules for which the GAFLLB bit is set to 1 are used for data processing.

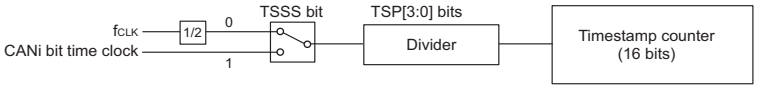
### 18.5.2 Timestamp

The timestamp counter is a 16-bit free-running counter used for recording message receive time. The timestamp counter value is fetched at the start-of-frame (SOF) timing of a message and is then stored in a receive buffer or a FIFO buffer together with the message ID and data. The clock obtained by frequency-dividing $f_{CLK}$ by 2 ($f_{CLK}/2$) or CANi bit time clock is selectable as a timestamp counter clock source from the TSSS bit in the GCFGL register. The clock obtained by dividing the selected clock source by the TSP[3:0] value in the GCFGL register is used as the timestamp counter count source.

When the CANi bit time clock is used as a clock source, the timestamp counter stops when the corresponding channel transitions to channel reset mode or channel halt mode. When the clock obtained by frequency-dividing $f_{CLK}$ by 2 ($f_{CLK}/2$) is used as a clock source, the timestamp function is not affected by channel mode.

The timestamp counter value is reset to H'0000 by setting the TSRST bit in the GCTRH register to 1.

**Figure 18-6. Timestamp Function Block Diagram**




Remark TSSS, TSP[3:0]: Bits in the GCFGL register

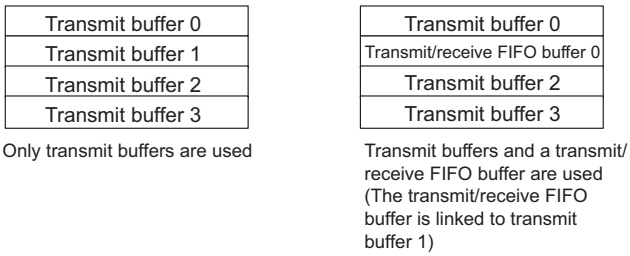
## 18.6 Transmission Functions

There are two types of transmission.

- Transmission using transmit buffers:
  Each channel has 4 buffers.
- Transmission using transmit/receive FIFO buffers (transmit mode):
  Each channel has one FIFO buffer. Up to 16 messages can be contained in a single FIFO buffer. Each FIFO buffer is used with a link to a transmit buffer. Only the message to be transmitted next in a FIFO buffer becomes the target of transmit priority determination. Messages are transmitted sequentially on a first-in, first-out basis.

Figure 18-7 shows the allocation of transmit/receive FIFO buffer link.

**Figure 18-7. Allocation of Transmit/Receive FIFO Buffer Links**




### 18.6.1 Transmit Priority Determination

If transmit requests are issued from multiple buffers in the same channel, transmit priority is determined.

The priority is determined by using one of the following methods.

- ID priority (TPRI bit = 0)
- Transmit buffer number priority (TPRI bit = 1)

The setting of the TPRI bit in the GCFGL register is enabled in all CAN channels.

When the TPRI bit is set to 0, messages are transmitted according to the priority of stored message IDs. Priority of IDs conforms to the CAN bus arbitration specification defined in the CAN specifications. IDs of messages stored in transmit buffers and transmit/receive FIFO buffers (set to transmit mode) are targets of priority determination. When transmit/receive FIFO buffers are used, the oldest message in a FIFO buffer becomes the target of priority determination. When a message is being transmitted from a transmit/receive FIFO buffer, the next message in the FIFO buffer becomes the target of priority determination.

When the TPRI bit is set to 1, the message in the transmit buffer of the minimum number among buffers with a transmit request is transmitted first. When transmit/receive FIFO buffers are linked to transmit buffers, transmit priority is determined according to linked transmit buffer numbers.

When messages are retransmitted due to an arbitration lost or an error, transmit priority determination is made again regardless of the TPRI bit setting.

### 18.6.2 Transmission Using Transmit Buffers

Setting the transmit request bit (TMTR bit in the TMCp register) in a transmit buffer to 1 (transmission is requested) allows transmission of data frames or remote frames.

Transmit result is shown by the TMTRF[1:0] flag in the corresponding TMSTSp register (p = 0 to 3). When transmit completes successfully, the TMTRF[1:0] flag is set to B'10 (transmission has been completed (without transmit abort request)) or B'11 (transmission has been completed (with transmit abort request)).

#### (1) Transmit Abort Function

With respect to transmit buffers for which the TMTRM bit in the TMSTSp register is set to 1 (a transmit request is present), when the TMTAR bit in the TMCp register is set to 1 (transmit abort is requested), the transmit request is canceled. When transmit abort is completed, the TMTRF[1:0] flag in the TMSTSp register is set to B'01 (transmit abort has been completed) and the transmit request is canceled (clearing the TMTRM bit to 0).

A message that is being transmitted or a message to be transmitted next according to the transmit priority determination cannot be aborted. However, when an arbitration lost or an error has occurred while a message for which the TMTAR bit is set to 1 is being transmitted, retransmission is not performed.

#### (2) One-Shot Transmission Function (Retransmission Disabling Function)

When the TMOM bit in the TMCp register is set to 1 (one-shot transmission is enabled), transmission is performed only once. Even if an arbitration lost or an error occurs, retransmission is not performed.

One-shot transmit result is shown by the TMTRF[1:0] flag in the corresponding TMSTSp register. When one-shot transmit completes successfully, the TMTRF[1:0] flag is set to B'10 or B'11. When an arbitration lost or an error has occurred, the TMTRF[1:0] flag is set to B'01 (transmit abort has been completed).

### 18.6.3 Transmission Using FIFO Buffers

Messages of a volume of the FIFO buffer depth set by the CFDC[2:0] bits in the CFCCLk register can be stored in a single transmit/receive FIFO buffer. Messages are transmitted sequentially on a first-in, first-out basis.

Transmit/receive FIFO buffers are linked to transmit buffers selected by the CFTML[1:0] bits in the CFCCHk register. When the CFE bit in the CFCCLk register is set to 1 (transmit/receive FIFO buffers are used), transmit/receive FIFO buffers become targets of transmit priority determination. Priority determination is made for only the message to be transmitted next in a FIFO buffer.

When the CFE bit is set to 0 (no transmit/receive FIFO buffer is used), the CFEMP flag is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) at the timing below.

- The transmit/receive FIFO buffer becomes empty immediately when the message in it is not being transmitted or is not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration lost when the message in it is being transmitted or to be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages cannot be stored in FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

#### (1) Interval Transmission Function

To transmit messages from the same FIFO buffer while a transmit/receive FIFO buffer that is set to transmit mode is in use, message transmission interval time can be set.

Immediately after the first message has been transmitted successfully from the FIFO buffer with the CFE bit in the CFCCLk register set to 1, the interval timer starts counting (after EOF7 of the CAN protocol). After that, when the interval time has passed, the next message is transmitted. The interval timer stops in channel reset mode or by clearing the CFE bit to 0.

The interval time is set by the CFITT[7:0] bits in the CFCCHk register. When the interval timer is not used, set the CFITT[7:0] bits to H'00.

Select an interval timer count source by the CFITR and CFITSS bits in the CFCCHk register. When the CFITR and CFITSS bits are set to B'00, the clock obtained by frequency-dividing $f_{CLK}/2$ by the ITRCP[15:0] value is used as a count source. When the CFITR and CFITSS bits are set to B'10, the clock obtained by frequency-dividing $f_{CLK}/2$ by the ITRCP[15:0] value $\times$ 10 is used as a count source. When the CFITR and CFITSS bits are set to B'x1, the CANi bit time clock is used as a count source.

The interval time is calculated by the following equations where M is the set ITRCP[15:0] value and N is the set CFITT[7:0] value.

1. When CFITR and CFITSS = B’00

$$\frac{1}{f_{CLK}} \times 2 \times M \times N$$

2. When CFITR and CFITSS = B’10

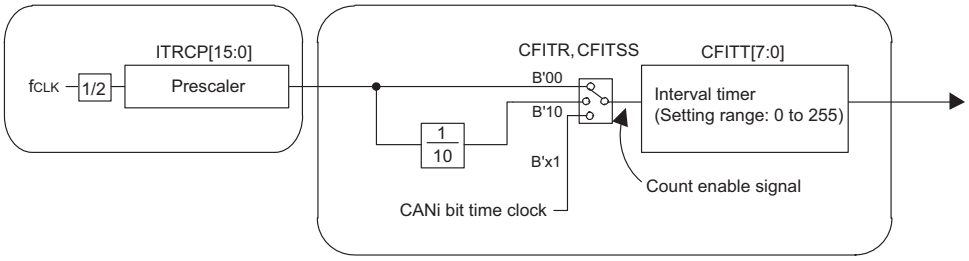
$$\frac{1}{f_{CLK}} \times 2 \times M \times 10 \times N$$

3. When CFITR and CFITSS = B’x1 ($f_{CANBIT}$ is CANi bit time clock frequency)

$$\frac{1}{f_{CANBIT}} \times N$$

Figure 18-8 shows the interval timer block diagram.

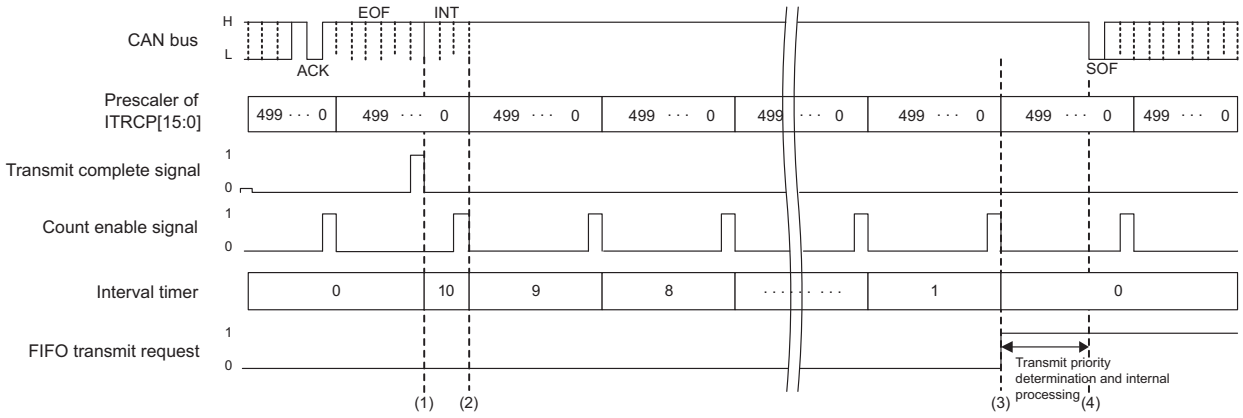
**Figure 18-8. Interval Timer Block Diagram**




Remark ITRCP[15:0]: Bits in the GCFGH register
CFITR, CFITSS, CFITT[7:0]: Bits in the CFCCHk register (k = 0)

Figure 18-9 shows the interval timer timing chart.

**Figure 18-9. Interval Timer Timing Chart**




$$\text{Interval time (logical value)} = \frac{1}{f_{CLK}} \times 2 \times m\ (\text{ITRCP[15:0] value}) \times \text{CFITT[7:0] value}$$

Remark $f_{CLK}$ : CPU/peripheral hardware clock
ITRCP[15:0]: Bits in the GCFGH register (The set value is 500 in this figure.)
CFITT[7:0]: Bits in the CFCCHk register (The set value is 10 in this figure.)

(1) The interval timer starts counting upon completion of transmission. Since the prescaler is not initialized at the time of transmission completion, the first interval time contains an error of up to one count of the interval timer.
(2) The interval timer is decremented by the next count enable signal.
(3) When the interval timer has decreased to 0, the transmit/receive FIFO buffer issues a transmit request.
(4) The transmit/receive FIFO buffer is determined for the next transmission by the priority determination, it starts transmitting data. Transmission starts with a delay of three CANi bit time clock cycles or less from the issue of transmit request.

### 18.6.4 Transmit History Function

Information of transmitted messages can be stored in the transmit history buffer. Each channel has a single transmit history buffer that can contain 8 sets of transmit history data.

A message transmit source buffer type can be selected by the THLDTE bit in the THLCCi register. Whether to store transmit history data for each message can be set by the THLEN bit in the CFIDHk register.

After transmit completes successfully, information of the following transmit messages is stored in the transmit history buffer as transmit history data. After successful completion of transmit, process may be delayed by up to 38 clocks of $f_{CLK}$ before the transmit history data is stored.

- Buffer type B'01: Transmit buffer
  B'10: Transmit/receive FIFO buffer
- Buffer number Number of source transmit buffer or transmit/receive FIFO buffer.
  This number depends on buffer types. See Table 18-11.
- Label data Label information of transmit message

**Table 18-11. Transmit History Data Buffer Numbers**

| Buffer type / Buffer No. | B'01 | B'10 |
|---|---|---|
| B'00 | Transmit buffer 0 | Numbers of transmit buffers linked to transmit/receive FIFO buffers by the CFTML[1:0] bits in the CFCCHk register. |
| B'01 | Transmit buffer 1 | |
| B'10 | Transmit buffer 2 | |
| B'11 | Transmit buffer 3 | |

Label data is used to identify each message. A unique label data can be added to each message transmitted from a transmit buffer or transmit/receive FIFO buffer.

Transmit history data can be read from the THLACCi register. If it is attempted to store new transmit history data while the buffer is full, the buffer overflows and the new data is discarded.

## 18.7 Test Function

The test function is classified into communication tests and global tests.

Communication tests: Performed for each channel.

- Standard test mode
- Listen-only mode
- Self-test mode 0 (external loopback mode)
- Self-test mode 1 (internal loopback mode)

Global tests: Performed in entire module

- RAM test (read/write test)

### 18.7.1 Standard Test Mode

Standard test mode allows CRC test.

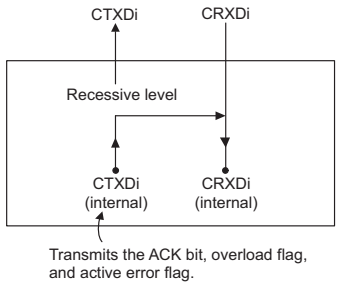
### 18.7.2 Listen-Only Mode

Listen-only mode allows reception of data frames and remote frames. Only recessive bits are transmitted on the CAN bus, and the ACK bit, overload flag, and active error flag are not transmitted. Listen-only mode is available for detecting the communication speed.

Do not make a transmit request from any buffer in listen-only mode.

Figure 18-10 shows the connection when listen-only mode is selected.

**Figure 18-10. Connection when Listen-Only Mode is Selected**




Remark i = 0

### 18.7.3 Self-Test Mode (Loopback Mode)

In self-test mode, transmitted messages are compared with the receive rule of the own channel and the messages are stored in a buffer if they have passed through the filter processing. Messages transmitted from other CAN nodes are compared only with the receive rule for which the GAFLLB bit in the GAFLIDHj register (j = 0 to 15) is set to 0 (when a message transmitted from another CAN node is received).

If the mirror function and self-test mode are both enabled, the self-test mode setting takes precedence.

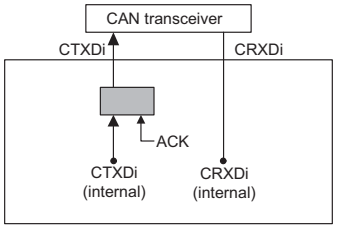
#### (1) Self-Test Mode 0 (External Loopback Mode)

Self-test mode 0 is used to perform a loopback test within a channel including the CAN transceiver.

In self-test mode 0, transmitted messages are handled as messages received through the CAN transceiver and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

Figure 18-11 shows the connection when self-test mode 0 is selected.

**Figure 18-11. Connection when Self-Test Mode 0 is Selected**




Remark i = 0

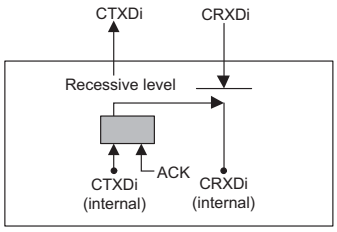
#### (2) Self-Test Mode 1 (Internal Loopback Mode)

In self-test mode 1, transmitted messages are handled as received messages and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

In self-test mode 1, internal feedback from the internal CTXDi pin to the internal CRXDi pin is performed. The external CRXDi pin input is isolated. The external CTXDi pin outputs only recessive bits.

Figure 18-12 shows the connection when self-test mode 1 is selected.

**Figure 18-12. Connection when Self-Test Mode 1 is Selected**




Remark i = 0

### 18.7.4 RAM Test

The RAM test function allows accesses to all CAN RAM addresses.

When the RAM test function is used, the RAM is divided into pages of 256 bytes each. RAM test page is set by the RTMPS[2:0] bits in the GTSTCFG register. Data in the set page can be read from and written to the RPGACCr register (r = 0 to 127). The available total RAM size is 544 bytes (H'0220).

## 18.8 Interrupt

The CAN module has 6 interrupts that are grouped into global interrupts and channel interrupts.

Global interrupts (2 interrupts):

- CAN global receive FIFO interrupt
- CAN global error interrupt

Channel interrupts (4 interrupts per channel):

- CANi channel transmit interrupt
  CANi transmit complete interrupt
  CANi transmit abort interrupt
  CANi transmit/receive FIFO transmit complete interrupt (transmit mode)
  CANi transmit history interrupt
- CANi transmit/receive FIFO receive interrupt
- CANi channel error interrupt
- CANi wakeup interrupt

- CAN interrupt except CANi wakeup interrupt
  When an interrupt request is generated, the corresponding CAN module interrupt request flag is set to 1 (interrupt request present). In that case, when the interrupt enable bit is set to 1 (enabling interrupts), an interrupt request is output from the CAN module. (Generation of interrupts also is controlled by an interrupt functions.)
  Setting the interrupt request flag to 0 (no interrupt request present) or setting the interrupt enable bit to 0 (disabling interrupts) clears the current interrupt request. The next interrupt request is not generated until the interrupt request is cleared.

- CANi wakeup interrupt
  The CANi wakeup interrupt is generated in every mode when a falling edge in the CRXDi pin is detected. When the CANi wakeup interrupt is used, set the function of the corresponding port to CRXDi.
  The CANi wakeup interrupt is controlled by the interrupt function.

For details on the setting of the interrupt functions, refer to **CHAPTER 21 INTERRUPT FUNCTIONS**.

Table 18-12 lists the CAN interrupt sources. Figure 18-13 shows the CAN global interrupt block diagram. Figure 18-14 shows the CAN channel interrupt block diagram.

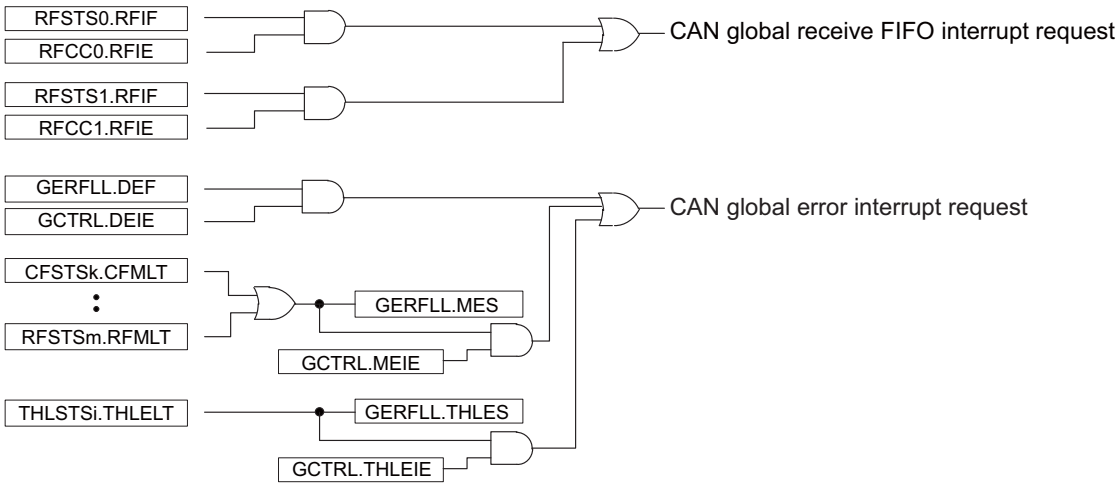
**Table 18-12. List of CAN Interrupt Sources**

| Item | Interrupt Source | | Corresponding Interrupt Request Flag[Note] | Corresponding Interrupt Enable Bit[Note] |
|---|---|---|---|---|
| Global interrupts | CAN global receive FIFO | Receive FIFO 0 | RFIF in the RFSTS0 register | RFIE in the RFCC0 register |
| | | Receive FIFO 1 | RFIF in the RFSTS1 register | RFIE in the RFCC1 register |
| | CAN global error | | DEF in the GERFLL register | DEIE in the GCTRL register |
| | | | MES in the GERFLL register | MEIE in the GCTRL register |
| | | | THLES in the GERFLL register | THLEIE in the GCTRL register |
| Channel interrupts | CANi channel transmit | CANi transmit complete | TMTRF[1:0] in the TMSTSp register | TMIEp in the TMIEC register |
| | | CANi transmit abort | TMTRF[1:0] in the TMSTSp register | TAIE in the CiCTRH register |
| | | CANi transmit/receive FIFO transmit | CFTXIF in the CFSTSk register | CFTXIE in the CFCCLk register |
| | | CANi transmit history | THLIF in the THLSTSi register | THLIE in the THLCCi register |
| | CANi transmit/receive FIFO receive | | CFRXIF in the CFSTSk register | CFRXIE in the CFCCLk register |
| | CANi channel error | | BEF in the CiERFLL register | BEIE in the CiCTRL register |
| | | | ALF in the CiERFLL register | ALIE in the CiCTRL register |
| | | | BLF in the CiERFLL register | BLIE in the CiCTRL register |
| | | | OVLF in the CiERFLL register | OLIE in the CiCTRL register |
| | | | BORF in the CiERFLL register | BORIE in the CiCTRL register |
| | | | BOEF in the CiERFLL register | BOEIE in the CiCTRL register |
| | | | EPF in the CiERFLL register | EPIE in the CiCTRL register |
| | | | EWF in the CiERFLL register | EWIE in the CiCTRL register |
| | CANi wakeup | | None | None |

**Note** For details on the interrupt request flags and interrupt enable bits, refer to **CHAPTER 21 INTERRUPT FUNCTIONS**.

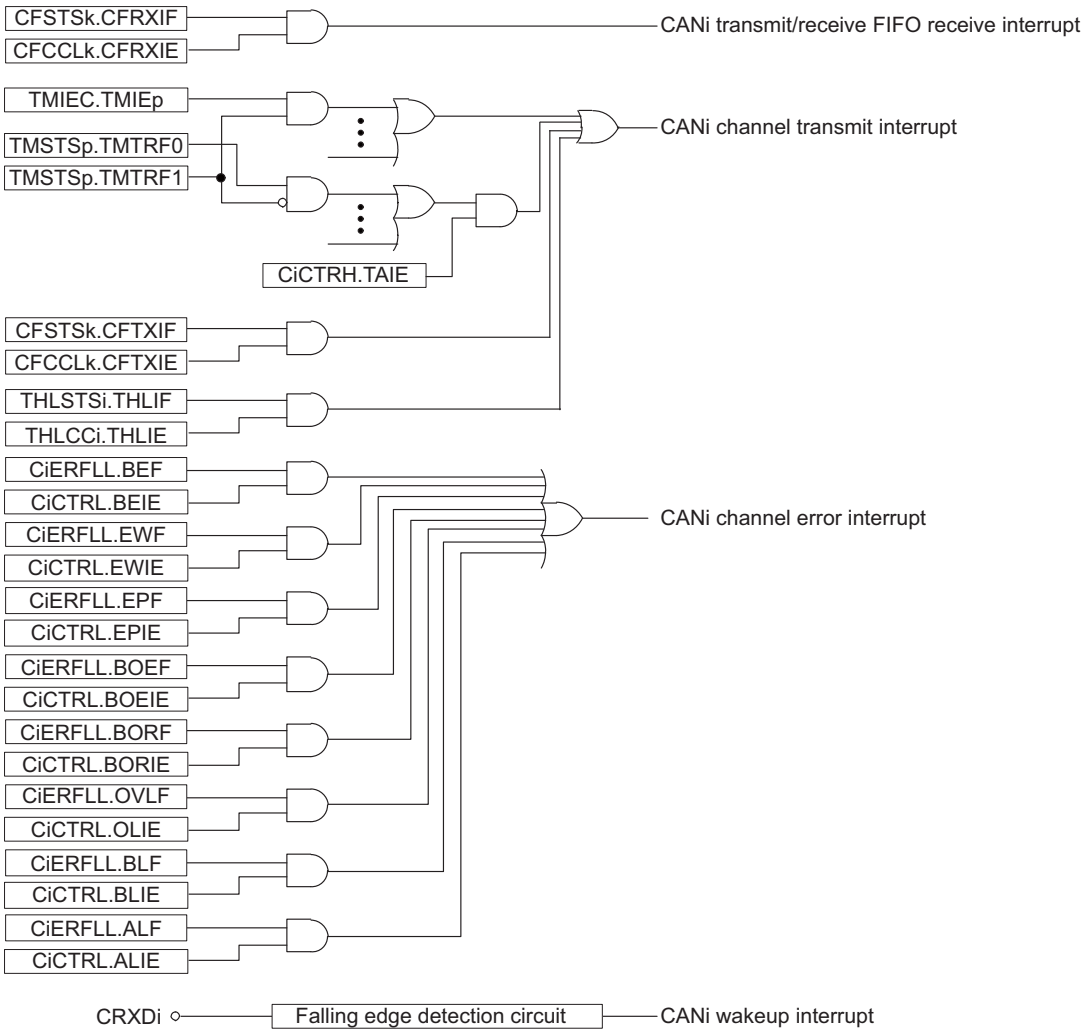
i = 0, k = 0, p = 0 to 3

**Figure 18-13. CAN Global Interrupt Block Diagram**




Remark i = 0
k = 0
m = 0, 1

**Figure 18-14. CAN Channel Interrupt Block Diagram**




Remark i = 0
k = 0
p = 0 to 3

## 18.9 RAM Window

The CAN area from H'F03A0 to H'F0681 consists of two windows. The RPAGE bit in the GRWCR register is used to switch the allocation of registers.

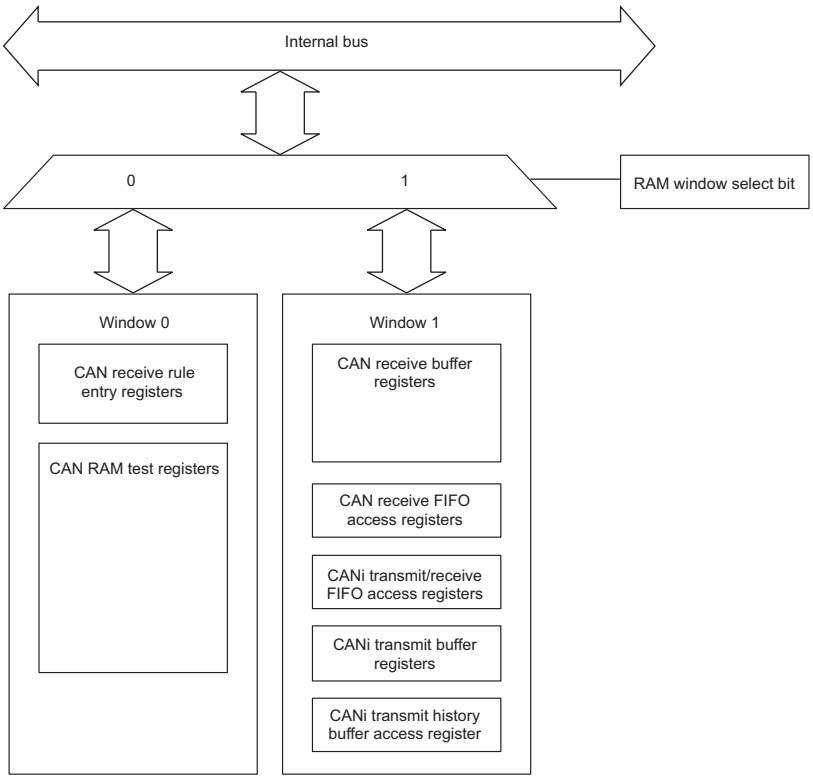
**[Registers allocated when the RPAGE bit is set to 0 (window 0 selected)]**

- CAN receive rule entry registers: GAFLIDLj, GAFLIDHj, GAFLMLj, GAFLMHj, GAFLPLj, GAFLPHj
- CAN RAM test registers: RPGACCr

**[Registers allocated when the RPAGE bit is set to 1 (window 1 selected)]**

- CAN receive buffer registers: RMIDLn, RMIDHn, RMTSn, RMPTRn, RMDF0n to RMDF3n
- CAN receive FIFO access registers: RFIDLm, RFIDHm, RFTSm, RFPTRm, RFDF0m to RFDF3m
- CANi transmit/receive FIFO access registers: CFIDLk, CFIDHk, CFTSk, CFPTRk, CFDF0k to CFDF3k
- CANi transmit buffer registers: TMIDLp, TMIDHp, TMPTRp, TMDF0p to TMDF3p
- CANi transmit history buffer access register: THLACCi

**Figure 18-15. RAM Window**

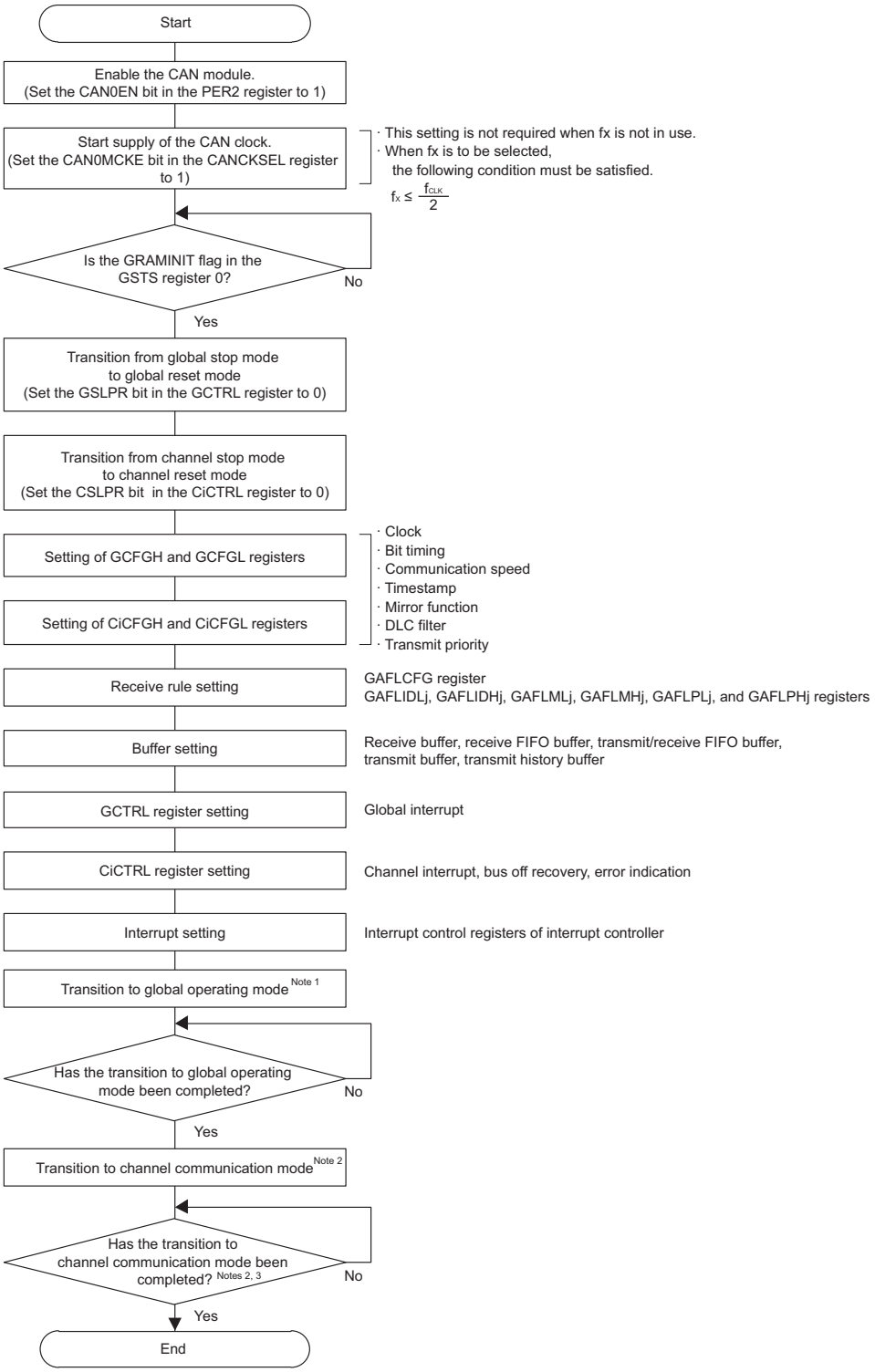


## 18.10 Initial Settings

The CAN module initializes the CAN RAM after the operation of the CAN module is enabled. The RAM initialization time is 276 cycles of $f_{CLK}$. The GRAMINIT flag in the GSTS register is set to 1 (CAN RAM initialization is ongoing) during the RAM initialization and is cleared to 0 (CAN RAM initialization is finished) when the initialization is completed. Make CAN settings after the GRAMINIT flag is cleared to 0.

Figure 18-16 shows the CAN setting procedure after the operation of the CAN module is enabled.

**Figure 18-16. CAN Setting Procedure after the Operation of the CAN Module is Enabled**




Notes 1. If the high-speed system clock ($f_{MX}$) is to be selected as the LIN communications clock source, and the highspeed on-chip oscillator clock ($f_{IH}$) or the PLL clock with its source as the high-speed on-chip oscillator clock is to be selected as the source of the clock signal for $f_{CLK}$, make sure that the condition (LIN communications clock source) < $f_{CLK}$ is satisfied.

2. If the high-speed system clock is to be selected as $f_{CLK}$, do not select $f_X$ as $f_{CAN}$.

Remark i = 0
j = 0 to 15

### 18.10.1 Clock Setting

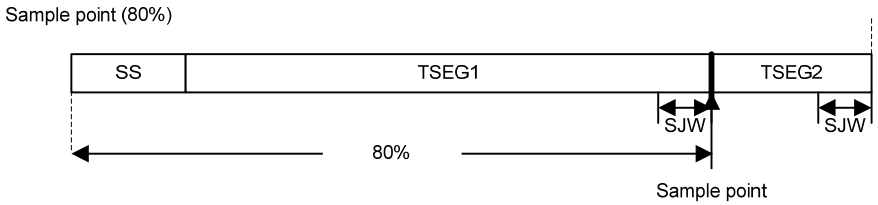
Set the CAN clock ($f_{CAN}$) as a clock source of the CAN module. Select the clock obtained by frequency-dividing $f_{CLK}$ by 2 ($f_{CLK}/2$) or the X1 clock (fx) with the DCS bit in the GCFGL register.

### 18.10.2 Bit Timing Setting

In the CAN protocol, one bit of a communication frame consists of three segments, SS, TSEG1, and TSEG2. Two of the segments, TSEG1 and TSEG2, can be set by the CiCFGH register for each channel. Sample point timing can be determined by setting two segments. This timing can be adjusted in units of 1 Time Quantum (referred to as Tq hereinafter). 1 Tq equals to one CANi Tq clock cycle. The CANi Tq clock is obtained by selecting the clock source with the DCS bit in the GCFGL register and selecting the clock division ratio with the BRP[9:0] bits in the CiCFGL register.

Figure 18-17 shows the bit timing chart. Table 18-13 shows an example of bit timing setting.

**Figure 18-17. Bit Timing Chart**




Remark SS = 1 Tq fixed
Set TSEG1 to a range of 4 Tq to 16 Tq.
Set TSEG2 to a range of 2 Tq to 8 Tq.
Set SJW to a range of 1 Tq to 4 Tq.
Set SS + TSEG1 + TSEG2 to a range of 8 Tq to 25 Tq.
TSEG1 > TSEG2 ≥ SJW

- SS (synchronization segment):
  The SS is a segment that performs synchronization by monitoring the edge from recessive to dominant bits in the Interframe Space.
  Interframe Space consists of Intermission, Suspend Transmission, and Bus Idle. All nodes can start transmission during Bus Idle.
- TSEG1 (time segment 1):
  TSEG1 is a segment that absorbs physical delay on the CAN network. The physical delay on the CAN network is twice of the total of the delay on the CAN bus, the delay in the input comparator, and the delay in the output driver.
- TSEG2 (time segment 2):
  TSEG2 is a segment that compensates phase error due to an error in frequency.
- SJW (resynchronization jump width)
  The SJW is a length to extend or reduce time segment to compensate an error in phase due to phase error.

**Table 18-13. Example of Bit Timing Setting**

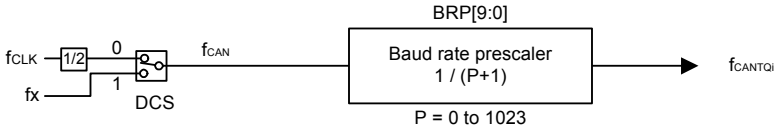
| 1 Bit | Set Value (Tq) | | | | Sample Point (%) Note: See Figure 18-17. |
|---|---|---|---|---|---|
| | SS | TSEG1 | TSEG2 | SJW | |
| 8 Tq | 1 | 4 | 3 | 1 | 62.50 |
| | 1 | 5 | 2 | 1 | 75.00 |
| 10 Tq | 1 | 6 | 3 | 1 | 70.00 |
| | 1 | 7 | 2 | 1 | 80.00 |
| 16 Tq | 1 | 10 | 5 | 1 | 68.75 |
| | 1 | 11 | 4 | 1 | 75.00 |
| 20 Tq | 1 | 13 | 6 | 1 | 70.00 |
| | 1 | 15 | 4 | 3 | 80.00 |
| 24 Tq | 1 | 15 | 8 | 1 | 66.67 |
| | 1 | 16 | 7 | 1 | 70.83 |

### 18.10.3 Communication Speed Setting

Set the CAN communication speed for each channel using the $f_{CAN}$, baud rate prescaler division value (BRP[9:0] bits in the CiCFGL register), and Tq count per bit time.

Figure 18-18 shows the CAN clock control block diagram, and Table 18-14 shows an example of the communication speed setting.

**Figure 18-18. CAN Clock Control Block Diagram**




$$\text{Communication speed} = \frac{f_{CAN}}{\text{Baud rate prescaler division value} \times (\text{Tq count of 1 bit time})}$$

Caution When fx is to be selected, the following condition must be satisfied.

$$f_X \leq \frac{f_{CLK}}{2}$$ Notes 1, 2

Notes 1. If the high-speed on-chip oscillator clock ($f_{IH}$) or the PLL clock with its source as the high-speed on-chip oscillator clock is to be selected as the source of the clock signal for $f_{CLK}$, make sure that the condition $f_X < f_{CLK}/2$ is satisfied.

2. If the high-speed system clock is to be selected as $f_{CLK}$, do not select $f_X$ as $f_{CAN}$.

Remark i = 0
DCS: Bit in the GCFGL register
BRP[9:0]: Bits in the CiCFGL register
$f_{CAN}$: CAN clock
$f_{CANTQi}$: CANi Tq clock

**Table 18-14. Example of Communication Speed Setting**

| $f_{CAN}$ Note / Communication Speed | 16 MHz | 8 MHz |
|---|---|---|
| 1 Mbps | 8 Tq (2)<br>16 Tq (1) | 8 Tq (1) |
| 500 Kbps | 8 Tq (4)<br>16 Tq (2) | 8 Tq (2)<br>16 Tq (1) |
| 250 Kbps | 8 Tq (8)<br>16 Tq (4) | 8 Tq (4)<br>16 Tq (2) |
| 83.3 Kbps | 8 Tq (24)<br>16 Tq (12) | 8 Tq (12)<br>16 Tq (6) |
| 33.3 Kbps | 8 Tq (60)<br>10 Tq (48)<br>16 Tq (30)<br>20 Tq (24) | 8 Tq (30)<br>10 Tq (24)<br>16 Tq (15)<br>20 Tq (12) |

**Note** Values in ( ) are baud rate prescaler division values.

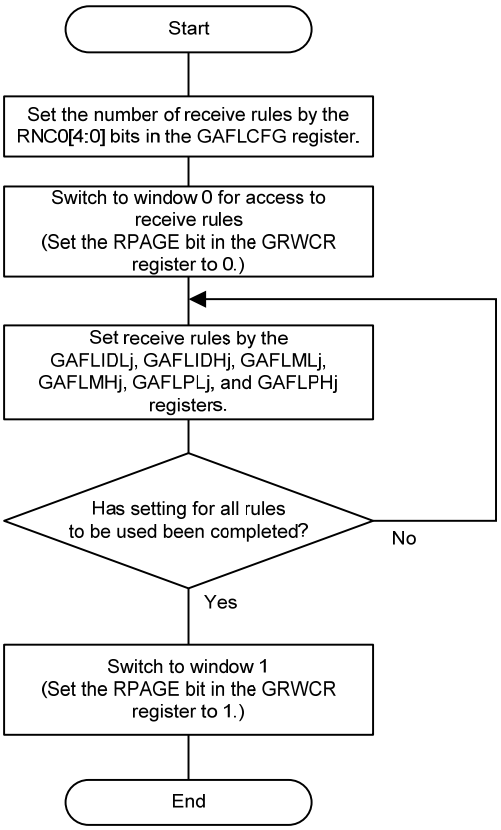
### 18.10.4 Receive Rule Setting

Receive rules can be set using receive rule-related registers.

Up to 16 receive rules can be registered.

Figure 18-19 shows the receive rule setting procedure.

**Figure 18-19. Receive Rule Setting Procedure**

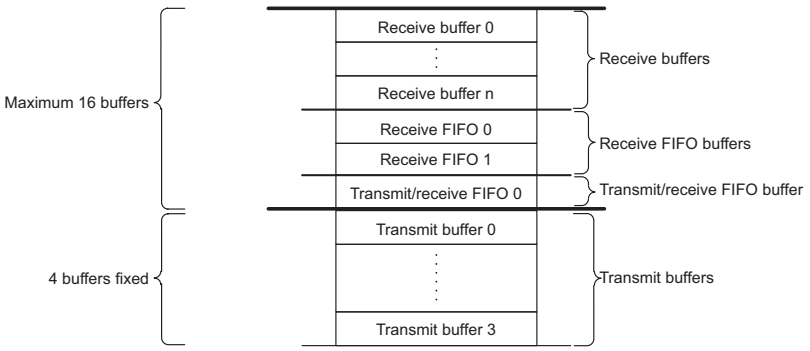


### 18.10.5 Buffer Setting

Set sizes and interrupt sources of buffers. For transmit/receive FIFO buffers that are set to transmit mode, set transmit buffers to be linked.

Figure 18-20 shows the buffer configuration. Figure 18-21 shows the buffer setting procedure.

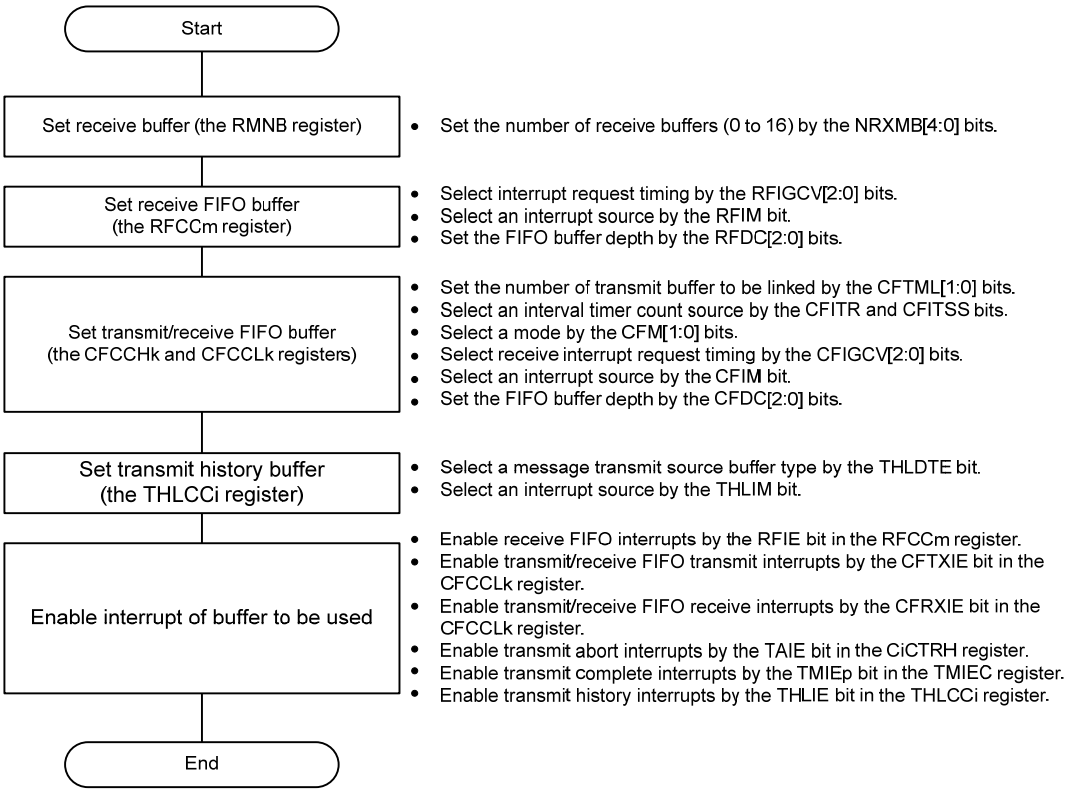
**Figure 18-20. Buffer Configuration**




Caution Receive buffers, receive FIFO buffers, transmit/receive FIFO buffer, and transmit buffers are located in succession.

Remark n = 0 to 15

**Figure 18-21. Buffer Setting Procedure**



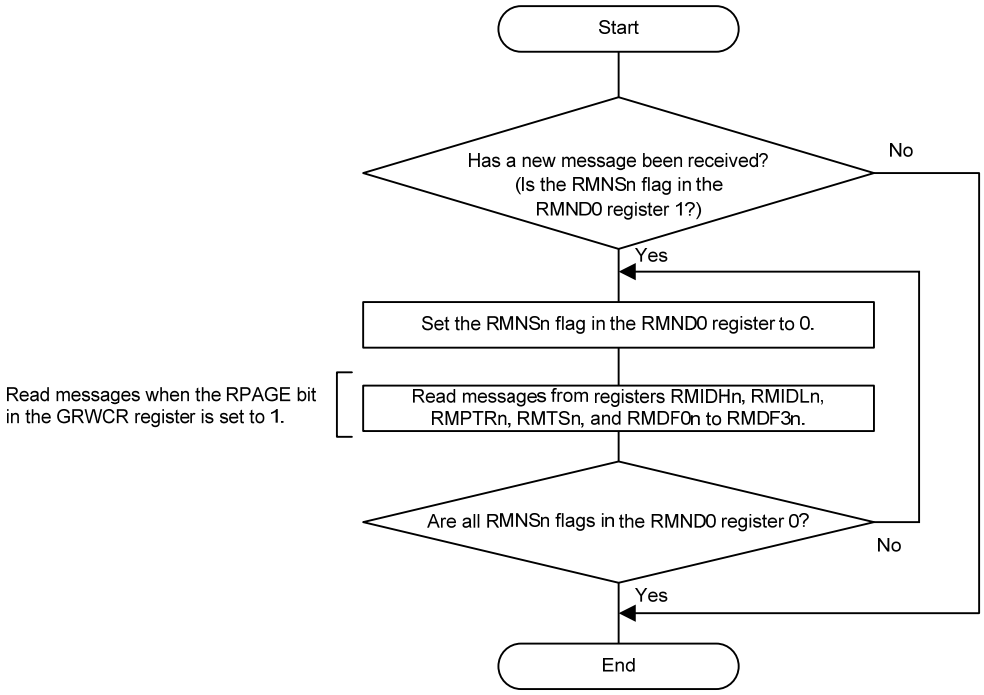

Remark i = 0
k = 0
m = 0, 1
p = 0 to 3

## 18.11 Reception Procedure

### 18.11.1 Receive Buffer Reading Procedure

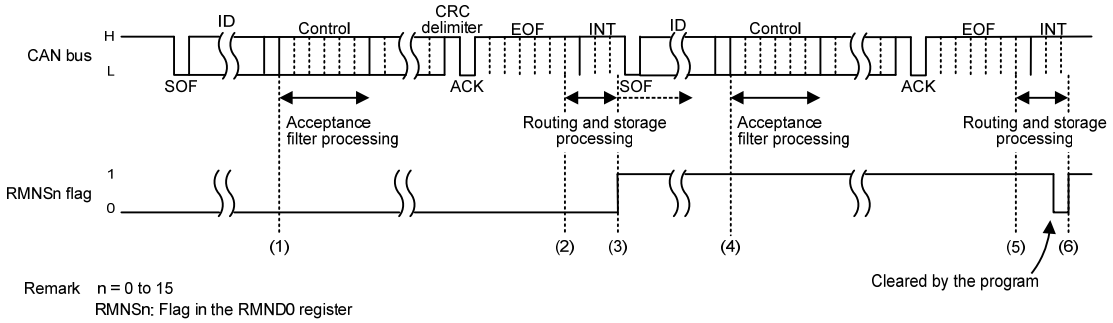
When the processing to store received messages in a receive buffer starts, the RMNSn flag (n = 0 to 15) in the RMND0 register is set to 1 (receive buffer n contains a new message). Messages can be read from the RMIDLn, RMIDHn, RMTSn, RMPTRn, and RMDF0n to RMDF3n registers. If the next message has been received before the current message is read from the receive buffer, the message is overwritten. Figure 18-22 shows the receive buffer reading procedure.

**Figure 18-22. Receive Buffer Reading Procedure**




Remark n = 0 to 15

**Figure 18-23. Receive Buffer Reception Timing Chart**




(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the GCFGL register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing, the processing to store the message in the specified receive buffer starts.
When the message storage processing starts, the corresponding RMNSn flag in the RMND0 register is set to 1 (receive buffer n contains a new message (n = 0 to 15)). If other channels are performing filter processing or transmit priority determination processing, the routing processing and the storage processing may be delayed.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the GCFGL register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(6) When the corresponding RMNSn flag is cleared to 0 (receive buffer n contains no new message (n = 0 to 15)), this flag is set to 1 again when the message storage processing starts. Even if the RMNSn flag remains 1, a new message is overwritten to the receive buffer. The RMNSn flag should not be cleared to 0 during storage of messages.

### 18.11.2 FIFO Buffer Reading Procedure

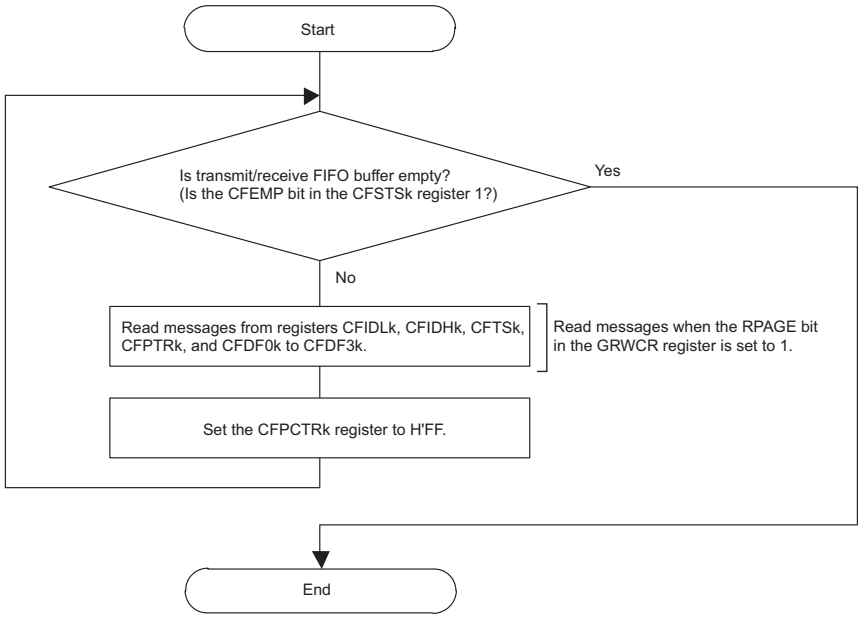
When received messages have been stored in one or more receive FIFO buffers or a transmit/receive FIFO buffer that is set to receive mode, the corresponding message count display counter (RFMC[5:0] bits in the RFSTSm register (m = 0, 1) or CFMC[5:0] bits in the CFSTSk register) is incremented. At this time, when the RFIE bit (receive FIFO interrupt is enabled) in the RFCCm register or the CFRXIE bit (transmit/receive FIFO receive interrupt is enabled) in the CFCCLk register is set to 1, an interrupt request is generated. Received messages can be read from the RFIDLm, RFIDHm, RFTSm, RFPTRm, and RFDF0m to RFDF3m registers (receive FIFO buffers) or the CFIDLk, CFIDHk, CFTSk, CFPTRk, and CFDF0k to CFDF3k registers (transmit/receive FIFO buffers). Messages in FIFO buffers can be read sequentially on a first-in, first-out basis.

When the message count display counter value matches the FIFO buffer depth (a value set by the RFDC[2:0] bits in the RFCCm register or the CFDC[2:0] bits in the CFCCLk register), the RFFLL or CFFLL flag is set to 1 (the receive FIFO buffer is full).

When all messages have been read out of the FIFO buffer, the RFEMP flag in the RFSTSm register or the CFEMP flag in the CFSTSk register is set to 1 (the receive FIFO buffer contains no unread message (buffer empty)).

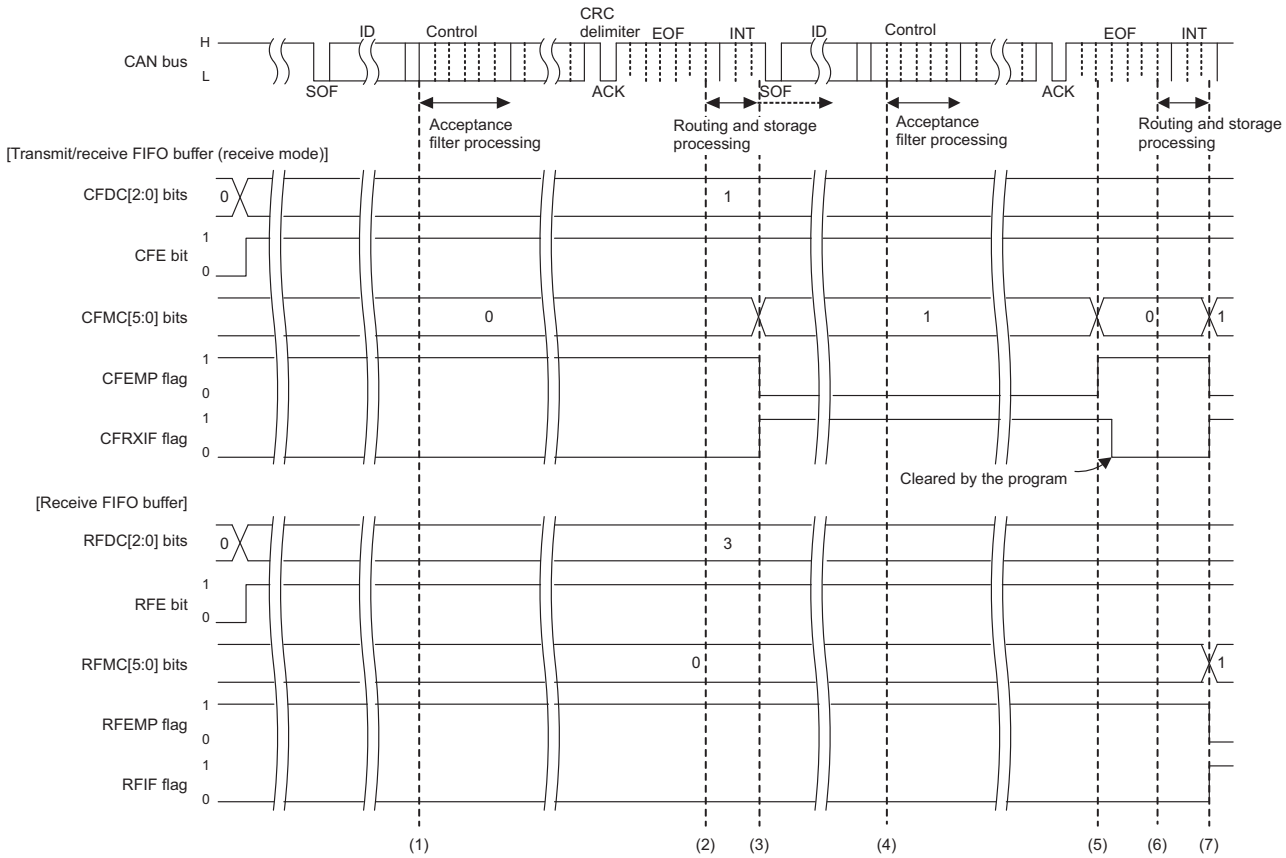
If the RFE bit or the CFE bit is cleared to 0 (no receive FIFO buffer is used) with the interrupt request flag (RFIF flag in the RFSTSm register or CFRXIF flag in the CFSTSk register) set to 1 (a receive FIFO interrupt request is present), the interrupt request flag is not automatically cleared to 0. Clear the interrupt request flag to 0 by the program.

**Figure 18-24. Transmit/Receive FIFO Buffer Reading Procedure**




Remark k = 0

**Figure 18-25. FIFO Buffer Reception Timing Chart**




Remark k = 0, m = 0, 1
CFDC[2:0], CFE: Bits in the CFCCLk register
CFMC[5:0], CFEMP, CFRXIF: Flags in the CFSTSk register
RFDC[2:0], RFE: Bits in the RFCCm register
RFMC[5:0], RFEMP, RFIF: Flags in the RFSTSm register

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the GCFGL register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing and the CFE value in the CFCCLk register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the CFCCLk register is B'001 or more, the message is stored in the transmit/receive FIFO buffer that is set to receive mode. The CFMC[5:0] value in the CFSTSk register is incremented and becomes H'01. When the CFIM bit in the CFCCLk register is set to 1 (a FIFO receive interrupt request is generated each time a message has been received), the CFRXIF flag in the CFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). The CFRXIF flag can be reset to 0 by the program.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) Read received messages from the CFIDLk, CFIDHk, CFTSk, CFPTRk, and CFDF0k to CFDF3k registers and write H'FF to the CFPCTRk register. Thereby the CFMC[5:0] bits in the CFSTSk register are decremented and become H'00, and the CFEMP flag in the CFSTSk register becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

(6) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the GCFGL register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(7) The message is stored in the transmit/receive FIFO buffer set in receive mode, when the message has passed through the DLC filter process if the CFE bit is set to 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] bits are set to B'001 or more.

The CFMC[5:0] value is incremented to H'01. When the CFIM bit is set to 1 (an interrupt occurs each time a message has been received), the CFRXIF flag is set to 1 (a transmit/receive FIFO receive interrupt request is present).

The message is stored in the receive FIFO buffer, if the RFE bit in the RFCCm register is set to 1 (receive FIFO buffers are used) and RFDC[2:0] bits in the RFCCm register are set to B'001 or more.

The RFMC[5:0] value in the RFSTSm register is incremented to H'01. When the RFIM bit in the RFCCm register is set to 1 (an interrupt occurs each time a message has been received), the RFIF flag in the RFSTSm register is set to 1 (a receive FIFO interrupt request is present).

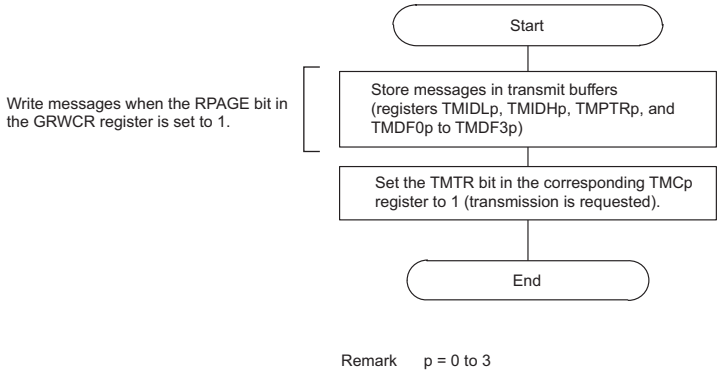
## 18.12 Transmission Procedure

### 18.12.1 Procedure for Transmission from Transmit Buffers

Figure 18-26 shows the procedure for transmission from transmit buffers.

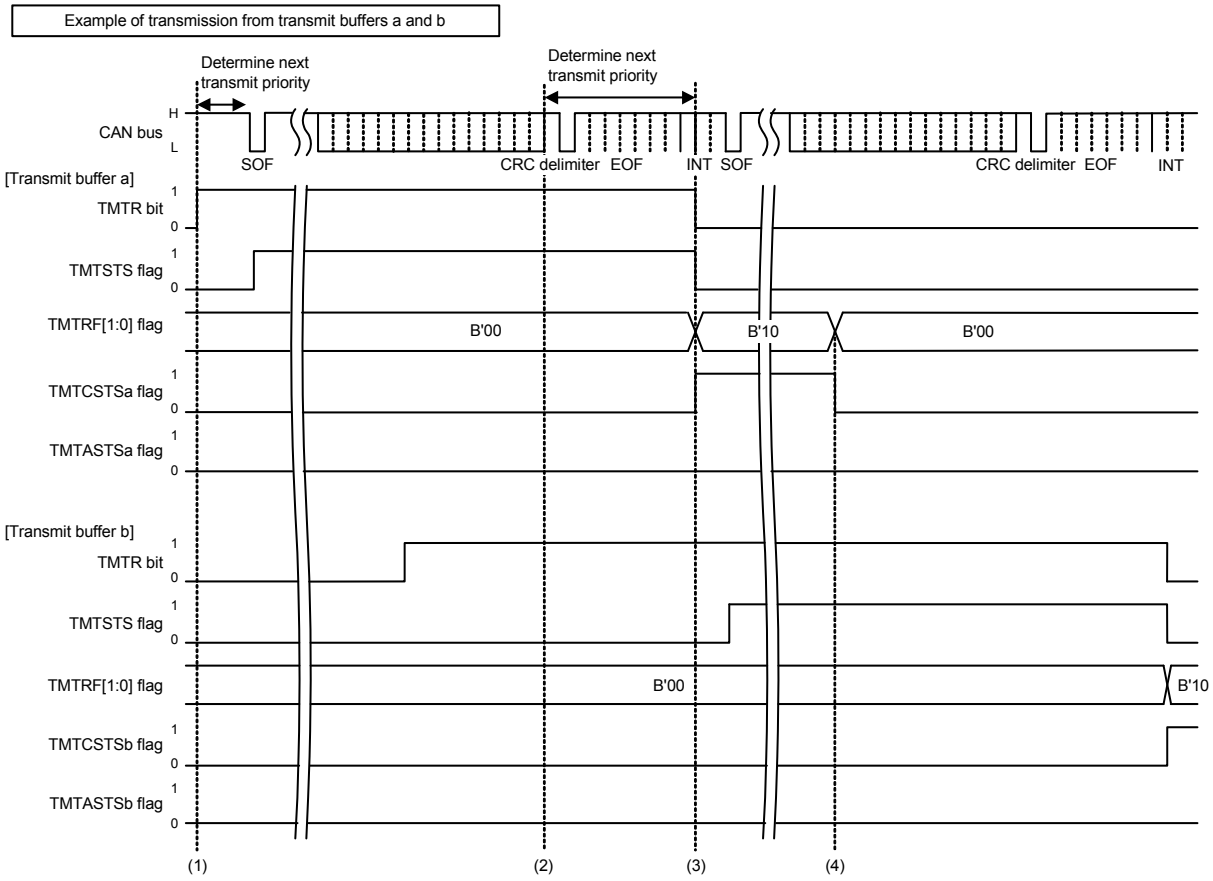
Figure 18-27 shows a timing chart where messages are transmitted from two transmit buffers and transmission has been successfully completed. Figure 18-28 shows a timing chart where messages are transmitted from two transmit buffers and transmit abort has been completed.

**Figure 18-26. Procedure for Transmission from Transmit Buffers**




Remark p = 0 to 3

**Figure 18-27. Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully)**

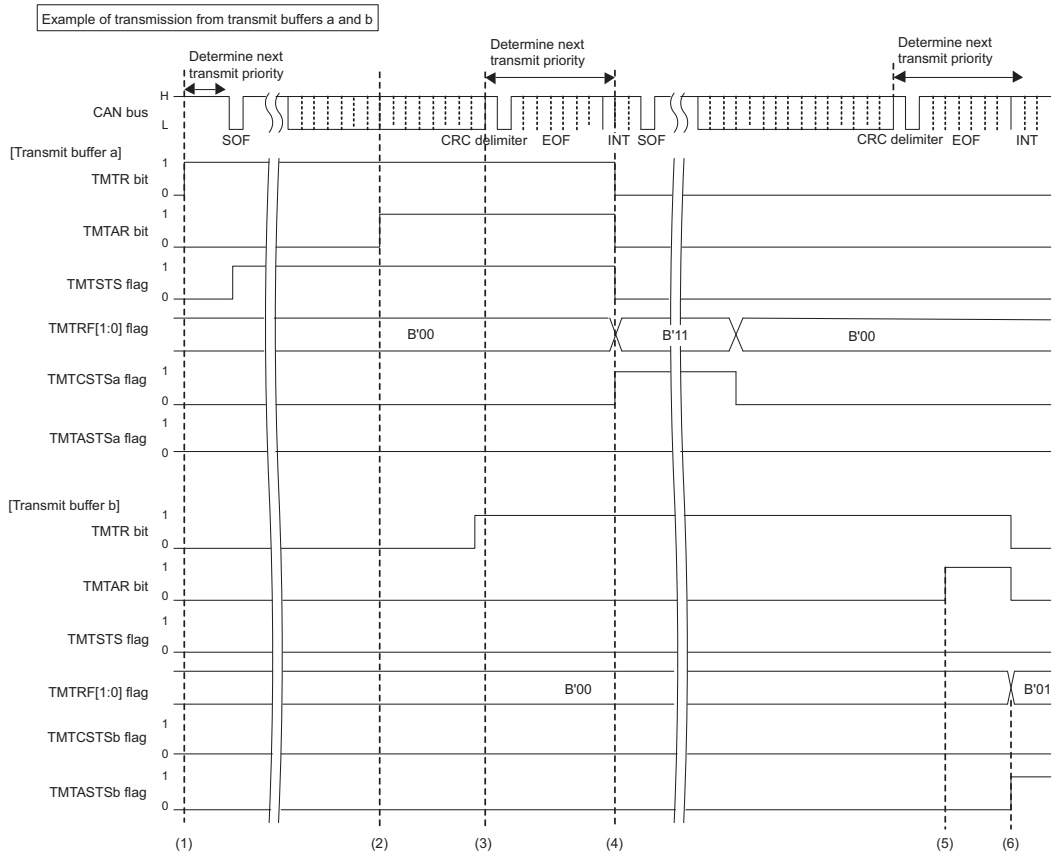



Remark a = 0 to 3, b = 0 to 3
TMTR: Bit in the TMCa or TMCb register
TMTSTS, TMTRF[1:0]: Flags in the TMSTSa or TMSTSb register
TMTCSTSa, TMTCSTSb: Flag in the TMTCSTS register
TMTASTSa, TMTASTSb: Flag in the TMTASTS register

(1) When the TMTR bit in the TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.
(2) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission.
(3) When transmit completes successfully, the TMTRF[1:0] flag in the TMSTSa register is set to B'10 (transmission has been completed (without transmit abort request)), the TMTSTS flag and the TMTR bit in the TMCa register are cleared to 0, and the TMTCSTSa bit in the TMTCSTS register is set to 1. When the TMIEa value in the TMIEC register is 1 (transmit buffer interrupt is enabled), a CANi transmit interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to B'00 (transmission is in progress or no transmit request is present).
(4) Before starting the next transmission, set the TMTRF[1:0] flag to B'00. Write the next message to the transmit buffer, and then set the TMTR bit to 1 (transmission is requested). The TMTR bit can be set to 1 only when the TMTRF[1:0] flag value is B'00.

If an arbitration lost has occurred after transmission is started, the TMTSTS flag is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration lost, the priority determination processing is reexecuted during transmission of an error frame.

**Figure 18-28. Transmit Buffer Transmission Timing Chart (Transmit Abort Completed)**




Remark a = 0 to 3, b = 0 to 3
TMTR: Bit in the TMCa or TMCb register
TMTSTS, TMTRF[1:0]: Flags in the TMSTSa or TMSTSb register
TMTCSTSa, TMTCSTSb: Flag in the TMTCSTS register
TMTASTSa, TMTASTSb: Flag in the TMTASTS register

(1) When the TMTR bit in the TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.
(2) When it is determined that the transmit buffer is used for the next transmission or transmission is in progress, message transmission is not aborted unless an error or arbitration lost occurs even if the TMTAR bit is set to 1 (transmit abort is requested).
(3) The priority determination starts with the CRC delimiter for the next transmission. In this timing chart, buffer b is not selected as the next transmit buffer.
(4) When transmit completes successfully, the TMTRF[1:0] flag in the TMSTSa register is set to B'11 (transmission has been completed (with transmit abort request)), the TMTSTS flag and the TMTR bit in the TMCa register are cleared to 0, and the TMTCSTSa bit in the TMTCSTS register is set to 1. When the TMIEa value in the TMIEC register is 1 (transmit buffer interrupt is enabled), a CANi transmit interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to B'00 (transmission is in progress or no transmit request is present).
(5) While another CAN node is transmitting data on the CAN bus (TMTSTS flag = 0), if the TMTAR bit is set to 1 while the corresponding channel is determining transmit priority, the TMTR bit cannot be cleared to 0.
(6) After the internal processing time has passed, the transmission is terminated and the TMTRF[1:0] flag is set to B'01 and the TMTASTSb bit in the TMTASTS register is set to 1. When the transmit buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not being made, an abort request is immediately accepted and the TMTRF[1:0] flag is set to B'01. At this time, the TMTR and TMTAR bits are cleared to 0. When transmit abort is completed with the TAIE bit in the CiCTRH register set to 1 (transmit abort interrupt is enabled), an interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to B'00.

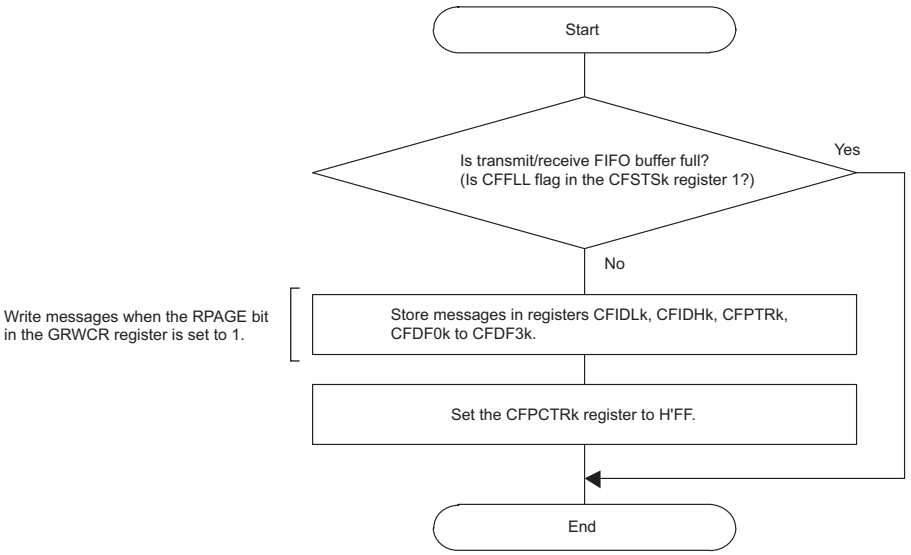
If an arbitration lost has occurred after the CAN channel started transmission, the TMTSTS bit is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration lost, the priority determination processing is reexecuted during transmission of an error frame.

### 18.12.2 Procedure for Transmission from Transmit/Receive FIFO Buffers

Figure 18-29 shows the procedure for transmission from transmit/receive FIFO buffers.

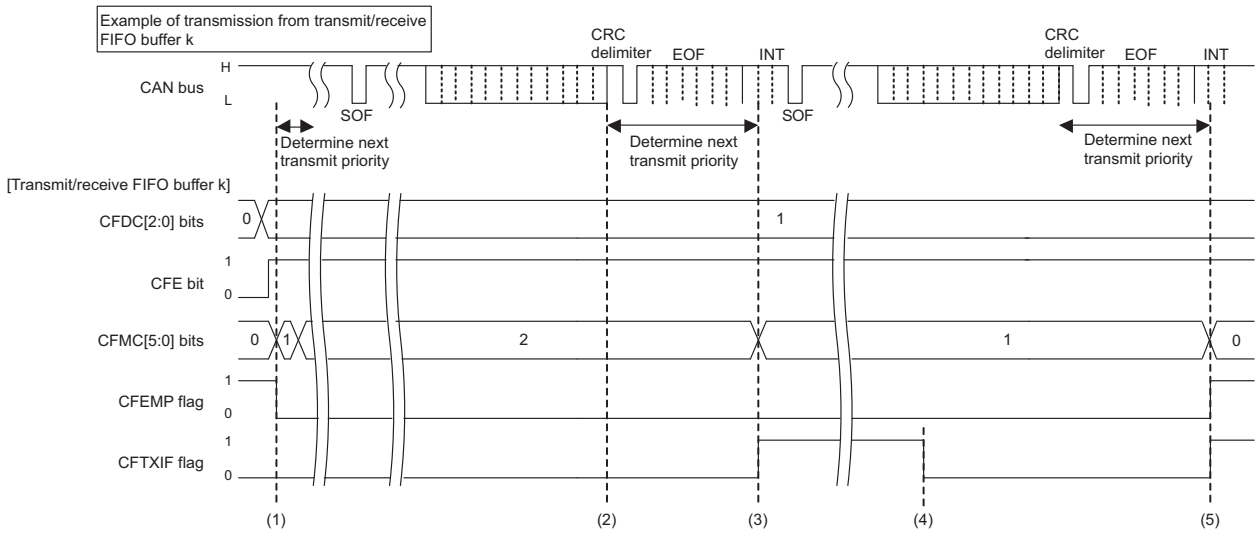
Figure 18-30 shows a timing chart where messages are transmitted from the transmit/receive FIFO buffers and transmission has been successfully completed. Figure 18-31 shows a timing chart where messages are transmitted from the transmit/receive FIFO buffers and transmit abort has been completed.

**Figure 18-29. Procedure for Transmission from Transmit/Receive FIFO Buffers**




Remark k = 0

**Figure 18-30. Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully)**




Remark k = 0
CFDC[2:0], CFE: Bits in the CFCCLk register
CFMC[5:0], CFEMP, CFTXIF: Flags in the CFSTSk register

(1) While the CAN bus is idle, when the CFE value in the CFCCLk register is 1 (transmit/receive FIFO buffer k is used) and the CFDC[2:0] value in the CFCCLk register is B'001 (4 messages) or more and the CFMC[5:0] value in the CFSTSk register is H'01 or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts.

(2) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission.

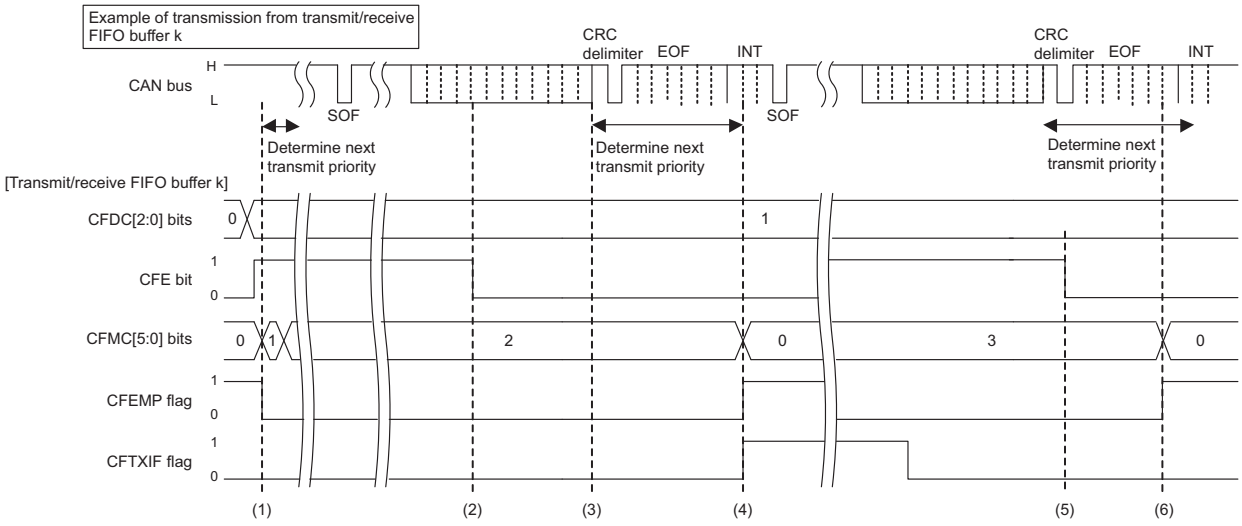
(3) When transmit completes successfully, the CFMC[5:0] value in the CFSTSk register is decremented. Setting the CFIM bit in the CFCCLk register to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the CFSTSk register to 1 (a transmit/receive FIFO transmit interrupt request is present).

(4) The CFTXIF flag can be cleared by the program.

(5) Message transmission from transmit/receive FIFO buffer k has been completed and the CFMC[5:0] value in the CFSTSk register is decremented. The CFMC[5:0] bits are cleared to H'00 and therefore the CFEMP flag in the CFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

Transmission is continued until the CFEMP flag is set to 1. It is possible to continuously store transmit messages in FIFO buffers until the CFFLL flag in the CFSTSk register is set to 1 (the transmit/receive FIFO buffer is full).

**Figure 18-31. Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)**




Remark k = 0
CFDC[2:0], CFE: Bits in the CFCCLk register
CFMC[5:0], CFEMP, CFTXIF: Flags in the CFSTSk register

(1) While the CAN bus is idle, when the CFE value in the CFCCLk register is 1 (transmit/receive FIFO buffer k is used) and the CFDC[2:0] value in the CFCCLk register is B'001 (4 messages) or more and the CFMC[5:0] value in the CFSTSk register is H'01 or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts.

(2) When transmission is in progress or it is determined that the transmit/receive FIFO buffer is used for the next transmission, message transmission is not aborted unless an error or arbitration lost occurs even if the CFE bit is set to 0 (no transmit/receive FIFO buffer k is used).

(3) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission. In this figure, transmit/receive FIFO buffer k is not selected as a buffer for the next transmission.

(4) When transmit completes successfully, the CFMC[5:0] value is cleared to H'00. Setting the CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the CFSTSk register to 1 (a transmit/receive FIFO transmit interrupt request is present). The CFTXIF flag can be cleared by the program.

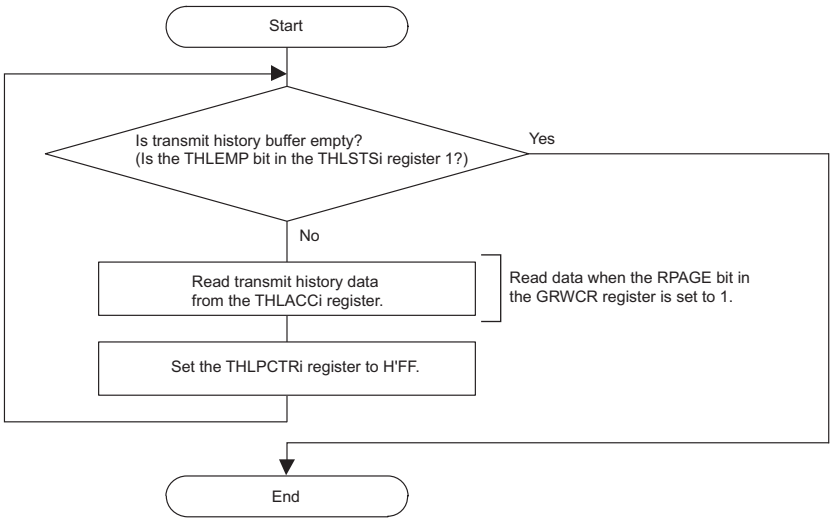
(5) If another CAN node on the CAN bus is transmitting data (not from transmit/receive FIFO buffer k), transmit/receive FIFO buffer k cannot be disabled immediately even if the CFE bit in the CFCCLk register is cleared to 0 (no transmit/receive FIFO buffer k is used) during transmit priority determination. (The CFEMP flag in the CFSTSk register is not set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) immediately.)

(6) After the internal processing time has passed, transmit/receive FIFO buffers are disabled and the CFMC[5:0] bits in the CFSTSk register are cleared to H'00 and the CFEMP flag is set to 1. When the transmit/receive FIFO buffer k is not transmitting data and is not selected as the next transmit buffer and priority determination is not in progress, the transmit/receive FIFO buffer k is immediately disabled. (The CFMC[5:0] bits are cleared to H'00 and the CFEMP flag is set to 1.)

### 18.12.3 Transmit History Buffer Reading Procedure

Transmit history data can be read from the THLACCi register. The next data can be accessed by writing H'FF to the corresponding THLPCTRi register after reading a set of data. Figure 18-32 shows the transmit history buffer reading procedure.

**Figure 18-32. Transmit History Buffer Reading Procedure**




Remark i = 0

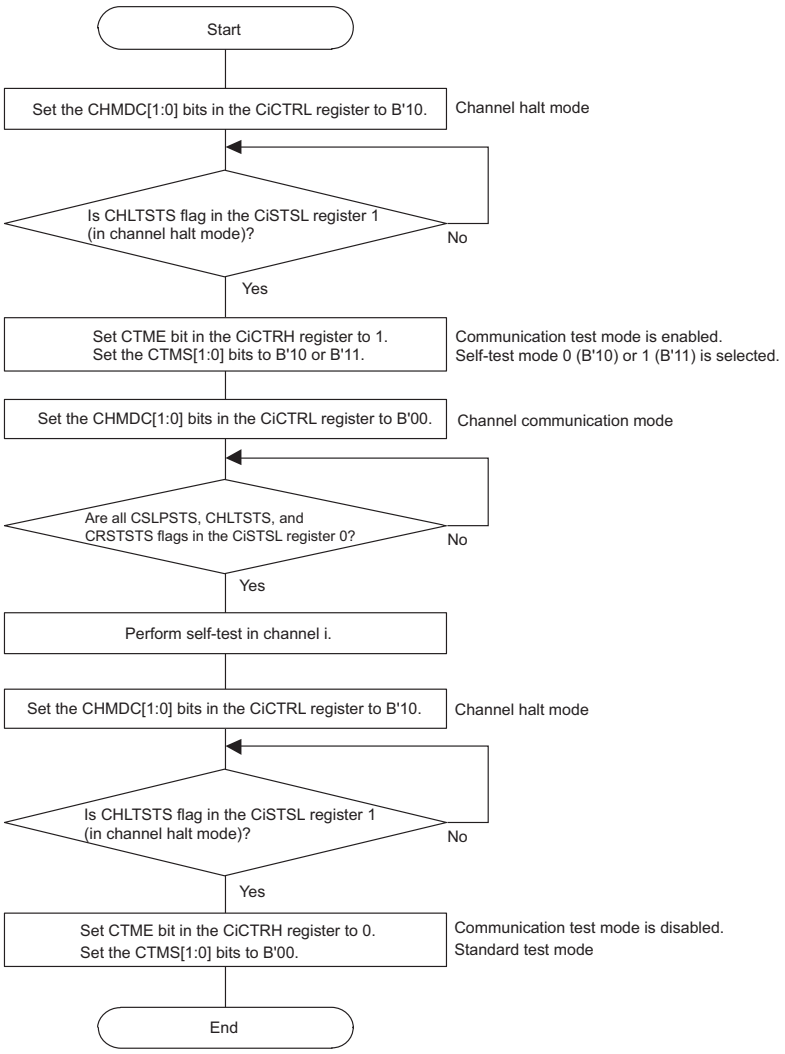
## 18.13 Test Settings

### 18.13.1 Self-Test Mode Setting Procedure

Self-test mode allows communication test on a channel basis by receiving messages transmitted from the own node.

Figure 18-33 shows the self-test mode setting procedure.

**Figure 18-33. Self-Test Mode Setting Procedure**




Remark i = 0

### 18.13.2 Protection Unlock Procedure

Since the global test functions shown in Table 18-15 are protected, write unlock data 1 and unlock data 2 in succession to the LOCK[15:0] bits in the GLOCKK register, and then set each test function bit to 1.

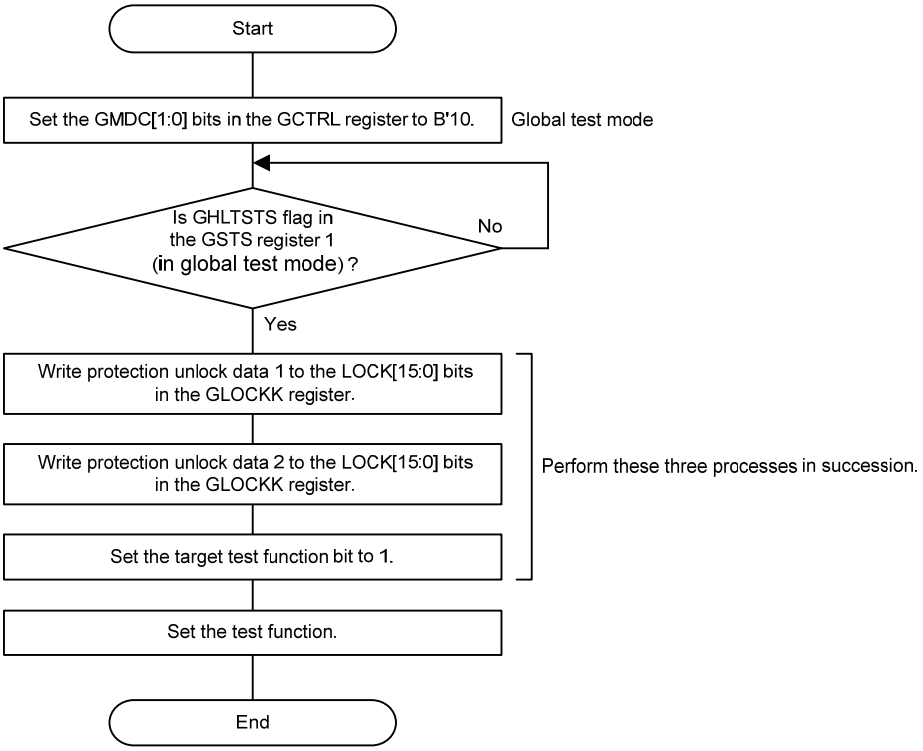
**Table 18-15. Protection Unlock Data for Test Functions**

| Test Function | Protection Unlock Data 1 | Protection Unlock Data 2 | Target Bit |
|---|---|---|---|
| RAM test | H'7575 | H'8A8A | RTME bit in the GTSTCTRL register |

If an incorrect value has been written to the LOCK[15:0] bits, retry the procedure above from writing of unlock data 1.

Figure 18-34 shows the protection unlock procedure.

**Figure 18-34. Protection Unlock Procedure**

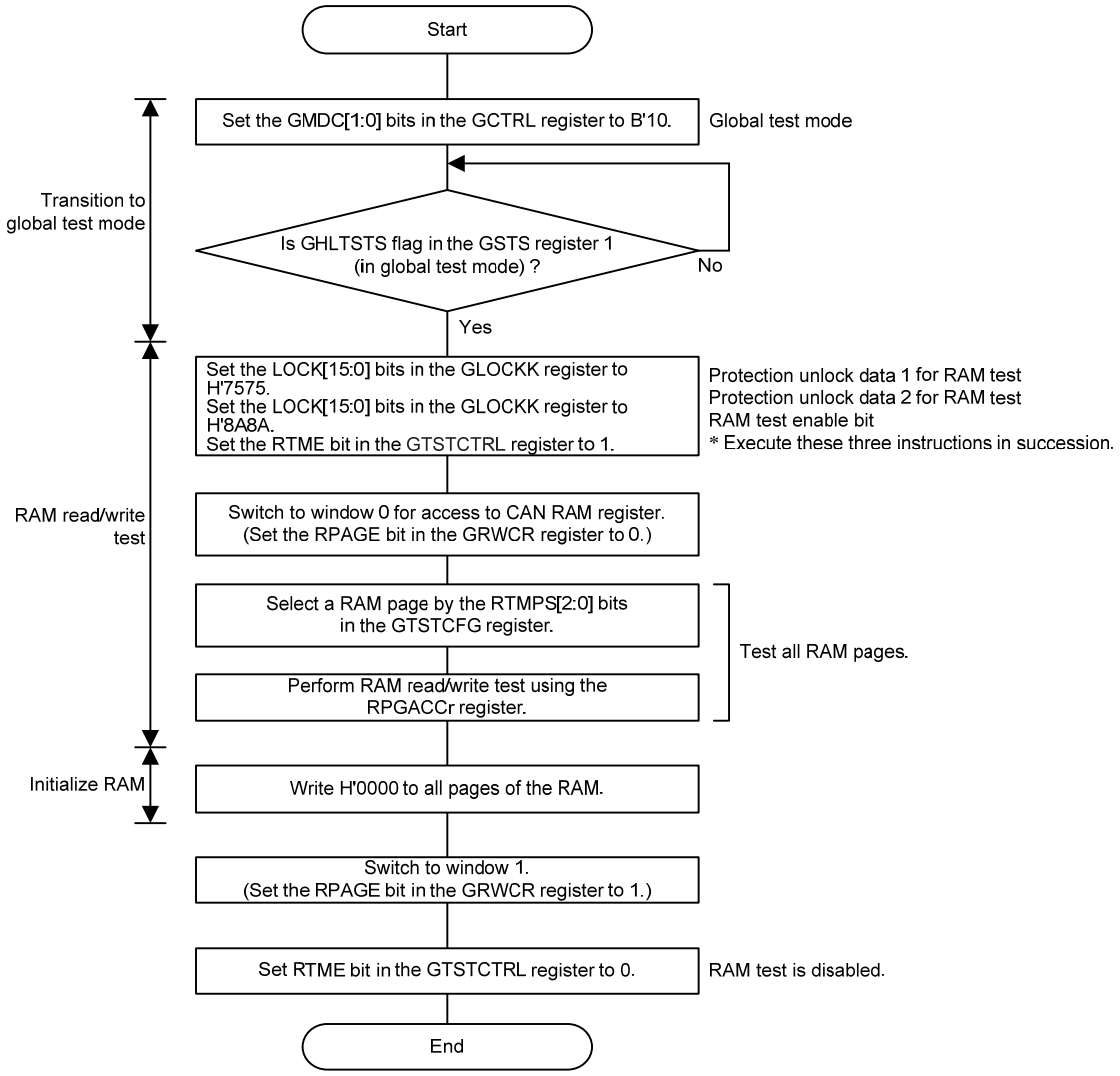


### 18.13.3 RAM Test Setting Procedure

RAM tests include CAN RAM read/write test. The read/write test verifies that data written to the RAM is read correctly. Before closing the RAM test, write H'0000 to all pages of the CAN RAM.

Figure 18-35 shows the RAM test setting procedure.

**Figure 18-35. RAM Test Setting Procedure**




Remark r = 0 to 127

## 18.14 Notes on the CAN Module

- When changing a global mode, check the GSLPSTS, GHLTSTS, and GRSTSTS flags in the GSTS register for transitions. When changing a channel mode, check the CSLPSTS, CHLTSTS, and CRSTSTS flags in the CiSTSL register for transitions.
- The acceptance filter processing checks receive rules sequentially in ascending order from the minimum rule number. If the same ID, IDE bit, or RTR bit value is set for multiple receive rules, the minimum number of receive rule is used for the acceptance filter processing. If the message does not pass through the subsequent DLC filter processing, the data processing is terminated without returning to the acceptance filter processing and the message is not stored in the buffer.
- When linking transmit buffers to transmit/receive FIFO buffers, set the control register (TCMp) of the corresponding transmit buffer to H'00. The status register (TMSTSp) of the corresponding transmit buffer should not be used. Flags in other status registers (registers TMTRSTS, TMTCSTS, and TMTASTS), which correspond to transmit buffers linked to transmit/receive FIFO buffers remain unchanged. Set the enable bit in the corresponding interrupt enable register (the TMIEC register) to 0 (transmit buffer interrupt is disabled).
- When the CANi bit time clock is selected as a timestamp counter clock source, the timestamp counter stops when the corresponding channel has transitioned to channel reset mode or channel halt mode.
- In case of an attempt to store a new receive message when the receive FIFO buffer and the transmit/receive FIFO buffer are full, the new message is discarded. If you wish to store a new transmit message in the transmit/receive FIFO buffer, check that the transmit/receive FIFO buffer is not full.
- Since an interrupt request flag in the CAN module is not automatically cleared to 0 when an interrupt is accepted, the flags must be cleared to 0 by software. After the corresponding interrupt request flag has been set to 1, an interrupt is not generated even if an interrupt source condition is satisfied.
- In order to generate the CAN related interrupt that several interrupt sources are gathered, the following condition should be met:
  All interrupt request flags corresponding to these interrupt sources in the CAN module are set to 0 (note that this only applies to those interrupt request flags for which the corresponding interrupt enable bits shown in Table 18-12 are set to 1).
- The values of unused CAN receive buffer registers (RMIDLn, RMIDHn, RMTSn, RMPTRn, and RMDF0n to RMDF3n), CAN receive FIFO access registers (RFIDLm, RFIDHm, RFTSm, RFPTRm, and RFDF0m to RFDF3m), and CANi transmit/receive FIFO access registers (CFIDLk, CFIDHk, CFTSk, CFPTRk, and CFDF0k to CFDF3k) become undefined once the CAN module exits from global reset mode and enters global operation mode or global test mode.

# CHAPTER 19 DTC

The DTC (data transfer controller) is a function that transfers data between memories without using the CPU. The DTC is activated by a peripheral function interrupt to perform data transfers. The DTC and CPU use the same bus, and the DTC takes priority over the CPU in using the bus.

To control DTC data transfers, control data comprised of a transfer source address, a transfer destination address, and operating modes are allocated in the DTC control data area. Each time the DTC is activated, the DTC reads control data to perform data transfers. The DTC control data area is allocated in the RAM space set by the DTCBAR register.

The high-speed transfer is realized by allocating the dedicated control data in the SFR area instead of the RAM area.

## 19.1 Overview

Table 19-1 lists the DTC specifications.

**Table 19-1. DTC Specifications**

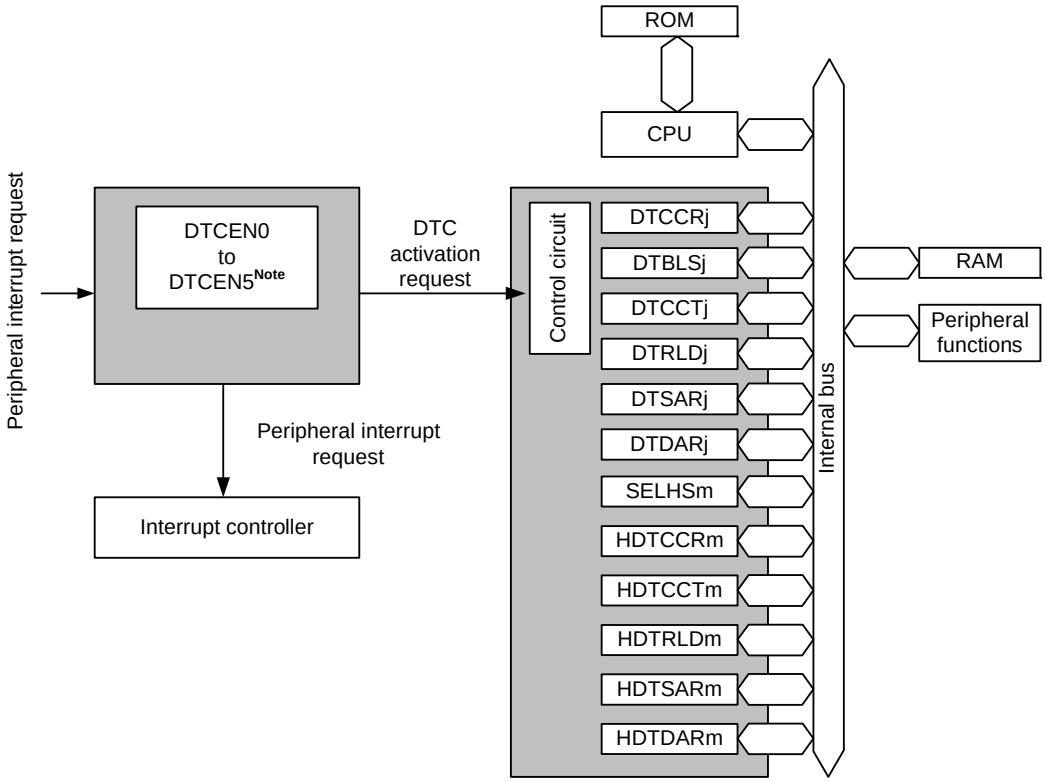
| Item | | Specification |
|---|---|---|
| Activation sources | | 44 sources max. |
| Allocatable control data | | 24 sets/2 sets (high-speed transfer) |
| Address space which can be transferred | Address space | 64 Kbytes (F0000H to FFFFFH), excluding general-purpose registers |
| | Sources Note 2 | 1st SFR area, RAM area (excluding general-purpose registers), mirror area Note 1, data flash memory area Note 1, 2nd SFR area |
| | Destinations | 1st SFR area, RAM area (excluding general-purpose registers), 2nd SFR area |
| Maximum number of transfers | Normal mode | 256 times |
| | Repeat mode | 255 times |
| Maximum size of block to be transferred | Normal mode (8-bit transfer) | 256 bytes/1 byte (high-speed transfer) |
| | Normal mode (16-bit transfer) | 512 bytes/2 bytes (high-speed transfer) |
| | Repeat mode | 255 bytes/1 byte at 8-bit transfer (high-speed transfer)/2 bytes at 16-bit transfer (high-speed transfer) |
| Unit of transfers | | 8 bits/16 bits |
| Transfer mode | Normal mode | Transfers end on completion of the transfer causing the DTCCTj and HDTCCTm registers value to change from 1 to 0. |
| | Repeat mode | On completion of the transfer causing the values of the DTCCTj and HDTCCTm registers to change from 1 to 0, the repeat area address is initialized and the DTRLDj register value is reloaded to the DTCCTj and HDTCCTm registers to continue transfers. |
| Address control | Normal mode | Fixed or incremented |
| | Repeat mode | Addresses of the area not selected as the repeat area are fixed or incremented. |
| Priority of activation sources | | Refer to **Table 19-5 DTC Activation Sources and DTC Vector Addresses**. |
| Interrupt request | Normal mode | When the data transfer causing the DTCCTj and HDTCCTm registers value to change from 1 to 0 is performed, the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the data transfer. |
| | Repeat mode | When the data transfer causing the values of the DTCCTj and HDTCCTm registers to change from 1 to 0 is performed while the RPTINT and HRPTINTm bits in the DTCCRj and HDTCCRm registers, respectively are 1 (interrupt generation enabled), the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the transfer. |
| Transfer start | | When bits DTCENi0 to DTCENi7 in the DTCENi registers are 1 (activation enabled), data transfer is started each time the corresponding DTC activation sources are generated. |
| Transfer stop | Normal mode | • When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled).<br>• When the data transfer causing the DTCCTj and HDTCCTm registers value to change from 1 to 0 is completed. |
| | Repeat mode | • When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled).<br>• When the data transfer causing the values of the DTCCTj and HDTCCTm registers to change from 1 to 0 is completed while the RPTINT bit is 1 (interrupt generation enabled). |
| Operation in standby mode | HALT state | DTC operates |
| | SNOOZE state | DTC operates |
| | STOP state | DTC stops |

**Notes 1.** In the HALT, STOP, and SNOOZE modes, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped.

**2.** High-speed transfer is enabled only for the 1st SFR area and the 2nd SFR area.

**Remark** Products of groups A, B, C, and D: i = 0 to 4, j = 0 to 23, m = 0, 1
Products of group E: i = 0 to 5, j = 0 to 23, m = 0, 1

**Figure 19-1. DTC Block Diagram**




**Note** Products of groups A, B, C, and D do not have the DTCEN5 register.

**Remark** j = 0 to 23, m = 0, 1

DTCCRj: DTC control register j
DTBLSj: DTC block size register j
DTCCTj: DTC transfer count register j
DTRLDj: DTC transfer count reload register j
DTSARj: DTC source address register j
DTDARj: DTC destination address register j
DTCEN0 to DTCEN5Note: DTC activation enable registers 0 to 5Note
SELHSm: High-speed DTC channel select register m
HDTCCRm: High-speed DTC control register m
HDTCCTm: High-speed DTC transfer count register m
HDTRLDm: High-speed DTC transfer count reload register m
HDTSARm: High-speed DTC source address register m
HDTDARm: High-speed DTC destination address register m

## 19.2 Registers

Tables 19-2 and 19-4 list the DTC register configuration.

**Table 19-2. DTC Register Configuration (1)**

| Register Name | Symbol | After Reset | Address | Access Size |
|---|---|---|---|---|
| Peripheral Enable Register 1 | PER1 | 00H | F02C0H | 1, 8 |
| DTC Activation Enable Register 0 | DTCEN0 | 00H | F02E8H | 1, 8 |
| DTC Activation Enable Register 1 | DTCEN1 | 00H | F02E9H | 1, 8 |
| DTC Activation Enable Register 2 | DTCEN2 | 00H | F02EAH | 1, 8 |
| DTC Activation Enable Register 3 | DTCEN3 | 00H | F02EBH | 1, 8 |
| DTC Activation Enable Register 4 | DTCEN4 | 00H | F02ECH | 1, 8 |
| DTC Activation Enable Register 5 Note | DTCEN5 | 00H | F02EDH | 1, 8 |
| DTC Base Address Register | DTCBAR | FDH | F02E0H | 8 |

**Note** Only in products of group E.

Table 19-3 lists DTC control data.

DTC control data is allocated in the DTC control data area in RAM.

The DTCBAR register is used to set the 256-byte area, including the DTC control data area and the DTC vector table area where the start address for control data is stored.

**Table 19-3. DTC Control Data**

| Register Name | Symbol |
|---|---|
| DTC Control Register j | DTCCRj |
| DTC Block Size Register j | DTBLSj |
| DTC Transfer Count Register j | DTCCTj |
| DTC Transfer Count Reload Register j | DTRLDj |
| DTC Source Address Register j | DTSARj |
| DTC Destination Address Register j | DTDARj |

**Remark** j = 0 to 23

**Table 19-4. DTC Register Configuration (2)**

| Register Name | Symbol | After Reset | Address | Access Size |
|---|---|---|---|---|
| High-speed DTC Channel Select Register 0 | SELHS0 | 3FH | F02E1H | 1, 8 |
| High-speed DTC Channel Select Register 1 | SELHS1 | 3FH | F02E2H | 1, 8 |
| High-speed DTC Control Register 0 | HDTCCR0 | 00H | F02D0H | 1, 8 |
| High-speed DTC Transfer Count Register 0 | HDTCCT0 | 00H | F02D2H | 1, 8 |
| High-speed DTC Transfer Count Reload Register 0 | HDTRLD0 | 00H | F02D3H | 1, 8 |
| High-speed DTC Source Address Register 0 | HDTSAR0 | 00H | F02D4H, F02D5H | 16 |
| High-speed DTC Destination Address Register 0 | HDTDAR0 | 00H | F02D6H, F02D7H | 16 |
| High-speed DTC Control Register 1 | HDTCCR1 | 00H | F02D8H | 1, 8 |
| High-speed DTC Transfer Count Register 1 | HDTCCT1 | 00H | F02DAH | 1, 8 |
| High-speed DTC Transfer Count Reload Register 1 | HDTRLD1 | 00H | F02DBH | 1, 8 |
| High-speed DTC Source Address Register 1 | HDTSAR1 | 00H | F02DCH, F02DDH | 16 |
| High-speed DTC Destination Address Register 1 | HDTDAR1 | 00H | F02DEH, F02DFH | 16 |

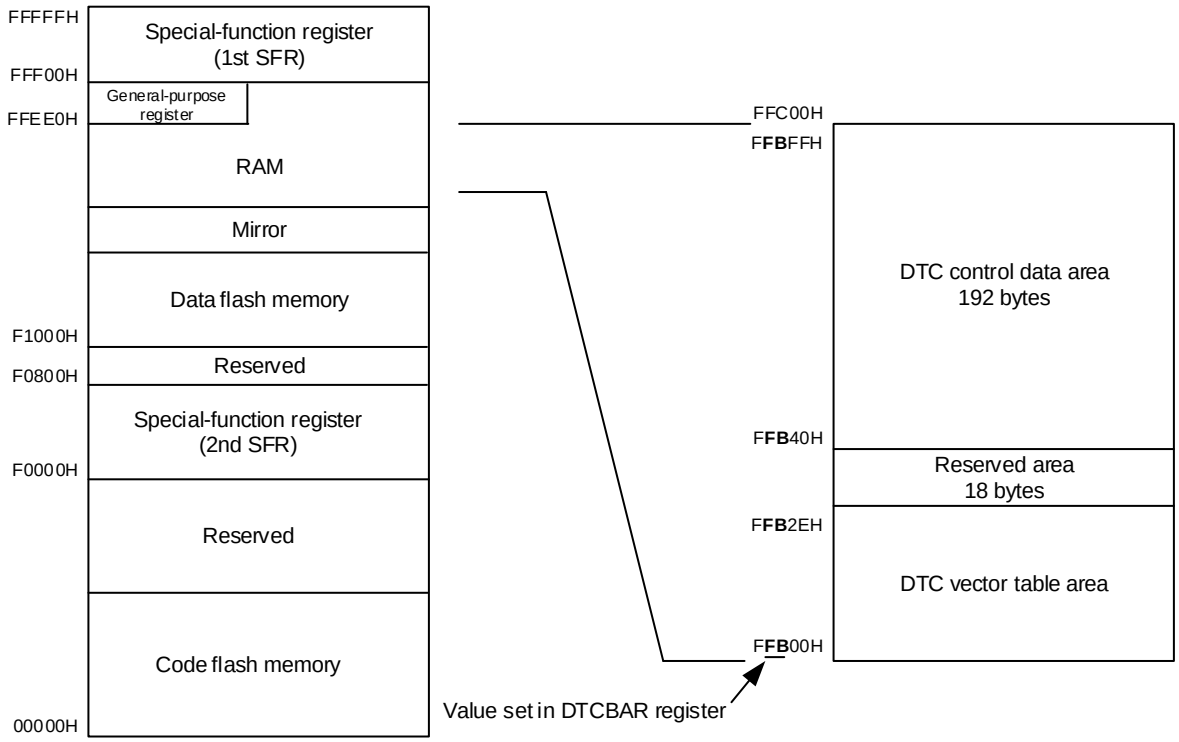
### 19.2.1 Allocation of DTC Control Data Area and DTC Vector Table Area

The DTCBAR register is used to set the 256-byte area where DTC control data and the vector table within the RAM area.

Figure 19-2 shows a memory map example when DTCBAR register is set to FBH.

In the 192-byte DTC control data area, the space not used by the DTC can be used as RAM.

**Figure 19-2. Memory Map Example when DTCBAR Register is Set to FBH**




The areas where the DTC control data and vector table can be allocated differ depending on the usage conditions.

**Cautions 1. It is prohibited to use the general-purpose register (FFF00H to FFEE0H) space as the DTC control data area or DTC vector table area.**

**2. The 18-byte area between the DTC vector table area and the DTC control data area is reserved for use when the number of DTC activation sources is expanded.**

**3. The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the self-programming function and data flash function.**

**R5F10AmE (m = 6, A, B, G, L) : FEF00H to FF2FFH**
**R5F10PmF (m = G, L, M) : FDF00H to FE2FFH**
**R5F10AmG (m = G, L, M), R5F10BnG (n = A, B, G, L, M) : FDF00H to FE2FFH**
**R5F10PmJ (m = G, L, M, P) : FAF00H to FB2FFH**

**4. The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the tracing function of on-chip debugging.**

**R5F10AmE (m = 6, A, B, G, L), R5F10AnD (n = 6, A, B, G, L) : FF300H to FF37FH**
**R5F10AmG (m = G, L, M), R5F10BnG (n = A, B, G, L, M) : FE300H to FE4FFH**
**R5F10PmF (m = G, L, M) : FE300H to FE4FFH**
**R5F10PmJ (m = G, L, M, P) : FB300H to FB4FFH**

**5. The internal RAM in the following products cannot be used as stack memory when the hot plug-in function is used or when the DTC is in use for the RRM or DMM function.**

**R5F10AmD, R5F10AmE (m = 6, A, B, G, L) : FF400H to FF42FH**
**R5F10AmG (m = G, L, M), R5F10BnG (n = A, B, G, L, M) : FE500H to FE52FH**
**R5F10PmF (m = G, L, M) : FE500H to FE52FH**
**R5F10PmJ (m = G, L, M, P) : FB500H to FB52FH**

### 19.2.2 DTC Control Data Allocation

Control data is allocated beginning with each start address in the order: Registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj (j = 0 to 23).

The higher 8 bits for start addresses 0 to 23 are set by the DTCBAR register, and the lower 8 bits are separately set according to the vector table assigned to each activation source.

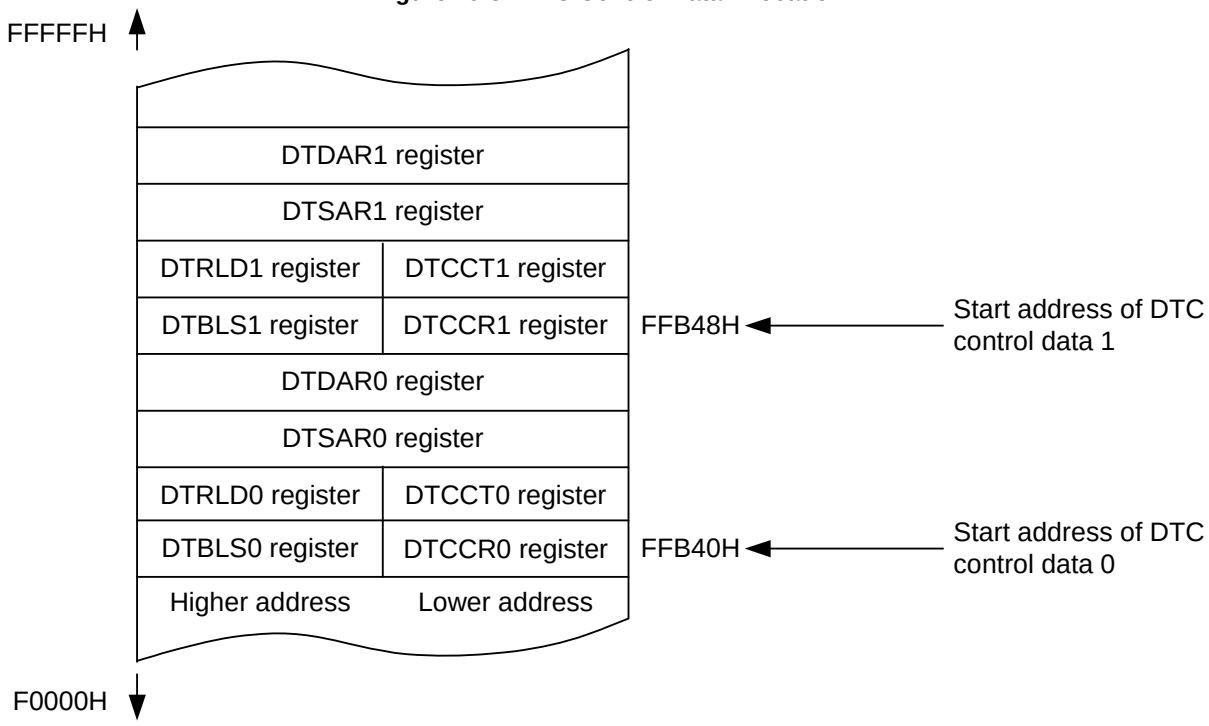
Figure 19-3 shows an example of DTC control data allocation when the DTCBAR register is set to FBH.

**Cautions**

- Change the data in registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 5[Note]) in the DTCENi register is set to 0 (DTC activation disabled).
- Do not access DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj using a DTC transfer.

**Note** Products of groups A, B, C, and D: i = 0 to 4
Products of group E: i = 0 to 5

**Figure 19-3. DTC Control Data Allocation**



### 19.2.3 DTC Vector Table

When the DTC is activated, one control data is selected according to the data read from the vector table which has been assigned to each activation source, and the selected control data is read from the DTC control data area.

Table 19-5 lists the DTC activation sources and DTC vector addresses. A one byte of the DTC vector table is assigned to each activation source, and the lower 8 bits for the start address of the DTC control data are stored in each area to select one of the 24 sets. The higher 8 bits for the DTC vector address are set by the DTCBAR register, and 00H to 2DH are allocated to the lower 8 bits corresponding to the DTC activation source.

**Caution**

- Change the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 5[Note]) in the DTCENi register is set to 0 (DTC activation disabled).

**Note** Products of groups A, B, C, and D: i = 0 to 4
Products of group E: i = 0 to 5

**Table 19-5. DTC Activation Sources and DTC Vector Addresses (1/2)**

| Interrupt Request Source | Source No. | DTC Vector Address | Priority |
|---|---|---|---|
| Reserved | 0 | Address set in DTCBAR register +00H | Highest |
| INTP0 | 1 | Address set in DTCBAR register +01H | |
| INTP1 | 2 | Address set in DTCBAR register +02H | |
| INTP2 | 3 | Address set in DTCBAR register +03H | |
| INTP3 | 4 | Address set in DTCBAR register +04H | |
| INTP4 | 5 | Address set in DTCBAR register +05H | |
| INTP5 Note 1 | 6 | Address set in DTCBAR register +06H | |
| INTP6 Notes 1, 2 | 7 | Address set in DTCBAR register +07H | |
| Key input | 8 | Address set in DTCBAR register +08H | |
| A/D conversion end | 9 | Address set in DTCBAR register +09H | |
| UART0 reception transfer end/CSI01 transfer end or buffer empty/IIC01 transfer end | 10 | Address set in DTCBAR register +0AH | |
| UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer end | 11 | Address set in DTCBAR register +0BH | |
| UART1 reception transfer end/CSI11 transfer end or buffer empty/IIC11 transfer end Notes 3, 4 | 12 | Address set in DTCBAR register +0CH | |
| UART1 transmission transfer end/CSI10 transfer end or buffer empty/IIC10 transfer end Note 3 | 13 | Address set in DTCBAR register +0DH | |
| LIN0 reception end | 14 | Address set in DTCBAR register +0EH | |
| LIN0 transmission start/end | 15 | Address set in DTCBAR register +0FH | |
| CAN reception end Note 5 | 16 | Address set in DTCBAR register +10H | |
| Reserved | 17 | Address set in DTCBAR register +11H | |
| End of channel 0 of timer array unit 0 count or capture | 18 | Address set in DTCBAR register +12H | |
| End of channel 1 of timer array unit 0 count or capture | 19 | Address set in DTCBAR register +13H | |
| End of channel 2 of timer array unit 0 count or capture | 20 | Address set in DTCBAR register +14H | |
| End of channel 3 of timer array unit 0 count or capture | 21 | Address set in DTCBAR register +15H | |
| End of channel 4 of timer array unit 0 count or capture | 22 | Address set in DTCBAR register +16H | |
| End of channel 5 of timer array unit 0 count or capture | 23 | Address set in DTCBAR register +17H | |
| End of channel 6 of timer array unit 0 count or capture | 24 | Address set in DTCBAR register +18H | |
| End of channel 7 of timer array unit 0 count or capture | 25 | Address set in DTCBAR register +19H | |

(**Notes** and **Remark** are listed on the next page.)

**Remark** Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory

Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory

Group C: RL78/F13 (LIN and CAN incorporated) products with 30, 32, 48, 64, or 80 pins and 32 Kbytes to 128 Kbytes of code flash memory

Group D: RL78/F14 products with 30, 32, 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory

Group E: RL78/F14 products with 48, 64, 80, or 100 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

**Notes 1.** Not provided in the 20-pin products.

**2.** Not provided in the products with 30 or 32 pins.

**3.** Provided only in the products of Groups B to E.

**4.** RL78/F13 (CAN and LIN incorporated) products with 30 or 32 pins and RL78/F14 products with 30 or 32 pins do not have CSI11 transfer end or buffer empty interrupt/IIC11 transfer end interrupt.

**5.** Provided only in the products of Groups C to E.

**Table 19-5. DTC Activation Sources and DTC Vector Addresses (2/2)**

| Interrupt Request Source | Source No. | DTC Vector Address | Priority |
|---|---|---|---|
| Timer RD compare match A0 | 26 | Address set in DTCBAR register +1AH | |
| Timer RD compare match B0 | 27 | Address set in DTCBAR register +1BH | |
| Timer RD compare match C0 | 28 | Address set in DTCBAR register +1CH | |
| Timer RD compare match D0 | 29 | Address set in DTCBAR register +1DH | |
| Timer RD compare match A1 | 30 | Address set in DTCBAR register +1EH | |
| Timer RD compare match B1 | 31 | Address set in DTCBAR register +1FH | |
| Timer RD compare match C1 | 32 | Address set in DTCBAR register +20H | |
| Timer RD compare match D1 | 33 | Address set in DTCBAR register +21H | |
| Timer RJ0 | 34 | Address set in DTCBAR register +22H | |
| Comparator detection 0 Note 1 | 35 | Address set in DTCBAR register +23H | |
| End of channel 0 of timer array unit 1 count or capture Note 2 | 36 | Address set in DTCBAR register +24H | |
| End of channel 1 of timer array unit 1 count or capture Note 2 | 37 | Address set in DTCBAR register +25H | |
| End of channel 2 of timer array unit 1 count or capture Note 2 | 38 | Address set in DTCBAR register +26H | |
| End of channel 3 of timer array unit 1 count or capture Note 2 | 39 | Address set in DTCBAR register +27H | |
| LIN1 reception end Note 3 | 40 | Address set in DTCBAR register +28H | |
| LIN1 transmission start/end Note 3 | 41 | Address set in DTCBAR register +29H | |
| End of channel 4 of timer array unit 1 count or capture Note 3 | 42 | Address set in DTCBAR register +2AH | |
| End of channel 5 of timer array unit 1 count or capture Note 3 | 43 | Address set in DTCBAR register +2BH | |
| End of channel 6 of timer array unit 1 count or capture Note 3 | 44 | Address set in DTCBAR register +2CH | |
| End of channel 7 of timer array unit 1 count or capture Note 3 | 45 | Address set in DTCBAR register +2DH | Lowest |

**Remark** Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory

Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory

Group C: RL78/F13 (LIN and CAN incorporated) products with 30, 32, 48, 64, or 80 pins and 32 Kbytes to 128 Kbytes of code flash memory

Group D: RL78/F14 products with 30, 32, 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory

Group E: RL78/F14 products with 48, 64, 80, or 100 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

**Notes 1.** Only in Group D and E products.

**2.** Only in Group B, C, D, and E products.

**3.** Only in Group E products.

### 19.2.4 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

When using the DTC, be sure to set bit 3 (DTCEN) to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 19-4. Format of Peripheral Enable Register 1 (PER1)**

Address: F02C0H After reset: 00H R/W

| Symbol | <7> | 6 | <5> | <4> | <3> | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| PER1 | DACEN Note 1 | 0 | CMPEN Note 1 | TRD0EN Note 2 | DTCEN | 0 | 0 | TRJ0EN |

| DTCEN | Control of DTC input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>• DTC cannot run. |
| 1 | Enables input clock supply.<br>• DTC can run. |

**Note 1.** Only in RL78/F14.

**2.** When FRQSEL4 = 1 in the user option byte (000C2H/020C2H), set $f_{CLK}$ to $f_{IH}$ before setting the bit 4 (TRD0EN) of the peripheral enable register 1 (PER1). When changing $f_{CLK}$ to a clock other than $f_{IH}$, clear the bit 4 (TRD0EN) of the peripheral enable register 1 (PER1) before changing.

**Caution Be sure to clear the following bits to 0.**
**RL78/F13: bits 1, 2, 5, 6, and 7**
**RL78/F14: bits 1, 2, and 6**

### 19.2.5 DTC Activation Enable Register i (DTCENi) (i = 0 to 5)

This is an 8-bit register which enables or disables DTC activation by interrupt sources. Table 19-6 lists the correspondence between interrupt sources and bits DTCENi0 to DTCENi7.

Set the DTCENi register by an 8-bit or 1-bit memory manipulation instruction.

**Notes 1.** Modify bits DTCENi0 to DTCENi7 if an activation source corresponding to the bit has not been generated.

**2.** Do not access the DTCENi register using a DTC transfer.

**Figure 19-5. DTC Activation Enable Register i (DTCENi) (i = 0 to 5)**

Address: F02E8H (DTCEN0), F02E9H (DTCEN1), F02EAH (DTCEN2), F02EBH (DTCEN3), F02ECH (DTCEN4), F02EDH (DTCEN5)[Note] After reset: 00H

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| DTCENi | DTCENi7 | DTCENi6 | DTCENi5 | DTCENi4 | DTCENi3 | DTCENi2 | DTCENi1 | DTCENi0 |

| DTCENi7 | DTC activation enable i7 | R/W |
|---|---|---|
| 0 | Activation disabled | R/W |
| 1 | Activation enabled | |
| The DTCENi7 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt. | | |

| DTCENi6 | DTC activation enable i6 | R/W |
|---|---|---|
| 0 | Activation disabled | R/W |
| 1 | Activation enabled | |
| The DTCENi6 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt. | | |

| DTCENi5 | DTC activation enable i5 | R/W |
|---|---|---|
| 0 | Activation disabled | R/W |
| 1 | Activation enabled | |
| The DTCENi5 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt. | | |

| DTCENi4 | DTC activation enable i4 | R/W |
|---|---|---|
| 0 | Activation disabled | R/W |
| 1 | Activation enabled | |
| The DTCENi4 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt. | | |

| DTCENi3 | DTC activation enable i3 | R/W |
|---|---|---|
| 0 | Activation disabled | R/W |
| 1 | Activation enabled | |
| The DTCENi3 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt. | | |

(**Note** is listed on the next page.)

| DTCENi2 | DTC activation enable i2 | R/W |
|---|---|---|
| 0 | Activation disabled | R/W |
| 1 | Activation enabled | |
| The DTCENi2 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt. | | |

| DTCENi1 | DTC activation enable i1 | R/W |
|---|---|---|
| 0 | Activation disabled | R/W |
| 1 | Activation enabled | |
| The DTCENi1 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt. | | |

| DTCENi0 | DTC activation enable i0 | R/W |
|---|---|---|
| 0 | Activation disabled | R/W |
| 1 | Activation enabled | |
| The DTCENi0 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt. | | |

**Note** Only in products of group E.

**Table 19-6. Correspondences between Interrupt Sources and Bits DTCENi0 to DTCENi7**

| Register | DTCENi7 Bit | DTCENi6 Bit | DTCENi5 Bit | DTCENi4 Bit | DTCENi3 Bit | DTCENi2 Bit | DTCENi1 Bit | DTCENi0 Bit |
|---|---|---|---|---|---|---|---|---|
| DTCEN0 | Reserved | INTP0 | INTP1 | INTP2 | INTP3 | INTP4 | INTP5 | INTP6 |
| DTCEN1 | Key input | A/D conversion end | UART0 reception transfer end/CSI01 transfer end or buffer empty/IIC01 transfer end | UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer end | UART1 reception transfer end/CSI11 transfer end or buffer empty/IIC11 transfer end | UART1 transmission transfer end/CSI10 transfer end or buffer empty/IIC10 transfer end | LIN0 reception end | LIN0 transmission start/transmission end |
| DTCEN2 | CAN reception end | Reserved | End of channel 0 of timer array unit 0 count or capture | End of channel 1 of timer array unit 0 count or capture | End of channel 2 of timer array unit 0 count or capture | End of channel 3 of timer array unit 0 count or capture | End of channel 4 of timer array unit 0 count or capture | End of channel 5 of timer array unit 1 count or capture |
| DTCEN3 | End of channel 6 of timer array unit 0 count or capture | End of channel 7 of timer array unit 0 count or capture | Timer RD compare match A0 | Timer RD compare match B0 | Timer RD compare match C0 | Timer RD compare match D0 | Timer RD compare match A1 | Timer RD compare match B1 |
| DTCEN4 | Timer RD compare match C1 | Timer RD compare match D1 | Timer RJ0 | Comparator detection 0 | End of channel 0 of timer array unit 1 count or capture | End of channel 1 of timer array unit 1 count or capture | End of channel 2 of timer array unit 1 count or capture | End of channel 3 of timer array unit 1 count or capture |
| DTCEN5 Note | LIN1 reception end | LIN1 transmission start/transmission end | End of channel 4 of timer array unit 1 count or capture | End of channel 5 of timer array unit 1 count or capture | End of channel 6 of timer array unit 1 count or capture | End of channel 7 of timer array unit 1 count or capture | Reserved | Reserved |

**Note** Only in products of group E.

**Remark** i = 0 to 5

### 19.2.6 DTC Base Address Register (DTCBAR)

This is an 8-bit register used to set the following addresses: the vector address where the start address of the DTC control data area is stored and the address of the DTC control data area. The value of the DTCBAR register is handled as the higher 8 bits to generate a 16-bit address.

**Cautions 1. Modify the DTCBAR register value with all DTC activation sources set to activation disabled.**
**2. Do not rewrite the DTCBAR register more than once.**
**3. Do not access the DTCBAR register using a DTC transfer.**
**4. For the allocation of the DTC control data area and the DTC vector table area, refer to the notes on 19.2.1 Allocation of DTC Control Data Area and DTC Vector Table Area.**

**Figure 19-6. Format of DTC Base Address Register (DTCBAR)**

Address: F02E0H After reset: FDH

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DTCBAR | DTCBAR7 | DTCBAR6 | DTCBAR5 | DTCBAR4 | DTCBAR3 | DTCBAR2 | DTCBAR1 | DTCBAR0 |

### 19.2.7 DTC Control Register j (DTCCRj) (j = 0 to 23)

The DTCCRj register is used to control the DTC operating mode.

**Figure 19-7. Format of DTC Control Register j (DTCCRj)**

Address: Refer to **19.2.2 DTC Control Data Allocation**. After reset: Undefined

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DTCCRj | 0 | SZ | RPTINT | CHNE | DAMOD | SAMOD | RPTSEL | MODE |

| Bit 7 | Reserved | R/W |
|---|---|---|
| 0 | Set to 0. | R/W |

| SZ | Data size selection | R/W |
|---|---|---|
| 0 | 8 bits | R/W |
| 1 | 16 bits | |

| RPTINT | Enabling/disabling repeat mode interrupts | R/W |
|---|---|---|
| 0 | Interrupt generation disabled | R/W |
| 1 | Interrupt generation enabled | |
| The setting of the RPTINT bit is invalid when the MODE bit is 0 (normal mode). | | |

| CHNE | Enabling/disabling chain transfers | R/W |
|---|---|---|
| 0 | Chain transfers disabled | R/W |
| 1 | Chain transfers enabled | |
| Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled). | | |

| DAMOD | Transfer destination address control | R/W |
|---|---|---|
| 0 | Fixed | R/W |
| 1 | Incremented | |
| The setting of the DAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 0 (transfer destination is the repeat area). | | |

| SAMOD | Transfer source address control | R/W |
|---|---|---|
| 0 | Fixed | R/W |
| 1 | Incremented | |
| The setting of the SAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 1 (transfer source is the repeat area). | | |

| RPTSEL | Repeat area selection | R/W |
|---|---|---|
| 0 | Transfer destination is the repeat area | R/W |
| 1 | Transfer source is the repeat area | |
| The setting of the RPTSEL bit is invalid when the MODE bit is 0 (normal mode). | | |

| MODE | Transfer mode selection | R/W |
|---|---|---|
| 0 | Normal mode | R/W |
| 1 | Repeat mode | |

**Caution Do not access the DTCCRj register using a DTC transfer.**

### 19.2.8 DTC Block Size Register j (DTBLSj) (j = 0 to 23)

This register is used to set the block size of the data to be transferred by one activation.

**Figure 19-8. Format of DTC Block Size Register j (DTBLSj)**

Address: Refer to **19.2.2 DTC Control Data Allocation**. After reset: Undefined

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DTBLSj | DTBLSj7 | DTBLSj6 | DTBLSj5 | DTBLSj4 | DTBLSj3 | DTBLSj2 | DTBLSj1 | DTBLSj0 |

| DTBLSj | Transfer Block Size | | R/W |
|---|---|---|---|
| | 8-Bit Transfer | 16-Bit Transfer | |
| 00H | 256 bytes | 512 bytes | R/W |
| 01H | 1 byte | 2 bytes | |
| 02H | 2 bytes | 4 bytes | |
| 03H | 3 bytes | 6 bytes | |
| . | . | . | |
| . | . | . | |
| . | . | . | |
| FDH | 253 bytes | 506 bytes | |
| FEH | 254 bytes | 508 bytes | |
| FFH | 255 bytes | 510 bytes | |

**Caution Do not access the DTBLSj register using a DTC transfer.**

### 19.2.9 DTC Transfer Count Register j (DTCCTj) (j = 0 to 23)

This register is used to set the number of DTC data transfers. The value is decremented by 1 each time DTC transfer is activated once.

**Figure 19-9. Format of DTC Transfer Count Register j (DTCCTj)**

Address: Refer to **19.2.2 DTC Control Data Allocation**. After reset: Undefined

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DTCCTj | DTCCTj7 | DTCCTj6 | DTCCTj5 | DTCCTj4 | DTCCTj3 | DTCCTj2 | DTCCTj1 | DTCCTj0 |

| DTCCTj | Number of Transfers | R/W |
|---|---|---|
| 00H | 256 times | R/W |
| 01H | Once | |
| 02H | 2 times | |
| 03H | 3 times | |
| . | . | |
| . | . | |
| . | . | |
| FDH | 253 times | |
| FEH | 254 times | |
| FFH | 255 times | |

**Caution Do not access the DTCCTj register using a DTC transfer.**

### 19.2.10 DTC Transfer Count Reload Register j (DTRLDj) (j = 0 to 23)

This register is used to set the initial value of the transfer count register in repeat mode. Since the value of this register is reloaded to the DTCCT register in repeat mode, set the same value as the initial value of the DTCCT register.

**Figure 19-10. Format of DTC Transfer Count Reload Register j (DTRLDj)**

Address: Refer to **19.2.2 DTC Control Data Allocation**. After reset: Undefined

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| DTRLDj | DTRLDj7 | DTRLDj6 | DTRLDj5 | DTRLDj4 | DTRLDj3 | DTRLDj2 | DTRLDj1 | DTRLDj0 |

**Caution Do not access the DTRLDj register using a DTC transfer.**

### 19.2.11 DTC Source Address Register j (DTSARj) (j = 0 to 23)

This register is used to specify the transfer source address for data transfer.

When the SZ bit in the DTCCRj register is set to 1 (16-bit transfer), the lowest bit is ignored and the address is handled as an even address.

**Figure 19-11. Format of DTC Source Address Register j (DTSARj)**

Address: Refer to **19.2.2 DTC Control Data Allocation**. After reset: Undefined

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DTSARj | DTS ARj15 | DTS ARj14 | DTS ARj13 | DTS ARj12 | DTS ARj11 | DTS ARj10 | DTS ARj9 | DTS ARj8 | DTS ARj7 | DTS ARj6 | DTS ARj5 | DTS ARj4 | DTS ARj3 | DTS ARj2 | DTS ARj1 | DTS ARj0 |

**Cautions 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.**

**2. Do not access the DTSARj register using a DTC transfer.**

### 19.2.12 DTC Destination Address Register j (DTDARj) (j = 0 to 23)

This register is used to specify the transfer destination address for data transfer.

When the SZ bit in the DTCCRj register is set to 1 (16-bit transfer), the lowest bit is ignored and the address is handled as an even address.

**Figure 19-12. Format of DTC Destination Address Register j (DTDARj)**

Address: Refer to **19.2.2 DTC Control Data Allocation**. After reset: Undefined

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DTDARj | DTD ARj15 | DTD ARj14 | DTD ARj13 | DTD ARj12 | DTD ARj11 | DTD ARj10 | DTD ARj9 | DTD ARj8 | DTD ARj7 | DTD ARj6 | DTD ARj5 | DTD ARj4 | DTD ARj3 | DTD ARj2 | DTD ARj1 | DTD ARj0 |

**Cautions 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.**

**2. Do not access the DTDARj register using a DTC transfer.**

### 19.2.13 High-speed DTC Channel Select Register 0 (SELHS0)

The SELHS0 register is used to select the high-speed DTC channel.

**Figure 19-13. Format of High-speed DTC Channel Select Register 0 (SELHS0)**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| SELHS0 | 0 | 0 | SELHS05 | SELHS04 | SELHS03 | SELHS02 | SELHS01 | SELHS00 |

The SELHS0 register can be set with an 8-bit or 1-bit memory manipulation instruction, not set with a 16-bit memory manipulation instruction.

After reset: 3FH

Address: F02E1H

Correspondence between the activation sources and SELHS0i (i = 0 to 5) bits is shown below.

| SELHS05 to SELHS00 | | | | | | Description |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | Activation source number 0 is selected as high-speed channel 0. |
| 0 | 0 | 0 | 0 | 0 | 1 | Activation source number 1 is selected as high-speed channel 0. |
| 0 | 0 | 0 | 0 | 1 | 1 | Activation source number 2 is selected as high-speed channel 0. |
| : | | | | | | : |
| : | | | | | | : |
| 1 | 0 | 1 | 0 | 1 | 1 | Activation source number 43 is selected as high-speed channel 0. |
| 1 | 0 | 1 | 1 | 0 | 0 | Activation source number 44 is selected as high-speed channel 0. |
| 1 | 0 | 1 | 1 | 0 | 1 | Activation source number 45 is selected as high-speed channel 0. |
| 1 | 1 | 1 | 1 | 1 | 1 | High-speed channel 0 is not used. |
| Other than above | | | | | | Setting prohibited |

**Cautions 1. Modify the data of the SELHS0 register when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 5[Note]) register is 0 (DTC activation disabled).**

**2. Do not access the SELHS0 register using a DTC transfer.**

**Note** Products of groups A, B, C, and D: i = 0 to 4
Products of group E: i = 0 to 5

### 19.2.14 High-speed DTC Channel Select Register 1 (SELHS1)

The SELHS1 register is an 8-bit register that is used to select the high-speed DTC channel.

**Figure 19-14. Format of High-speed DTC Channel Select Register 1 (SELHS1)**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| SELHS1 | 0 | 0 | SELHS15 | SELHS14 | SELHS13 | SELHS12 | SELHS11 | SELHS10 |

The SELHS1 register can be set with an 8-bit or 1-bit memory manipulation instruction, not set with a 16-bit memory manipulation instruction.

After reset: 3FH

Address: F02E2H

Correspondence between the activation sources and SELHS1i (i = 0 to 5) bits is shown below.

| SELHS15 to SELHS10 | | | | | | Description |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | Activation source number 0 is selected as high-speed channel 1. |
| 0 | 0 | 0 | 0 | 0 | 1 | Activation source number 1 is selected as high-speed channel 1. |
| 0 | 0 | 0 | 0 | 1 | 1 | Activation source number 2 is selected as high-speed channel 1. |
| : | | | | | | : |
| | | | | | | : |
| 1 | 0 | 1 | 0 | 1 | 1 | Activation source number 43 is selected as high-speed channel 1. |
| 1 | 0 | 1 | 1 | 0 | 0 | Activation source number 44 is selected as high-speed channel 1. |
| 1 | 0 | 1 | 1 | 0 | 1 | Activation source number 45 is selected as high-speed channel 1. |
| 1 | 1 | 1 | 1 | 1 | 1 | High-speed channel 1 is not used. |
| Other than above | | | | | | Setting prohibited |

**Cautions 1. Modify the data of the SELHS1 register when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 5[Note]) register is 0 (DTC activation disabled).**

**2. Do not access the SELHS1 register using a DTC transfer.**

**Note** Products of groups A, B, C, and D: i = 0 to 4
Products of group E: i = 0 to 5

### 19.2.15 High-speed DTC Control Register m (HDTCCR0/1) (m = 0, 1)

The HDTCCRm register is used to control the high-speed DTC transfer operating mode.

**Figure 19-15. Format of High-speed DTC Control Register m (HDTCCRm)**

Address: F02D0H (HDTCCR0), F02D8H (HDTCCR1) After reset: 00H

| Symbol | 7 | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| HDTCCRm | 0 | HSZm | HRPTINTm | HCHNEm | HDAMODm | HSAMODm | HRPTSELm | HMODEm |

| Bit 7 | Reserved | R/W |
|---|---|---|
| 0 | Set to 0. | R/W |

| HSZm | Data size selection | R/W |
|---|---|---|
| 0 | 8 bits | R/W |
| 1 | 16 bits | |

| HRPTINTm | Enabling/disabling repeat mode interrupts | R/W |
|---|---|---|
| 0 | Interrupt generation disabled | R/W |
| 1 | Interrupt generation enabled | |
| The setting of the HRPTINTm bit is invalid when the HMODEm bit is 0 (normal mode). | | |

| HCHNEm | Enabling/disabling chain transfers | R/W |
|---|---|---|
| 0 | Chain transfers disabled | R/W |
| 1 | Chain transfers enabled | |
| Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).<br>Set the HCHNEm bit to 0 (chain transfers disabled) when the activation source number is set to the maximum value by the SELHSm register. | | |

| HDAMODm | Transfer destination address control | R/W |
|---|---|---|
| 0 | Fixed | R/W |
| 1 | Incremented | |
| The setting of the HDAMODm bit is invalid when the HMODEm bit is 1 (repeat mode) and the HRPTSELm bit is 0 (transfer destination is the repeat area). | | |

| HSAMODm | Transfer source address control | R/W |
|---|---|---|
| 0 | Fixed | R/W |
| 1 | Incremented | |
| The setting of the HSAMODm bit is invalid when the HMODEm bit is 1 (repeat mode) and the HRPTSELm bit is 1 (transfer source is the repeat area). | | |

| HRPTSELm | Repeat area selection | R/W |
|---|---|---|
| 0 | Transfer destination is the repeat area | R/W |
| 1 | Transfer source is the repeat area | |
| The setting of the HRPTSELm bit is invalid when the HMODEm bit is 0 (normal mode). | | |

| HMODEm | Transfer mode selection | R/W |
|---|---|---|
| 0 | Normal mode | R/W |
| 1 | Repeat mode | |

**Caution Do not access the HDTCCRm register using a high-speed DTC transfer.**

### 19.2.16 High-speed DTC Transfer Count Register m (HDTCCT0/1) (m = 0, 1)

This register is used to set the number of high-speed DTC data transfers. The value is decremented by 1 each time DTC transfer is activated once.

**Figure 19-16. Format of High-speed DTC Transfer Count Register m (HDTCCTm)**

Address: F02D2H (HDTCCT0), F02DAH (HDTCCT1) After reset: 00H

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| HDTCCTm | HDTCCTm7 | HDTCCTm6 | HDTCCTm5 | HDTCCTm4 | HDTCCTm3 | HDTCCTm2 | HDTCCTm1 | HDTCCTm0 |

| HDTCCTm | Number of Transfers | R/W |
|---|---|---|
| 00H | 256 times | R/W |
| 01H | Once | |
| 02H | 2 times | |
| 03H | 3 times | |
| . | . | |
| . | . | |
| . | . | |
| FDH | 253 times | |
| FEH | 254 times | |
| FFH | 255 times | |

**Caution Do not access the HDTCCTm register using a high-speed DTC transfer.**

### 19.2.17 DTC Transfer Count Reload Register m (HDTRLD0/1) (m = 0, 1)

This register is used to set the initial value of the transfer count register in repeat mode. Since the value of this register is reloaded to the HDTCCTm register in repeat mode, set the same value as the initial value of the HDTCCTm register.

**Figure 19-17. Format of High-speed DTC Transfer Count Reload Register m (HDTRLDm)**

Address: F02D3H (HDTRLD0), F02DBH (HDTRLD1) After reset: 00H

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| HDTRLDm | HDTRLDm7 | HDTRLDm6 | HDTRLDm5 | HDTRLDm4 | HDTRLDm3 | HDTRLDm2 | HDTRLDm1 | HDTRLDm0 |

**Caution Do not access the HDTRLDm register using a high-speed DTC transfer.**

### 19.2.18 High-speed DTC Source Address Register m (HDTSAR0/1) (m = 0, 1)

Only the 1st SFR area and the 2nd SFR area can be set for the source address for high-speed transfer.

Set the lower 12-bit addresses of the HDTSARm.

The higher 4 bits are read as 0.

**Figure 19-18. Format of High-speed DTC Source Address Register m (HDTSARm)**

Address: F02D4H, F02D5H (HDTSAR0), F02DCH, F02DDH (HDTSAR1) After reset: 0000H

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HDTS ARm | 0 | 0 | 0 | 0 | HDTS ARm11 | HDTS ARm10 | HDTS ARm9 | HDTS ARm8 | HDTS ARm7 | HDTS ARm6 | HDTS ARm5 | HDTS ARm4 | HDTS ARm3 | HDTS ARm2 | HDTS ARm1 | HDTS ARm0 |

**Cautions 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.**

**2. Do not access the HDTSARm register using a high-speed DTC transfer.**

### 19.2.19 High-speed DTC Destination Address Register m (HDTDAR0/1) (m = 0, 1)

This register is used to specify the transfer destination address for data transfer.

**Figure 19-19. Format of High-speed DTC Destination Address Register m (HDTDARm)**

Address: F02D6H, F02D7H (HDTDAR0), F02DEH, F02DFH (HDTDAR1) After reset: 0000H

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HDTD ARm | HDTD ARm15 | HDTD ARm14 | HDTD ARm13 | HDTD ARm12 | HDTD ARm11 | HDTD ARm10 | HDTD ARm9 | HDTD ARm8 | HDTD ARm7 | HDTD ARm6 | HDTD ARm5 | HDTD ARm4 | HDTD ARm3 | HDTD ARm2 | HDTD ARm1 | HDTD ARm0 |

**Cautions 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer destination address.**

**2. Do not access the HDTDARm register using a high-speed DTC transfer.**

## 19.3 Operation

When the DTC is activated, control data is read from the DTC control data area to perform data transfers and control data after data transfer is written back to the DTC control data area. Twenty-four sets of control data can be stored in the DTC control data area, which allows 24 types of data transfers to be performed.

There are two transfer modes (normal mode and repeat mode) and two transfer sizes (8-bit transfer and 16-bit transfer). When the CHNE bit in the DTCCRj (j = 0 to 23) register is set to 1 (chain transfers enabled), multiple control data is read and data transfers are continuously performed by one activation source (chain transfers).

A transfer source address is specified by the 16-bit register DTSARj, and a transfer destination address is specified by the 16-bit register DTDARj.

The values in registers DTSARj and DTDARj are separately incremented or fixed according to the control data after the data transfer.

This product supports high-speed transfer operation. The high-speed transfer can be realized by allocating the dedicated control data to the SFR area instead of the RAM area. To perform basic operation, normal mode requires five clock cycles to read vector and control data, while high-speed transfer requires one cycle. In addition, to write back control data, normal mode requires a maximum of three clock cycles, while high-speed transfer requires one cycle.

### 19.3.1 Activation Sources

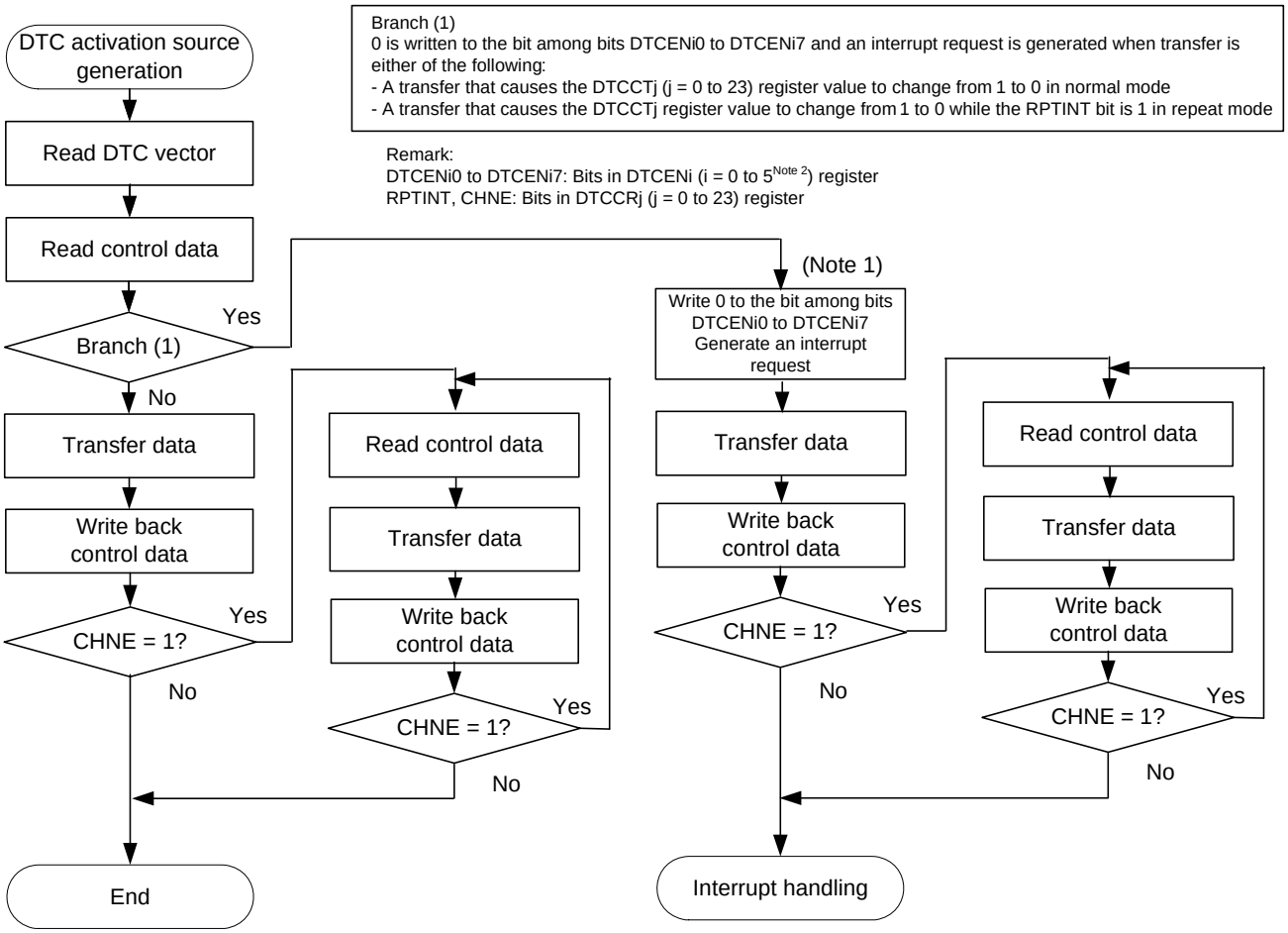
The DTC is activated by an interrupt signal from the peripheral functions. The interrupt signals to activate the DTC are selected with the DTCENi (i = 0 to 5[Note 2]) register.

The DTC sets the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi register to 0 (activation disabled) during operation when the setting of data transfer (the first transfer in chain transfers) is either of the following:

- A transfer that causes the DTCCTj (j = 0 to 23) register value to change to 0 in normal mode
- A transfer that causes the DTCCTj register value to change to 0 while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled) in repeat mode

Figure 19-20 shows the DTC internal operation flowchart.

**Figure 19-20. DTC Internal Operation Flowchart**




**Notes 1.** 0 is not written to the bit among bits DTCENi0 to DTCENi7 for data transfers activated by the setting to enable chain transfers (the CHNE bit is 1). Also, no interrupt request is generated.

**2.** Products of groups A, B, C, and D: i = 0 to 4
Products of group E: i = 0 to 5

### 19.3.2 Normal Mode

One to 256 bytes of data are transferred by one activation during 8-bit transfer and 2 to 512 bytes during 16-bit transfer. The number of transfers can be 1 to 256 times. When the data transfer causing the DTCCTj (j = 0 to 23) register value to change to 0 is performed, the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 5[Note]) in the DTCENi register to 0 (activation disabled).

Table 19-7 shows register functions in normal mode. Figure 19-21 shows data transfers in normal mode.

**Note** Products of groups A, B, C, and D: i = 0 to 4
Products of group E: i = 0 to 5

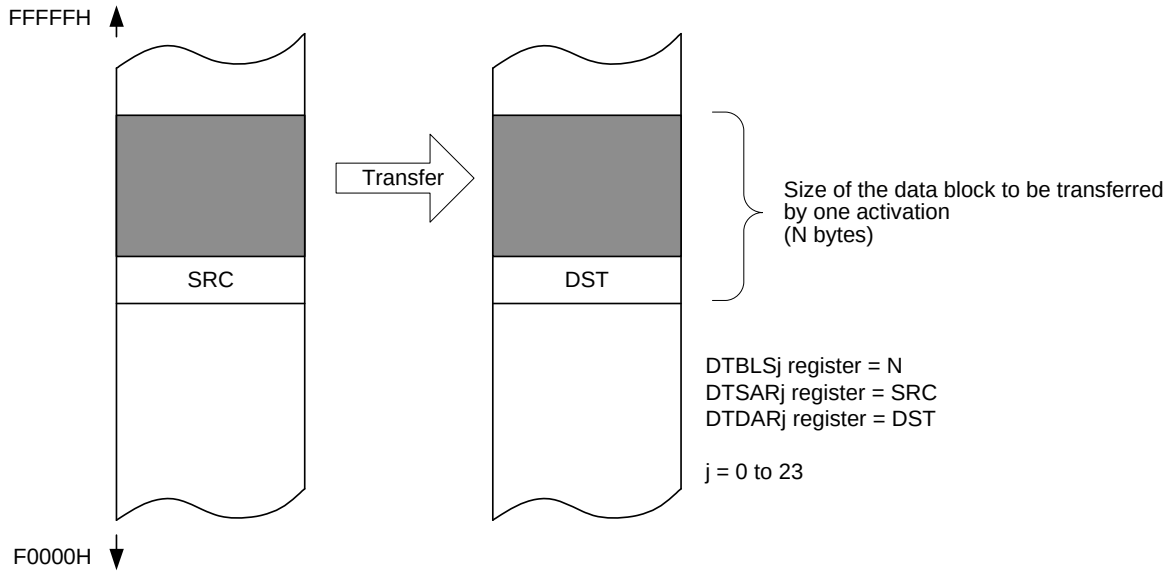
**Table 19-7. Register Functions in Normal Mode**

| Register Name | Symbol | Function |
|---|---|---|
| DTC block size register j | DTBLSj | Size of the data block to be transferred by one activation |
| DTC transfer count register j | DTCCTj | Number of data transfers |
| DTC transfer count reload register j | DTRLDj | Not used [Note] |
| DTC source address register j | DTSARj | Data transfer source address |
| DTC destination address register j | DTDARj | Data transfer destination address |

**Note** Initialize DTRLDj register with any desired value because the control data are read.

**Remark** j = 0 to 23

**Figure 19-21. Data Transfers in Normal Mode**




| DTCCR Register Setting | | | | Source Address Control | Destination Address Control | Source Address after Transfer | Destination Address after Transfer |
|---|---|---|---|---|---|---|---|
| DAMOD | SAMOD | RPTSEL | MODE | | | | |
| 0 | 0 | X | 0 | Fixed | Fixed | SRC | DST |
| 0 | 1 | X | 0 | Incremented | Fixed | SRC + N | DST |
| 1 | 0 | X | 0 | Fixed | Incremented | SRC | DST + N |
| 1 | 1 | X | 0 | Incremented | Incremented | SRC + N | DST + N |

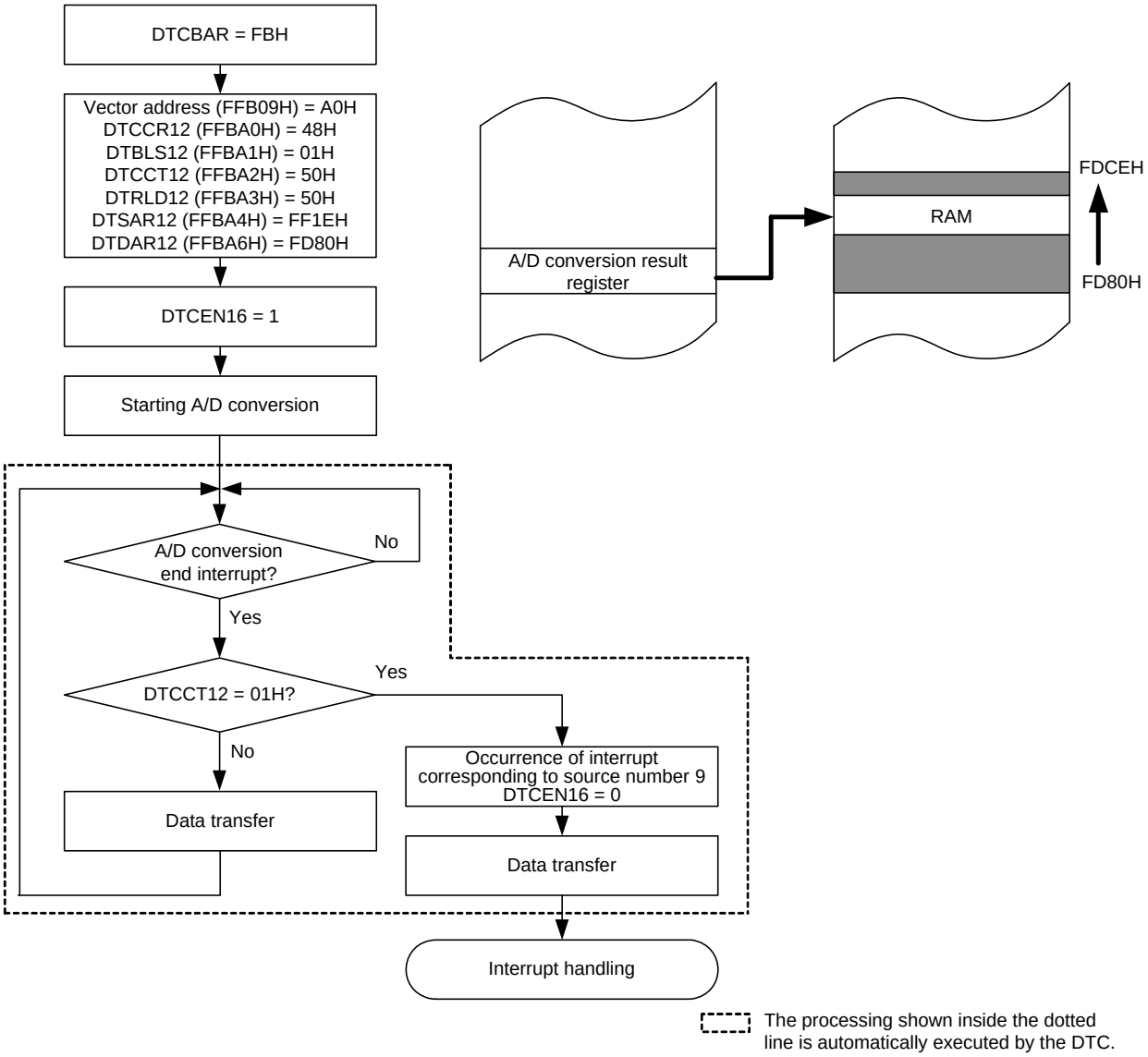
X: 0 or 1

**(1) Example 1 of using normal mode: Consecutively capturing A/D conversion results**

The DTC is activated by an A/D conversion end interrupt and the value of the A/D conversion result register is transferred to RAM.

- The vector address is FFB09H and control data is allocated at FFBA0H to FFBA7H.
- An A/D interrupt is assigned to source number 9.
- Transfers 2-byte data of the A/D conversion result register (FFF1EH, FFF1FH) to 80 bytes of FFD80H to FFDCFH of RAM.

**Figure 19-22. Example 1 of Using Normal Mode: Consecutively Capturing A/D Conversion Results**




The data of DTRLD12 does not affect DTC transfer operation because of normal mode.

### 19.3.3 Repeat Mode

One to 255 bytes of data are transferred by one activation. Either of the transfer source or destination should be specified as the repeat area. The number of transfers can be 1 to 255 times. On completion of the specified number of transfers, the DTCCTj (i = 0 to 23) register and the address specified for the repeat area are initialized to continue transfers. When the data transfer causing the DTCCTj register value to change to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 5 Note) to 0 (activation disabled). When the RPTINT bit in the DTCCRj register is 0 (interrupt generation disabled), no interrupt request is generated even if the data transfer causing the DTCCTj register value to change to 0 is performed. Also, bits DTCENi0 to DTCENi7 are not set to 0.

Table 19-8 lists register functions in repeat mode. Figure 19-23 shows data transfers in repeat mode.

**Note** Products of groups A, B, C, and D: i = 0 to 4
Products of group E: i = 0 to 5

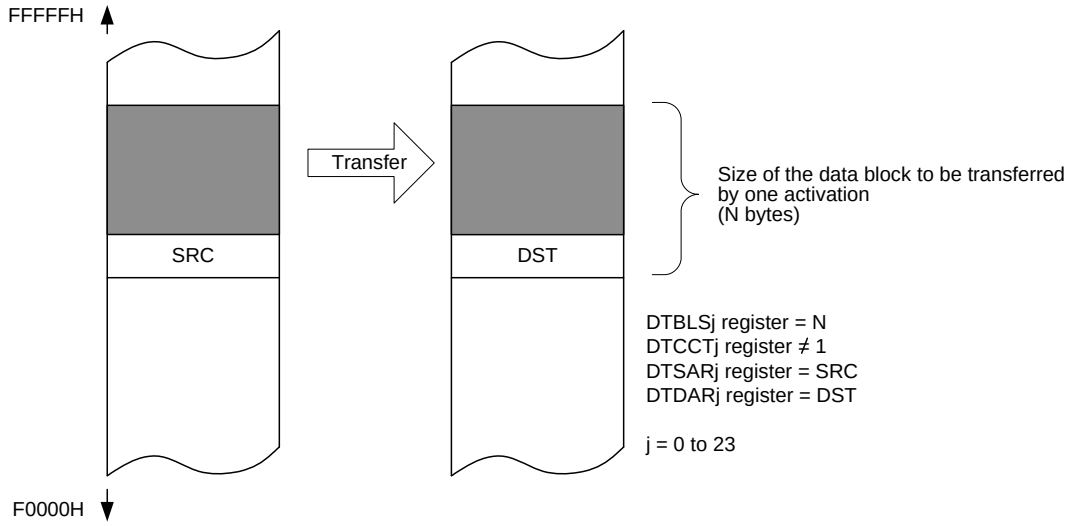
**Table 19-8. Register Functions in Repeat Mode**

| Register Name | Symbol | Function |
|---|---|---|
| DTC block size register j | DTBLSj | Size of the data block to be transferred by one activation |
| DTC transfer count register j | DTCCTj | Number of data transfers |
| DTC transfer count reload register j | DTRLDj | This register value is reloaded to the DTCCT register (the number of transfers is initialized). |
| DTC source address register j | DTSARj | Data transfer source address |
| DTC destination address register j | DTDARj | Data transfer destination address |

**Remark** j = 0 to 23

**Figure 19-23. Data Transfers in Repeat Mode**

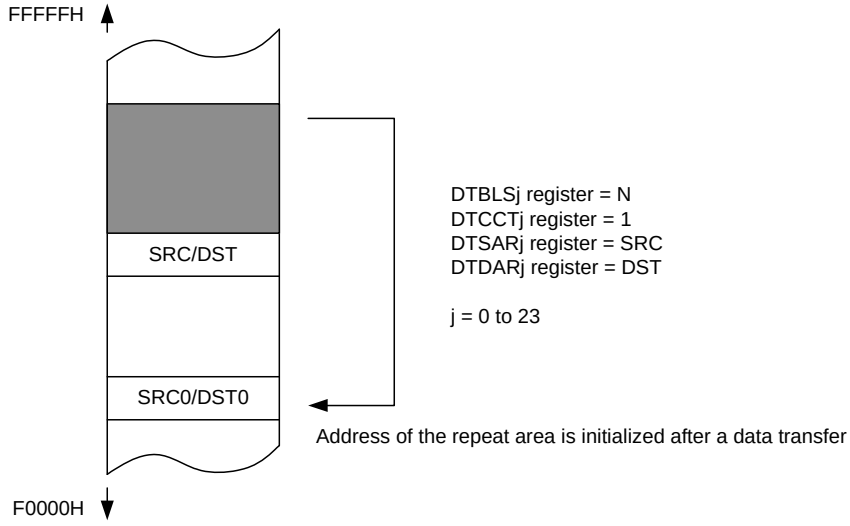
DTCCTj register ≠ 1




| DTCCR Register Setting | | | | Source Address Control | Destination Address Control | Source Address after Transfer | Destination Address after Transfer |
|---|---|---|---|---|---|---|---|
| DAMOD | SAMOD | RPTSEL | MODE | | | | |
| 0 | X | 1 | 1 | Repeat area | Fixed | SRC + N | DST |
| 1 | X | 1 | 1 | Repeat area | Incremented | SRC + N | DST + N |
| X | 0 | 0 | 1 | Fixed | Repeat area | SRC | DST + N |
| X | 1 | 0 | 1 | Incremented | Repeat area | SRC + N | DST + N |

X: 0 or 1

DTCCTj register = 1




| DTCCR Register Setting | | | | Source Address Control | Destination Address Control | Source Address after Transfer | Destination Address after Transfer |
|---|---|---|---|---|---|---|---|
| DAMOD | SAMOD | RPTSEL | MODE | | | | |
| 0 | X | 1 | 1 | Repeat area | Fixed | SRC0 | DST |
| 1 | X | 1 | 1 | Repeat area | Incremented | SRC0 | DST + N |
| X | 0 | 0 | 1 | Fixed | Repeat area | SRC | DST0 |
| X | 1 | 0 | 1 | Incremented | Repeat area | SRC + N | DST0 |

SRC0: Initial source address value
DST0: Initial destination address value
X: 0 or 1

**Cautions 1. When repeat mode is used, the lower 8 bits of the initial value for the repeat area address must be 00H.**

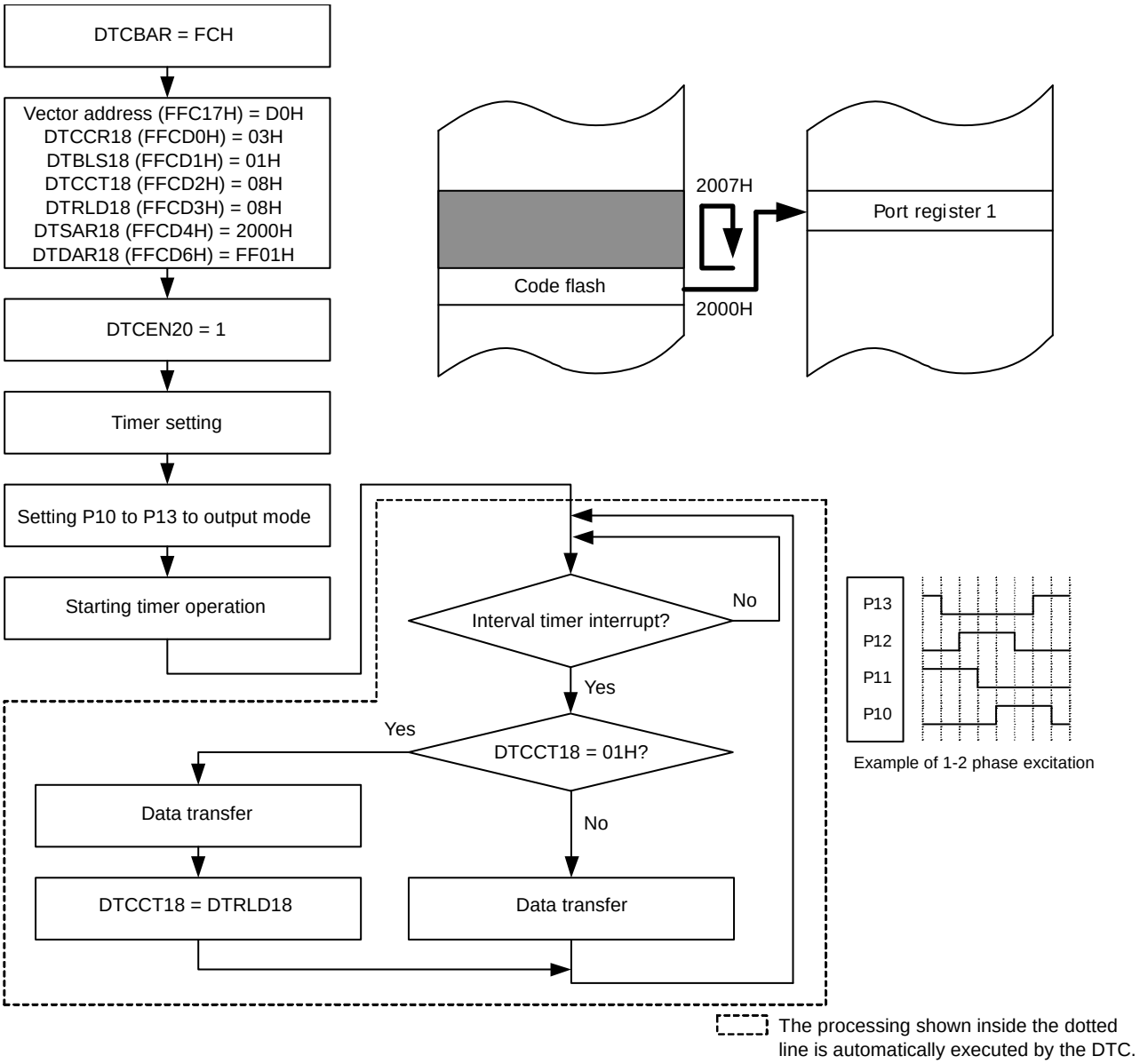
**2. When repeat mode is used, the data size of the repeat area must be set to 255 bytes or less.**

**(1) Example 1 of using repeat mode: Outputting a stepping motor control pulse using ports**

The DTC is activated by an interval timer interrupt and the pattern of the motor control pulse stored in the code flash memory is transferred to general-purpose ports.

- The vector address is FFC17H and control data is allocated at FFCD0H to FFCD7H.
- The timer interrupt is assigned to source number 23.
- Transfers 8-byte data of 02000H to 02007H of the code flash memory from the mirror space (F2000H to F2007H) to port register 1 (FFF01H).
- A repeat mode interrupt is disabled.

**Figure 19-24. Example 1 of Using Repeat Mode: Outputting a Stepping Motor Control Pulse Using Ports**




To stop the output, stop the timer first and then clear DTCEN20.

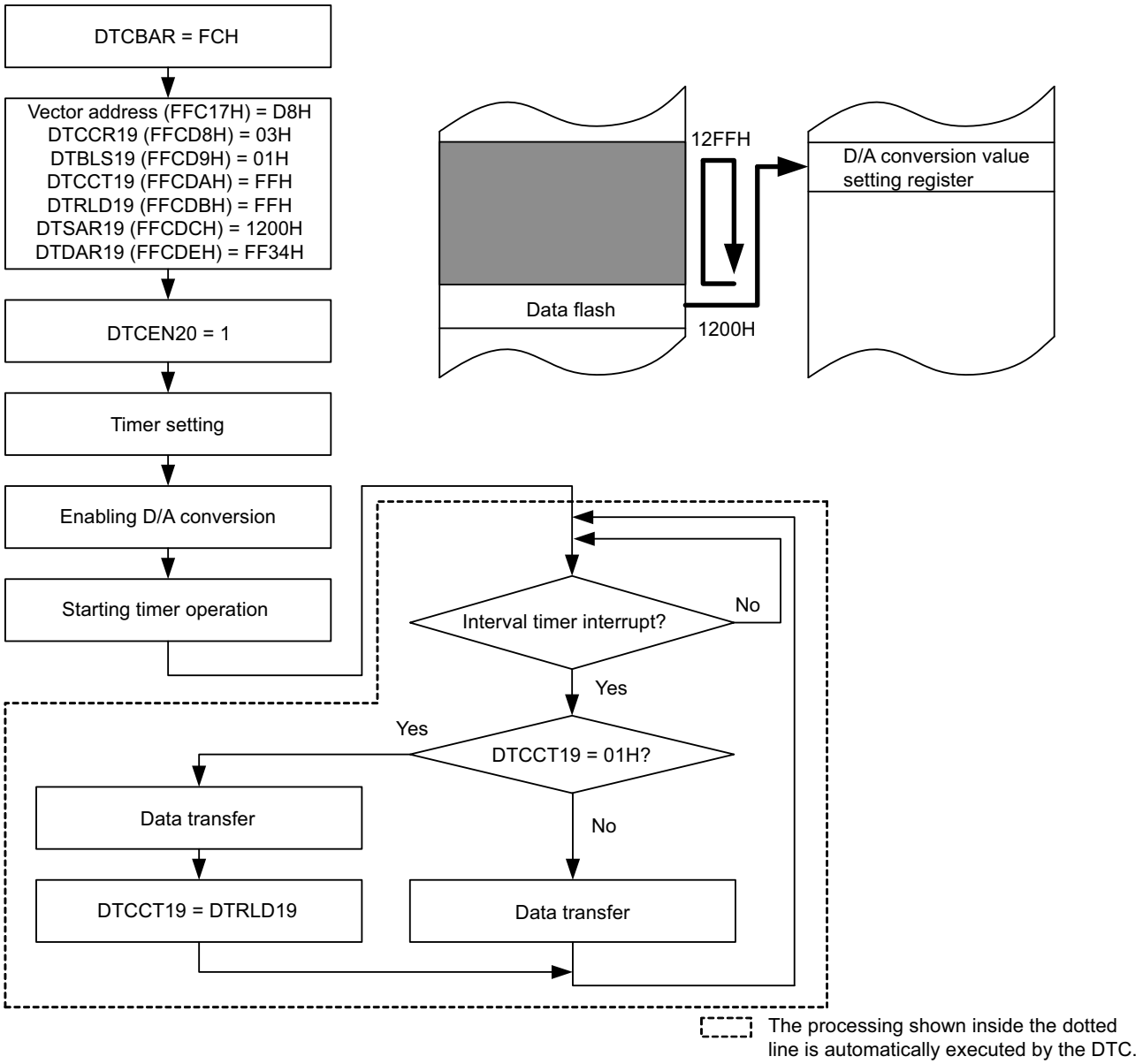
**(2) Example 2 of using repeat mode: Outputting a sine wave using the 8-bit D/A converter**

The DTC is activated by an interval timer interrupt and the table of the sine wave stored in the data flash memory is transferred to the 8-bit D/A conversion value setting register.

The timer interval time is set to the D/A output setup time.

- The vector address is FFC17H and control data is allocated at FFCD8H to FFCDFH.
- The timer interrupt is assigned to source bumber 23.
- Transfers 255-byte data of F1200H to F12FEH of the data flash memory to the D/A conversion value setting register (FFF34H).
- A repeat mode interrupt is disabled.

**Figure 19-25. Example 2 of Using Repeat Mode: Outputting a Sine Wave Using the 8-bit D/A Converter**




To stop the output, stop the timer first and then clear DTCEN20.

**Caution A D/A converter is provided in the products with 96 Kbytes or more of code flash memory.**

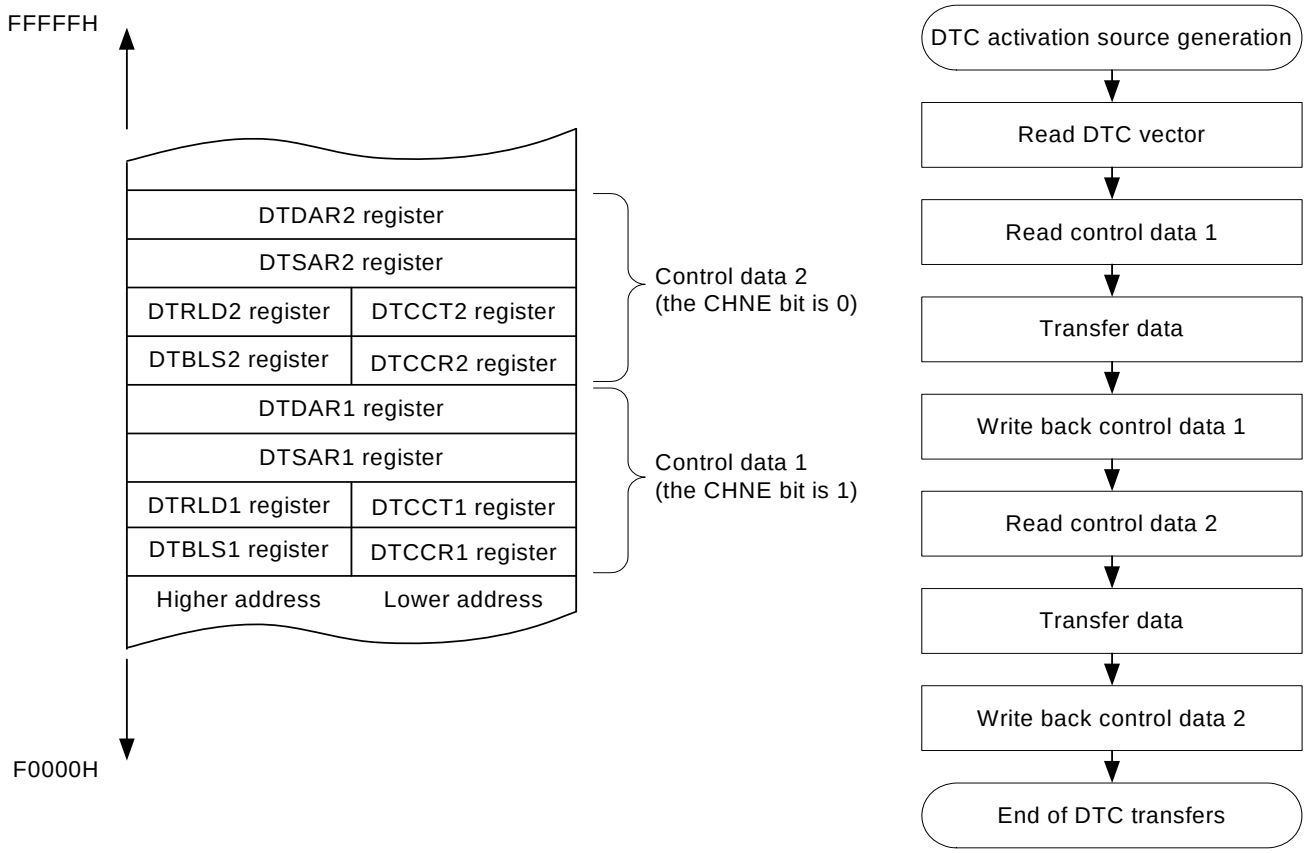
### 19.3.4 Chain Transfers

When the CHNE bit in the DTCCRj (j = 0 to 22) register is 1 (chain transfers enabled), multiple data transfers can be continuously performed by one activation source.

When the DTC is activated, one control data is selected according to the data read from the DTC vector address corresponding to the activation source, and the selected control data is read from the DTC control data area. When the CHNE bit for the control data is 1 (chain transfers enabled), the next control data immediately following the current control data is read and transferred after the current transfer is completed. This operation is repeated until the data transfer with the control data for which the CHNE bit is 0 (chain transfers disabled) is completed.

Figure 19-26 shows data transfers in chain transfers.

**Figure 19-26. Data Transfers during Chain Transfers**

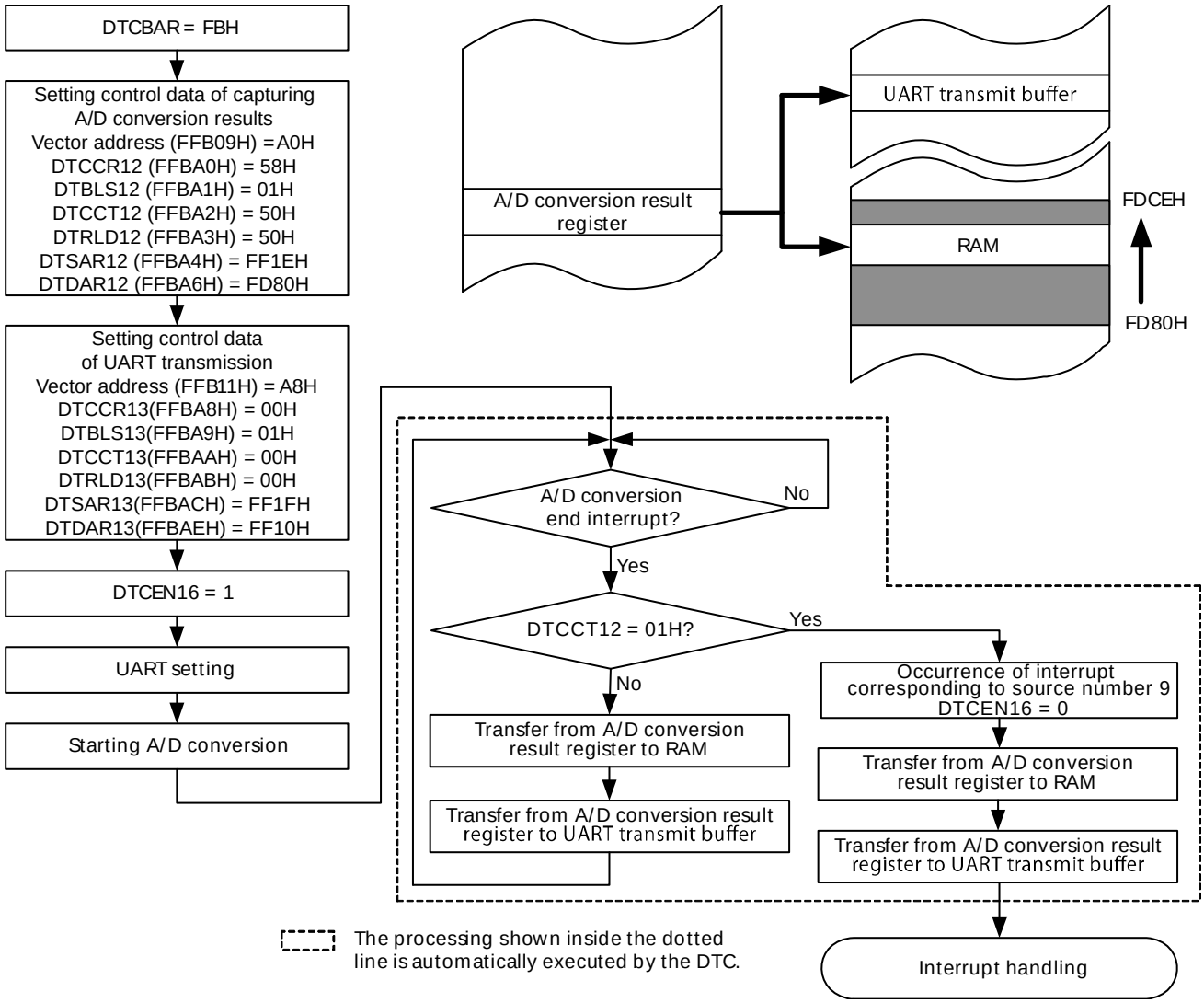


**(1) Example of using chain transfers: Consecutively capturing A/D conversion results and UART transmission**

The DTC is activated by an A/D conversion end interrupt and A/D conversion results are transferred to RAM, and then transmitted using the UART.

- The vector address is FFB09H.
- Control data of capturing A/D conversion results is allocated at FFBA0H to FFBA7H.
- Control data of UART transmission is allocated at FFBA8H at FFBAFH.
- An A/D conversion end interrupt is assigned to source number 9.
- Transfers 2-byte data of the A/D conversion result register (FFF1FH, FFF1EH) to FFD80H to FFDCFH of RAM, and transfers the upper 1 byte (FFF1FH) of the A/D conversion result register to the UART transmit buffer (FFF10H).

**Figure 19-27. Example of Using Chain Transfers: Consecutively Capturing A/D Conversion Results and UART Transmission**




**Cautions 1. Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).**

**2. For chain transfer, in the second and subsequent data transfers, bits DTCENi0 to DTCENi7 (i = 0 to 5[Note]) in the DTCENi register are not set to 0 (DTC activation disabled), and no interrupt request is generated.**

**Note** Products of groups A, B, C, and D: i = 0 to 4
Products of group E: i = 0 to 5

### 19.3.5 High-Speed Transfer Operation

There are two channels for high-speed transfer. Each DTC activation source is selected by the high-speed DTC channel select register m. When the DTC is activated by the source selected by the high-speed transfer channel, the control data dedicated for high-speed transfer instead of the control data specified by the DTC vector address is read and transferred.

Initialize the control data area with any desired value because the control data are read.

Block transfer always transfers 1-byte data for 8-bit transfer and 2-byte data for 16-bit transfer.

Chain transfer reads and transfers the control data consecutively allocated subsequent to the control data specified by the DTC vector address. During a chain transfer, when control data is for the source selected by the other high-speed transfer channel, the consecutively allocated control data instead of the control data dedicated for high-speed transfer is read and transferred.

Table 19-9 shows the register functions in high-speed transfer operation.

**Table 19-9. Register Functions in High-speed Transfer Mode**

| Register Name | Symbol | Function |
|---|---|---|
| High-speed DTC channel select register 0/1 | SELHS0/1 | Channel selection |
| High-speed DTC control register 0/1 | HDTCCR0/1 | Operating mode control |
| High-speed DTC transfer count register 0/1 | HDTCCT0/1 | Number of data transfers |
| High-speed DTC transfer count reload register 0/1 | HDTRLD0/1 | Initial value setting |
| High-speed DTC source address register 0/1 | HDTSAR0/1 | Data transfer source address |
| High-speed DTC destination address register 0/1 | HDTDAR0/1 | Data transfer destination address |

## 19.4 Notes on DTC

### 19.4.1 Setting DTC Registers and Vector Table

- Do not access the DTC SFRs, the DTC control data area, the DTC vector table area, or the general-register (FFEE0H to FFEFFH) space using a DTC transfer.
- Modify the DTC base address register (DTCBAR) while all DTC activation sources are set to activation disabled.
- Do not rewrite the DTC base address register (DTCBAR) twice or more.
- Modify the data of the DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj register when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 5[Note]) register is 0 (DTC activation disabled).
- Modify the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 5[Note]) register is 0 (DTC activation disabled).

**Note** Products of groups A, B, C, and D: i = 0 to 4
Products of group E: i = 0 to 5

### 19.4.2 Allocation of DTC Control Data Area and DTC Vector Table Area

The areas where the DTC control data and vector table can be allocated differ, depending on the usage conditions.

- It is prohibited to use the general-purpose register (FFF00H to FFEE0H) space as the DTC control data area or DTC vector table area.
- The 18-byte area between the DTC vector table area and the DTC control data area is a reserved area to be used when the number of DTC activation sources is expanded.

### 19.4.3 DTC Pending Instruction

If a transfer request is generated from the DTC to the CPU, the DTC is not activated immediately after the following instructions. Also, the DTC is not activated between PREFIX instruction code and the instruction immediately after that code.

- Call/return instruction
- Unconditional branch instruction
- Conditional branch instruction
- Read access instruction for code flash memory
- Bit manipulation instructions for IFxx, MKxx, PRxx, and PSW, and an 8-bit manipulation instruction that has the ES register as operand
- Instruction for accessing the data flash memory
- <R> Multiply, Divide, Multiply & accumulate instruction (exclude MULU instruction)

**Cautions 1.** On reception of a DTC transfer request, all interrupt requests are held pending until the DTC transfer is completed.

**2.** All interrupt requests are also held pending while a DTC transfer is suspended due to a DTC pending instruction.

### 19.4.4 Operations when an Instruction which Accesses an SFR Register that Requires a Wait is Executed

DTC transfer is suspended while an instruction which accesses an SFR register[Note] that requires a wait is executed. The DTC transfer remains suspended as long as polling of the SFR register that requires a wait continues.

**Note** SFR registers that require a wait are registers of the CAM and LIN modules, and the TRJ0 register of the timer RJ module.

### 19.4.5 Operation when Accessing Data Flash Memory Space

Because DTC data transfer is suspended to access the data flash space, be sure to add the DTC pending instruction. If the data flash space is accessed after an instruction execution from start of DTC data transfer, a 3-clock wait will be inserted to the next instruction.

```
Instruction 1
DTC data transfer
Instruction 2 ← The wait of three clock cycles occurs.
MOV A,        ! Data Flash space
```

### 19.4.6 Number of DTC Execution Clock Cycles

Table 19-10 lists the operations following DTC activation and required number of clock cycles for each operation.

**Table 19-10. Operations Following DTC Activation and Required Number of Cycles**

| Vector Read | Control Data | | Data Read | Data Write |
|---|---|---|---|---|
| | Read | Write-back | | |
| 1 | 4 | **Note 1** | **Note 2** | **Note 2** |

**Notes 1.** For the number of clock cycles required for control data write-back, refer to **Table 19-11 Number of Clock Cycles Required for Control Data Write-Back Operation**.

**2.** For the number of clock cycles required for data read/write, refer to **Table 19-12 Number of Clock Cycles Required for Data Read/Write Operation**.

**Table 19-11. Number of Clock Cycles Required for Control Data Write-Back Operation**

| DTCCR Register Setting | | | | Address Setting | | Control Register to be Written Back | | | | Number of Clock Cycles |
|---|---|---|---|---|---|---|---|---|---|---|
| DAMOD | SAMOD | RPTSEL | MODE | Source | Destination | DTCCTj Register | DTRLDj Register | DTSARj Register | DTDARj Register | |
| 0 | 0 | X | 0 | Fixed | Fixed | Written back | Written back | Not written back | Not written back | 1 |
| 0 | 1 | X | 0 | Incremented | Fixed | Written back | Written back | Written back | Not written back | 2 |
| 1 | 0 | X | 0 | Fixed | Incremented | Written back | Written back | Not written back | Written back | 2 |
| 1 | 1 | X | 0 | Incremented | Incremented | Written back | Written back | Written back | Written back | 3 |
| 0 | X | 1 | 1 | Repeat area | Fixed | Written back | Written back | Written back | Not written back | 2 |
| 1 | X | 1 | 1 | | Incremented | Written back | Written back | Written back | Written back | 3 |
| X | 0 | 0 | 1 | Fixed | Repeat area | Written back | Written back | Not written back | Written back | 2 |
| X | 1 | 0 | 1 | Incremented | | Written back | Written back | Written back | Written back | 3 |

**Remark** j = 0 to 23; X: 0 or 1

**Table 19-12. Number of Clock Cycles Required for Data Read/Write Operation**

| Operation | RAM | Code Flash Memory | Data Flash Memory | SFR | 2nd SFR | |
|---|---|---|---|---|---|---|
| | | | | | No Wait State | Wait States |
| Data read | 1 | 2 | 4 | 1 | 1 | 1 + number of wait states Note |
| Data write | 1 | - | - | 1 | 1 | 1 + number of wait states Note |

**Note** The number of wait states differs depending on the specifications of the register allocated to the second SFR to be accessed.

### 19.4.7 Number of High-speed DTC Execution Clock Cycles

Table 19-13 lists the operations following high-speed DTC activation and required number of clock cycles for each operation.

**Table 19-13. Operations Following High-speed DTC Activation and Required Number of Cycles**

| Vector Read | Control Data | | Data Read | Data Write |
|---|---|---|---|---|
| | Read | Write-back | | |
| 1 | | 1 Note 1 | **Note 2** | **Note 2** |

**Notes 1.** For the number of clock cycles required for control data write-back, refer to **Table 19-14 Number of Clock Cycles Required for Control Data Write-Back Operation**.

**2.** For the number of clock cycles required for data read/write, refer to **Table 19-15 Number of Clock Cycles Required for Data Read/Write Operation**.

**Table 19-14. Number of Clock Cycles Required for Control Data Write-Back Operation**

| HDTCCRm Register Setting | | | | Address Setting | | Control Register to be Written Back | | | | Number of Clock Cycles |
|---|---|---|---|---|---|---|---|---|---|---|
| HDAMODm | HSAMODm | HRPTSELm | HMODEm | Source | Destination | HDTCCTm Register | HDTRLDm Register | HDTSARm Register | HDTDARm Register | |
| 0 | 0 | X | 0 | Fixed | Fixed | Written back | Not written back | Not written back | Not written back | 1 |
| 0 | 1 | X | 0 | Incremented | Fixed | Written back | Not written back | Written back | Not written back | 1 |
| 1 | 0 | X | 0 | Fixed | Incremented | Written back | Not written back | Not written back | Written back | 1 |
| 1 | 1 | X | 0 | Incremented | Incremented | Written back | Not written back | Written back | Written back | 1 |
| 0 | X | 1 | 1 | Repeat area | Fixed | Written back | Not written back | Written back | Not written back | 1 |
| 1 | X | 1 | 1 | | Incremented | Written back | Not written back | Written back | Written back | 1 |
| X | 0 | 0 | 1 | Fixed | Repeat area | Written back | Not written back | Not written back | Written back | 1 |
| X | 1 | 0 | 1 | Incremented | | Written back | Not written back | Written back | Written back | 1 |

**Remark** m = 0, 1; X: 0 or 1

**Table 19-15. Number of Clock Cycles Required for Data Read/Write Operation**

| Operation | RAM | Code Flash Memory | Data Flash Memory | SFR | 2nd SFR | |
|---|---|---|---|---|---|---|
| | | | | | No Wait State | Wait States |
| Data read | - | - | - | 1 | 1 | 1 + number of wait states Note |
| Data write | 1 | - | - | 1 | 1 | 1 + number of wait states Note |

**Note** The number of wait states differs depending on the specifications of the register allocated to the second SFR to be accessed.

### 19.4.8 DTC Response Time

Table 19-16 lists the DTC response time. The DTC response time is the time from when the DTC activation source is detected until DTC transfer starts, excluding the number of DTC execution clocks.

The DTC response time in high-speed transfer is the same as that in the normal transfer.

**Table 19-16. DTC Response Time**

| | Minimum Time | Maximum Time |
|---|---|---|
| Response Time | 3 clocks | 19 clocks |

Note that the response from the DTC may be further delayed under the following cases. The number of delayed clock cycles differs depending on the conditions.

- When executing an instruction from the internal RAM
  Maximum response time: 20 clocks
- When executing a DTC pending instruction (refer to **19.4.3 DTC Pending Instruction**)
  Maximum response time: Maximum response time for each condition + execution clock cycles for the instruction to be held pending under the condition.
- When accessing the TRJ0 register that a wait occurs
  Maximum response time: Maximum response time for each condition + 1 clock

**Remark** 1 clock: $1/f_{CLK}$ ($f_{CLK}$: CPU/peripheral hardware clock)

### 19.4.9 DTC Activation Sources

- After inputting a DTC activation source, do not input the same activation source again until DTC transfer is completed.
- While a DTC activation source is generated, do not manipulate the DTC activation enable bit corresponding to the source.
- If DTC activation sources conflict, their priority levels are determined in order to select the source for activation when the CPU acknowledges the DTC transfer. For details on the priority levels of activation sources, refer to **19.2.3 DTC Vector Table**.
- When DTC activation is enabled under either of the following conditions, a DTC transfer is started and an interrupt is generated after completion of the transfer. Therefore, enable DTC activation after confirming the comparator output monitor flag (CMPMON0) as necessary.
  - The comparator is set to releasing STOP mode by comparator interrupt enabled (CSTEN = 1), comparator output not inverted (CINV = 0), and comparator input > reference voltage
  - The comparator is set to releasing STOP mode by comparator interrupt enabled (CSTEN = 1), comparator output inverted (CINV = 1), and comparator input < reference voltage

### 19.4.10 Operation in Standby Mode Status

| Status | DTC Operation |
|---|---|
| HALT mode | Operable (Operation is disabled while in the low power consumption RTC mode) |
| STOP mode | DTC activation sources can be accepted Note 2 |
| SNOOZE mode | Operable Notes 1, 3 |

**Notes 1.** The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected as $f_{CLK}$.

**2.** In the STOP mode, detecting a DTC activation source enables transition to SNOOZE mode and DTC transfer. After completion of transfer, the system returns to the STOP mode. However, since the code flash memory and the data flash memory are stopped during the SNOOZE mode, the flash memory cannot be set as the transfer source.

**3.** When an A/D conversion end interrupt is set as a DTC activation source from the A/D converter SNOOZE mode function, release the SNOOZE mode using the A/D conversion end interrupt to start CPU processing after completion of DTC transfer, or use a chained transfer to set the A/D converter SNOOZE mode function again after clear the AWC bit.

### 19.4.11 Notes When the RAM Area Is the Source of the Data for Transfer

Make initial settings with the desired values when the RAM area is the source of the data for transfer. RAM ECC interrupts may be produced in some cases.

### 19.4.12 Vector Address for High-Speed Transfer

For high-speed transfer, too, a DTC vector address allocated to each activation source is read out. For chain transfer at high-speed, control data indicated by the DTC vector address are allocated to contiguous areas, and the next control data are read out. For transfer other than chain transfer at high-speed, make initial settings with the desired values in the DTC vector addresses.

# CHAPTER 20 EVENT LINK CONTROLLER (ELC) (RL78/F14 Only)

The event link controller (ELC) mutually connects (links) events output from each peripheral function. By linking events, it becomes possible to coordinate operation between peripheral functions directly without going through the CPU.

## 20.1 Overview

The ELC has the following functions.

- Capable of directly linking event signals from 20 (Group D products) or 26 (Group E products) types of peripheral functions to specified peripheral functions
- Event signals can be used as activation sources for operating any one of seven (Group D products) or nine (Group E products) types of peripheral functions

**Remark** Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory

Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or RL78/F13 (LIN incorporated) products with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory

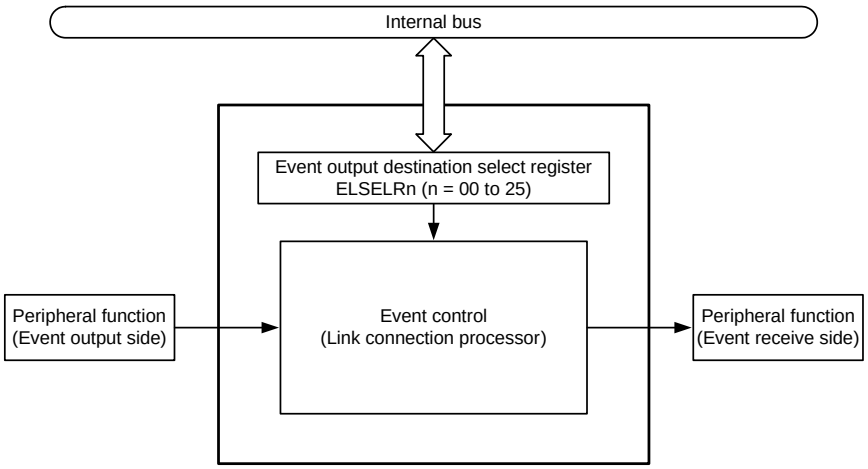
Group C: RL78/F13 (CAN incorporated) products with 30, 32, 48, 64, or 80 pins and 32 Kbytes to 128 Kbytes of code flash memory

Group D: RL78/F14 products with 30, 32, 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory

Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or RL78/F14 products with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

Figure 20-1 shows the Event Link Controller Block Diagram.

**Figure 20-1. Event Link Controller Block Diagram**



## 20.2 Registers

Table 20-1 lists the ELC register configuration.

**Table 20-1. ELC Register Configuration**

| Register name | Symbol | After Reset | Address | Access size |
|---|---|---|---|---|
| Event Output Destination Select Register 00 | ELSELR00 | 00H | F0780H | 1, 8 |
| Event Output Destination Select Register 01 | ELSELR01 | 00H | F0781H | 1, 8 |
| Event Output Destination Select Register 02 | ELSELR02 | 00H | F0782H | 1, 8 |
| Event Output Destination Select Register 03 | ELSELR03 | 00H | F0783H | 1, 8 |
| Event Output Destination Select Register 04 | ELSELR04 | 00H | F0784H | 1, 8 |
| Event Output Destination Select Register 05 | ELSELR05 | 00H | F0785H | 1, 8 |
| Event Output Destination Select Register 06 | ELSELR06 | 00H | F0786H | 1, 8 |
| Event Output Destination Select Register 07 | ELSELR07 | 00H | F0787H | 1, 8 |
| Event Output Destination Select Register 08 | ELSELR08 | 00H | F0788H | 1, 8 |
| Event Output Destination Select Register 09 | ELSELR09 | 00H | F0789H | 1, 8 |
| Event Output Destination Select Register 10 | ELSELR10 | 00H | F078AH | 1, 8 |
| Event Output Destination Select Register 11 | ELSELR11 | 00H | F078BH | 1, 8 |
| Event Output Destination Select Register 12 | ELSELR12 | 00H | F078CH | 1, 8 |
| Event Output Destination Select Register 13 | ELSELR13 | 00H | F078DH | 1, 8 |
| Event Output Destination Select Register 14 | ELSELR14 | 00H | F078EH | 1, 8 |
| Event Output Destination Select Register 15 | ELSELR15 | 00H | F078FH | 1, 8 |
| Event Output Destination Select Register 16 | ELSELR16 | 00H | F0790H | 1, 8 |
| Event Output Destination Select Register 17 | ELSELR17 | 00H | F0791H | 1, 8 |
| Event Output Destination Select Register 18 | ELSELR18 | 00H | F0792H | 1, 8 |
| Event Output Destination Select Register 19 | ELSELR19 | 00H | F0793H | 1, 8 |
| Event Output Destination Select Register 20 Note | ELSELR20 | 00H | F0794H | 1, 8 |
| Event Output Destination Select Register 21 Note | ELSELR21 | 00H | F0795H | 1, 8 |
| Event Output Destination Select Register 22 Note | ELSELR22 | 00H | F0796H | 1, 8 |
| Event Output Destination Select Register 23 Note | ELSELR23 | 00H | F0797H | 1, 8 |
| Event Output Destination Select Register 24 Note | ELSELR24 | 00H | F0798H | 1, 8 |
| Event Output Destination Select Register 25 Note | ELSELR25 | 00H | F0799H | 1, 8 |

**Note** For Group E products only.

### 20.2.1 Event Output Destination Select Register n (ELSELRn) (n = 00 to 25)

An ELSELRn register links each event signal to an operation of an event-receiving peripheral function (link destination peripheral function) after reception.

Do not set multiple event inputs to the same event output destination (event receive side). The operation of the event-receiving peripheral function will become undefined, and event signals may not be received correctly. In addition, do not set the event link generation source and the event link output destination to the same function.

Set an ELSELRn register during a period when no event output peripheral functions are generating event signals.

Table 20-2 lists the correspondence between ELSELRn (n = 00 to 25) registers and peripheral functions, and Table 20-3 lists the correspondence between values set to ELSELRn (n = 00 to 25) registers and operation of link destination peripheral functions at reception.

**Figure 20-2. Format of Event Output Destination Select Register n (ELSELRn) (n = 00 to 25)**

Address: F0780H (ELSELR00) to F0799H (ELSELR25) After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ELSELRn | - | - | - | - | ELSELRn3 Note 1 | ELSELRn2 | ELSELRn1 | ELSELRn0 |

| Bits 7 to 4 | Reserved |
|---|---|
| - | The read value is 0. |

| ELSELRn3 Note 1 | ELSELRn2 | ELSELRn1 | ELSELRn0 | Event Link Selection |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Event link disabled |
| 0 | 0 | 0 | 1 | Select operation of peripheral function to link Note 2 |
| 0 | 0 | 1 | 0 | Select operation of peripheral function to link Note 2 |
| 0 | 0 | 1 | 1 | Select operation of peripheral function to link Note 2 |
| 0 | 1 | 0 | 0 | Select operation of peripheral function to link Note 2 |
| 0 | 1 | 0 | 1 | Select operation of peripheral function to link Note 2 |
| 0 | 1 | 1 | 0 | Select operation of peripheral function to link Note 2 |
| 0 | 1 | 1 | 1 | Select operation of peripheral function to link Note 2 |
| 1 | 0 | 0 | 0 | Select operation of peripheral function to link Note 2 |
| 1 | 0 | 0 | 1 | Select operation of peripheral function to link Note 2 |

**Notes 1.** Only for Group E products. Set this bit to 0 in Group D products.

**2.** See **Table 20-3 Correspondence Between Values Set to ELSELRn (n = 00 to 25) Registers and Operation of Link Destination Peripheral Functions at Reception**.

**Table 20-2. Correspondence Between ELSELRn (n = 00 to 25) Registers and Peripheral Functions**

| Register Name | Event Generator (Output Origin of Event Input n) | Event Description |
|---|---|---|
| ELSELR00 | External interrupt edge detection 0 | INTP0 |
| ELSELR01 | External interrupt edge detection 1 | INTP1 |
| ELSELR02 | External interrupt edge detection 2 | INTP2 |
| ELSELR03 | External interrupt edge detection 3 | INTP3 |
| ELSELR04 | External interrupt edge detection 4 | INTP4 |
| ELSELR05 | External interrupt edge detection 5 | INTP5 |
| ELSELR06 | Key return signal detection | INTKR |
| ELSELR07 | RTC fixed-cycle signal/Alarm match detection | INTRTC |
| ELSELR08 | Timer RD0 Input capture A/Compare match A | INTTRD0 |
| ELSELR09 | Timer RD0 Input capture B/Compare match B | INTTRD0 |
| ELSELR10 | Timer RD1 Input capture A/Compare match A | INTTRD1 |
| ELSELR11 | Timer RD1 Input capture B/Compare match B | INTTRD1 |
| ELSELR12 | Timer RD1 Underflow | TRD1 underflow signal |
| ELSELR13 | Timer RJ0 | INTTRJ0 |
| ELSELR14 | TAU0 channel 0 Count end/Capture end | INTTM00 |
| ELSELR15 | TAU0 channel 1 Count end/Capture end | INTTM01 |
| ELSELR16 | TAU0 channel 2 Count end/Capture end | INTTM02 |
| ELSELR17 | TAU0 channel 3 Count end/Capture end | INTTM03 |
| ELSELR18 | TAU0 channel 4 Count end/Capture end | INTTM04 |
| ELSELR19 | Comparator detection 0 | INTCMP0 |
| ELSELR20 Note | TAU0 channel 5 Count end/Capture end | INTTM05 |
| ELSELR21 Note | TAU0 channel 6 Count end/Capture end | INTTM06 |
| ELSELR22 Note | TAU0 channel 7 Count end/Capture end | INTTM07 |
| ELSELR23 Note | TAU1 channel 0 Count end/Capture end | INTTM10 |
| ELSELR24 Note | TAU1 channel 1 Count end/Capture end | INTTM11 |
| ELSELR25 Note | TAU1 channel 2 Count end/Capture end | INTTM12 |

**Note** For Group E products only.

**Table 20-3. Correspondence Between Values Set to ELSELRn (n = 00 to 25) Registers and Operation of Link Destination Peripheral Functions at Reception**

| Bits ELSELRn3 to ELSELRn0 in ELSELRn Register | Link Destination Peripheral Function | Operation When Receiving Event |
|---|---|---|
| 0001B | A/D converter | A/D conversion starts |
| 0010B | Timer input of timer array unit 0 channel 0 Notes 1 and 2 | Delay counter, input pulse interval measurement, external event counter |
| 0011B | Timer input of timer array unit 0 channel 1 Notes 1 and 2 | |
| 0100B | Timer RJ0 | Count source |
| 0101B | Timer RD0 | TRDIOD0 input capture, pulse output cutoff |
| 0110B | Timer RD1 | TRDIOD1 input capture, pulse output cutoff |
| 0111B | DA0 Note 3 | Real-time output |
| 1000B | Timer input of timer array unit 0 channel 2 Notes 1 and 2 | Delay counter, input pulse interval measurement, external event counter (Group E products only) |
| 1001B | Timer input of timer array unit 0 channel 3 Notes 1 and 2 | |

**Notes 1.** To select the timer input of timer array unit 0 channel m as the link destination peripheral function, first set the operating clock for channel m to $f_{CLK}$ using timer clock select register 0 (TPS0), and then set the timer output used for channel m to an event input signal from the ELC using timer input select register 0 (TIS0).

**2.** Before selecting the timer input of timer array unit 0 channel m as the link destination peripheral function, set the noise filter of the corresponding link destination channel in the timer array unit 0 to OFF (set the TNFEN0m bit to 0) by using the noise filter enable register 1 (NFEN1).

**3.** When entering the STOP status while the real-time output event mode for D/A conversion is enabled, disable linking of ELC events before entering STOP.

**Remark** m = 0, 3

### 20.2.2 Timer input select register 0

The timer array unit channels 0 to 3 change the event input from the ELC to the source for each channel. For details, see **6.3.8 Timer input select register 0 (TIS0)**.

### 20.2.3 A/D converter mode register 1 (ADM1)

This register has the function of specifying the A/D conversion trigger. A/D conversion is performed by two-bit control and using the signal by the TAU0 channel 1, event signal by the ELC, or pretimed signal/alarm interrupt signal as triggers. For details, see **12.3.3 A/D converter mode register 1 (ADM1)**.

### 20.2.4 D/A converter mode register (DAM)

This register has the real-time output function, which starts D/A conversion using the event input from the ELC as triggers. For details, see **13.3.3 D/A converter mode register (DAM)**.

## 20.3 Operation

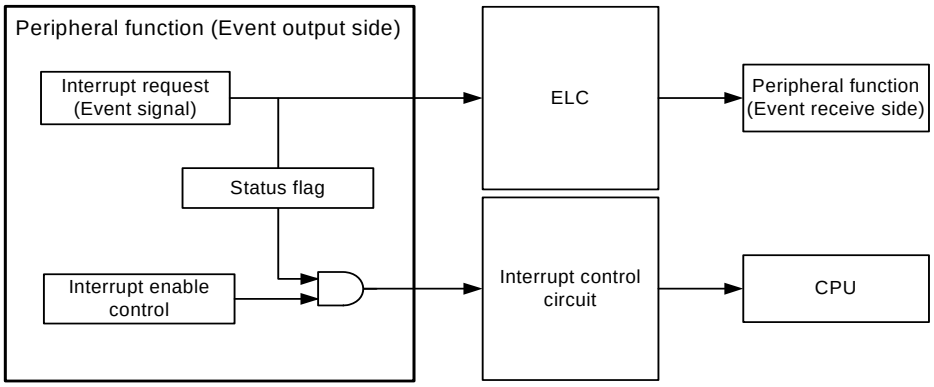
The path for using an event signal generated by a peripheral function as an interrupt request to the interrupt control circuit is independent from the path for using it as an ELC event. Therefore, each event signal can be used as an event signal for operation of an event-receiving peripheral function, regardless of interrupt control.

In addition, event link operation can be performed without being influenced by the presence or absence of a CPU clock supply. However, the operating clock of a peripheral function needs to be supplied and be in an operational state.

Figure 20-3 shows the relationship between interrupt handling and ELC. The figure show an example of an interrupt request status flag and a peripheral function possessing the enable bits that control enabling/disabling of such interrupts.

A peripheral function which receives an event from the ELC will perform the operation corresponding to the event-receiving peripheral function after reception of an event (see **Table 20-3 Correspondence Between Values Set to ELSELRn (n = 00 to 25) Registers and Operation of Link Destination Peripheral Functions at Reception**).

**Figure 20-3. Relationship Between Interrupt Handling and ELC**



# CHAPTER 21 INTERRUPT FUNCTIONS

If different processing is required during the execution of one program, interrupts provide a convenient and fast way of switching to another program to handle that processing.

After processing at the branch destination is completed, execution returns to the point where the original program was suspended.

The number of interrupt sources differs, depending on the product.

| | | RL78/F13 (LIN) | | | | | RL78/F13 (CAN and LIN) | | | | RL78/F14 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Group A | | | Group B | | Group C | | | | Group D | | | | Group E | | |
| | | 20 pins | 30/32 pins | 48/64 pins | 48 pins | 64/80 pins | 30 pins | 32 pins | 48 pins | 64/80 pins | 30 pins | 32 pins | 48 pins | 64/80 pins | 48 pins | 64 pins | 80/100 pins |
| Maskable interrupts | External | 7 | 8 | 10 | 12 | 13 | 9 | 9 | 13 | 14 | 9 | 9 | 13 | 14 | 14 | 15 | 16 |
| | Internal | 26 | 26 | 26 | 35 | 35 | 39 | 40 | 40 | 40 | 40 | 41 | 41 | 41 | 48 | 48 | 48 |

**Remark** Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory

Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory

Group C: RL78/F13 (CAN incorporated) products with 30, 32, 48, 64, or 80 pins and 32 Kbytes to 128 Kbytes of code flash memory

Group D: RL78/F14 products with 30, 32, 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory

Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

## 21.1 Interrupt Function Types

The following two types of interrupt functions are used.

### (1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the default priority of vectored interrupt servicing. For the default priority, see **Table 21-1**.

A standby release signal is generated and STOP, HALT, and SNOOZE modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

### (2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

## 21.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Table 21-1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

**Table 21-1. Interrupt Source List (1/4)**

| Interrupt Type | Default Priority Note 1 | Interrupt Source | | Internal/ External | Vector Table Address | Basic Configuration Type Note 2 | 80, 100-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Name | Trigger | | | | | | | | | |
| Maskable | 0 | INTWDTI | Watchdog timer interval Note 3 (75% of overflow time+1/2$f_{WDT}$) | Internal | 0004H | (A) | √ | √ | √ | √ | √ | √ |
| | 1 | INTLVI | Voltage detection Note 4 | | 0006H | | √ | √ | √ | √ | √ | √ |
| | 2 | INTP0 | Pin input edge detection 0 | External | 0008H | (B) | √ | √ | √ | √ | √ | √ |
| | 3 | INTP1 | Pin input edge detection 1 | | 000AH | | √ | √ | √ | √ | √ | √ |
| | 4 | INTP2 | Pin input edge detection 2 | | 000CH | | √ | √ | √ | √ | √ | √ |
| | 5 | INTP3 | Pin input edge detection 3 | | 000EH | | √ | √ | √ | √ | √ | √ |
| | 6 | INTP4 Note 7 | Pin input edge detection 4 | | 0010H | | √ | √ | √ | √ | √ | √ |
| | | INTSPM Note 7 | Stack pointer overflow/underflow | Internal | | (A) | | | | | | |
| | 7 | INTP5 Note 8 | Pin input edge detection 5 | External | 0012H | (B) | √ | √ | √ | √ | √ | – |
| | | INTCMP0 Note 8 | Comparator detection 0 | Internal | | (A) | √ Note 5 | √ Note 5 | √ Note 5 | √ Note 5 | √ Note 5 | – |
| | 8 | INTP13 Note 9 | Pin input edge detection 13 | External | 0014H | (B) | √ Note 6 | – | – | – | – | – |
| | | INTCLM Note 9 | Main clock or PLL clock stop | Internal | | (A) | √ | √ | √ | √ | √ | √ |
| | 9 | INTST0/ INTCSI00/ INTIIC00 | UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end | | 0016H | | √ | √ | √ | √ | √ | √ |
| | 10 | INTSR0/ INTCSI01/ INTIIC01 | UART0 reception transfer end/CSI01 transfer end or buffer empty interrupt/IIC01 transfer end | | 0018H | | √ | √ | √ | √ | √ | √ |
| | 11 | INTTRD0 | Timer RD0 input capture, compare match, overflow, underflow interrupt | | 001AH | | √ | √ | √ | √ | √ | √ |
| | 12 | INTTRD1 | Timer RD1 input capture, compare match, overflow, underflow interrupt | | 001CH | | √ | √ | √ | √ | √ | √ |
| | 13 | INTTRJ0 | Timer RJ0 | | 001EH | | √ | √ | √ | √ | √ | √ |
| | 14 | INTRAM | RAM 1-bit correction/2-bit error detection | | 0020H | | √ | √ | √ | √ | √ | √ |
| | 15 | INTLIN0TRM | LIN0 transmission | | 0022H | | √ | √ | √ | √ | √ | √ |
| | 16 | INTLIN0RVC | LIN0 reception end | | 0024H | | √ | √ | √ | √ | √ | √ |

**Notes 1.** The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 55 indicates the lowest priority.

**2.** Basic configuration types (A) to (F) correspond to (A) to (F) in Figure 21-1.

**3.** When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

**4.** When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

**5.** Provided only in products of Groups D and E.

**6.** Provided only in Group E products.

**7.** To determine whether the actual interrupt source is INTP4 or INTSPM, read the INTFLG00 bit in the INTFLG0 register or the stack pointer.

**8.** To determine whether the actual interrupt source is INTP5 or INTCMP0, read the INTFLG01 and INTFLG06 bits in the INTFLG0 register.

**9.** To determine whether the actual interrupt source is INTP13 or INTCLM, read the INTFLG07 bit in the INTFLG0 register and SELPLLS bit in the PLLSTS register.

**Table 21-1. Interrupt Source List (2/4)**

| Interrupt Type | Default Priority Note 1 | Interrupt Source | | Internal/ External | Vector Table Address | Basic Configuration Type Note 2 | 80, 100-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Name | Trigger | | | | | | | | | |
| Maskable | 17 | INTSTLIN0 STA/ INTLIN0 | LIN0 reception status/LIN0 interrupt | Internal | 0026H | (A) | √ | √ | √ | √ | √ | √ |
| | 18 | INTIICA0 | IICA0 transfer end | | 0028H | | √ | √ Note 3 | √ Note 3 | √ Note 3 | – | – |
| | 19 | INTP8 Note 6 | Pin input edge detection 8 | External | 002AH | (B) | √ | √ Note 3 | √ Note 3 | – | – | – |
| | | INTRTC Note 6 | RTC pretimed signal or alarm match detection | Internal | | (A) | √ | √ | √ | √ | √ | √ |
| | 20 | INTTM00 | End of timer channel 0 count/capture | | 002CH | | √ | √ | √ | √ | √ | √ |
| | 21 | INTTM01 | End of timer channel 1 count/capture | | 002EH | | √ | √ | √ | √ | √ | √ |
| | 22 | INTTM02 | End of timer channel 2 count/capture | | 0030H | | √ | √ | √ | √ | √ | √ |
| | 23 | INTTM03 | End of timer channel 3 count/capture | | 0032H | | √ | √ | √ | √ | √ | √ |
| | 24 | INTAD | End of A/D conversion | | 0034H | | √ | √ | √ | √ | √ | √ |
| | 25 | INTP6 Note 4 | Pin input edge detection 6 | External | 0036H | (B) | √ | √ | √ | – | – | – |
| | | INTTM11H | Upper 8-bit interval timer interrupt of TAU1 channel 1 (when 8-bit timer function is selected) | Internal | | (A) | √ | √ Note 3 | √ Note 3 | √ Note 3 | √ Note 3 | – |
| | 26 | INTP7 Note 4 | Pin input edge detection 7 | External | 0038H | (B) | √ | √ | √ | – | – | – |
| | | INTTM13H | Upper 8-bit interval timer interrupt of TAU1 channel 3 (when 8-bit timer function is selected) | Internal | | (A) | √ | √ Note 3 | √ Note 3 | √ Note 3 | √ Note 3 | – |
| | 27 | INTP9 Note 4 | Pin input edge detection 9 | External | 003AH | (B) | √ | √ Note 3 | √ Note 3 | – | – | – |
| | | INTTM01H | Upper 8-bit interval timer interrupt of TAU0 channel 1 (when 8-bit timer function is selected) | Internal | | (A) | √ | √ | √ | √ | √ | √ |
| | 28 | INTP10 Note 4 | Pin input edge detection 10 | External | 003CH | (B) | √ | √ Note 3 | – | – | – | – |
| | | INTTM03H | Upper 8-bit interval timer interrupt of TAU0 channel 3 (when 8-bit timer function is selected) | Internal | | (A) | √ | √ | √ | √ | √ | √ |
| | 29 | INTST1/ INTCSI10/ INTIIC10 | UART1 transmission transfer end or buffer empty interrupt/CSI10 transfer end or buffer empty interrupt/IIC10 transfer end | | 003EH | | √ | √ Note 3 | √ Note 3 | √ Notes 3, 5 | √ Notes 3, 5 | – |

**Notes 1.** The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 55 indicates the lowest priority.

**2.** Basic configuration types (A) to (F) correspond to (A) to (F) in Figure 21-1.

**3.** Not provided in Group A products.

**4.** Whether the interrupt source is the detection of edge input on a pin or a TAU count end/capture end interrupt is not detectable.

**5.** Only INTST1 is provided.

**6.** To determine whether the actual interrupt source is INTP8 or INTRTC, read the INTFLG02 bit in the INTFLG0 register or the WAFG and RIFG bits in the RTCC1 register.

**Table 21-1. Interrupt Source List (3/4)**

| Interrupt Type | Default Priority Note 1 | Interrupt Source | | Internal/ External | Vector Table Address | Basic Configuration Type Note 2 | 80, 100-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Name | Trigger | | | | | | | | | |
| Maskable | 30 | INTSR1/ INTCSI11/ INTIIC11 | UART1 reception transfer end/CSI11 transfer end or buffer empty interrupt/IIC11 transfer end | Internal | 0040H | (A) | √ | √ Note 3 | √ Note 3 | √ Notes 3, 7 | √ Notes 3, 7 | – |
| | 31 | INTTM04 | End of timer channel 4 count/capture | | 0042H | | √ | √ | √ | √ | √ | √ |
| | 32 | INTTM05 | End of timer channel 5 count/capture | | 0044H | | √ | √ | √ | √ | √ | √ |
| | 33 | INTTM06 | End of timer channel 6 count/capture | | 0046H | | √ | √ | √ | √ | √ | √ |
| | 34 | INTTM07 | End of timer channel 7 count/capture | | 0048H | | √ | √ | √ | √ | √ | √ |
| | 35 | INTP11 Note 6 | Pin input edge detection 11 | External | 004AH | (B) | √ | √ Note 3 | – | – | – | – |
| | | INTLIN0WUP Note 6 | LIN0 reception pin input | | | (E) | √ | √ | √ | √ | √ | √ |
| | 36 | INTKR | Key interrupt detection | | 004CH | (C) | √ | √ | √ | √ | √ | √ |
| | 37 | INTCAN0ERR | CAN0 channel error | Internal | 004EH | (A) | √ Note 4 | √ Note 4 | √ Note 4 | √ Note 4 | √ Note 4 | – |
| | 38 | INTCAN0WUP | CAN0 wakeup | External | 0050H | (D) | √ Note 4 | √ Note 4 | √ Note 4 | √ Note 4 | √ Note 4 | – |
| | 39 | INTCAN0CFR | CAN0 transmit/receive FIFO receive | Internal | 0052H | (A) | √ Note 4 | √ Note 4 | √ Note 4 | √ Note 4 | √ Note 4 | – |
| | 40 | INTCAN0TRM | CAN0 channel transmit | | 0054H | | √ Note 4 | √ Note 4 | √ Note 4 | √ Note 4 | √ Note 4 | – |
| | 41 | INTCANGRFR | CAN global receive FIFO | | 0056H | | √ Note 4 | √ Note 4 | √ Note 4 | √ Note 4 | √ Note 4 | – |
| | 42 | INTCANGERR | CAN global error | | 0058H | | √ Note 4 | √ Note 4 | √ Note 4 | √ Note 4 | √ Note 4 | – |
| | 43 | INTTM10 | End of TAU1 channel 0 count/capture | | 005AH | | √ | √ Note 3 | √ Note 3 | √ Note 3 | √ Note 3 | – |
| | 44 | INTTM11 | End of TAU1 channel 1 count/capture | | 005CH | | √ | √ Note 3 | √ Note 3 | √ Note 3 | √ Note 3 | – |
| | 45 | INTTM12 | End of TAU1 channel 2 count/capture | | 005EH | | √ | √ Note 3 | √ Note 3 | √ Note 3 | √ Note 3 | – |
| | 46 | INTTM13 | End of TAU1 channel 3 count/capture | | 0060H | | √ | √ Note 3 | √ Note 3 | √ Note 3 | √ Note 3 | – |
| | 47 | INTFL | Reserved Note 5 | | 0062H | | √ | √ | √ | √ | √ | √ |

**Notes 1.** The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 55 indicates the lowest priority.

**2.** Basic configuration types (A) to (F) correspond to (A) to (F) in Figure 21-1.

**3.** Not provided in Group A products.

**4.** Not provided in products of Groups A and B.

<R> **5.** Do not use this interrupt.

**6.** Select INTP11 and INTLIN0WUP by the ISC2 bit in the ISC register.

**7.** Only INTSR1 is provided.

**Table 21-1. Interrupt Source List (4/4)**

| Interrupt Type | Default Priority Note 1 | Interrupt Source | | Internal/ External | Vector Table Address | Basic Configuration Type Note 2 | 80, 100-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Name | Trigger | | | | | | | | | |
| | | | | | | | | | | | | |
| | | INTLIN1WUP Note 6 | LIN1 reception pin input | | | (E) | √ Note 5 | √ Note 5 | √ Note 5 | – | – | – |
| | 49 | INTLIN1TRM | LIN1 transmission | Internal | 0066H | (A) | √ Note 5 | √ Note 5 | √ Note 5 | – | – | – |
| | 50 | INTLIN1RVC | LIN1 reception end | | 0068H | | √ Note 5 | √ Note 5 | √ Note 5 | – | – | – |
| | 51 | INTLIN1STA/ INTLIN1 | LIN1 reception status/LIN1 interrupt | | 006AH | | √ Note 5 | √ Note 5 | √ Note 5 | – | – | – |
| | 52 | INTTM14 | End of TAU1 channel 4 count/capture | | 006CH | | √ Note 5 | √ Note 5 | √ Note 5 | – | – | – |
| | 53 | INTTM15 | End of TAU1 channel 5 count/capture | | 006EH | | √ Note 5 | √ Note 5 | √ Note 5 | – | – | – |
| | 54 | INTTM16 | End of TAU1 channel 6 count/capture | | 0070H | | √ Note 5 | √ Note 5 | √ Note 5 | – | – | – |
| | 55 | INTTM17 | End of TAU1 channel 7 count/capture | | 0072H | | √ Note 5 | √ Note 5 | √ Note 5 | – | – | – |
| Software | – | BRK | Execution of BRK instruction | – | 007EH | (F) | √ | √ | √ | √ | √ | √ |
| Reset | – | RESET | $\overline{\text{RESET}}$ pin input | – | 0000H | – | √ | √ | √ | √ | √ | √ |
| | | POR | Power-on-reset | | | | √ | √ | √ | √ | √ | √ |
| | | LVD | Voltage detection Note 3 | | | | √ | √ | √ | √ | √ | √ |
| | | WDT | Overflow of watchdog timer | | | | √ | √ | √ | √ | √ | √ |
| | | TRAP | Execution of illegal instruction Note 4 | | | | √ | √ | √ | √ | √ | √ |
| | | IAW | Illegal-memory access | | | | √ | √ | √ | √ | √ | √ |
| | | CLM | Main clock oscillation stop | | | | √ | √ | √ | √ | √ | √ |

**Notes 1.** The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 55 indicates the lowest priority.

**2.** Basic configuration types (A) to (F) correspond to (A) to (F) in Figure 21-1.

**3.** When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

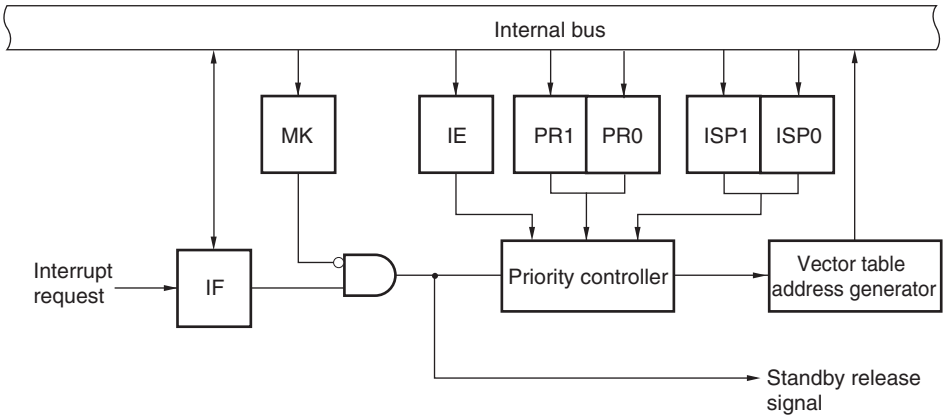
**4.** When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

**5.** Provided only in Group E products.

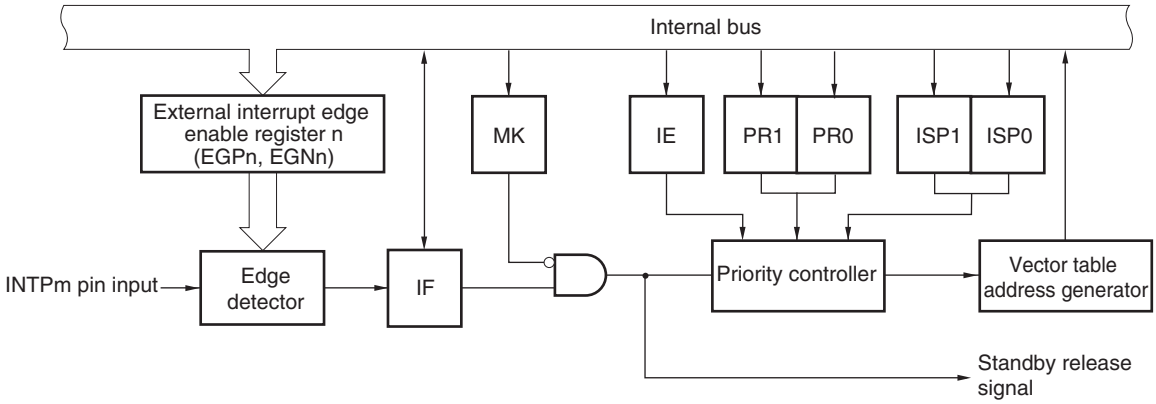
**6.** Select INTP12 and INTLIN1WUP by the ISC3 bit in the ISC register.

**Figure 21-1. Basic Configuration of Interrupt Function (1/3)**

**(A) Internal maskable interrupt**




**(B) External maskable interrupt (INTPm)**




IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

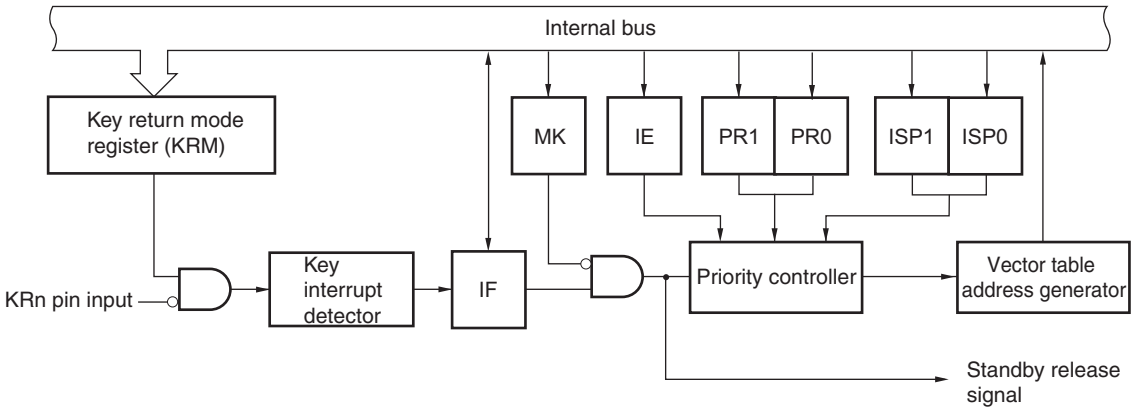
**Remark** n = 0, 1

| | |
|---|---|
| 20-pin: | m = 0 to 4 |
| 30-, 32-pin: | m = 0 to 5 |
| 48-pin: | m = 0 to 9 Note 1 |
| 64-pin: | m = 0 to 12 Notes 1, 2 |
| 80-, 100-pin: | m = 0 to 13 Note 2 |

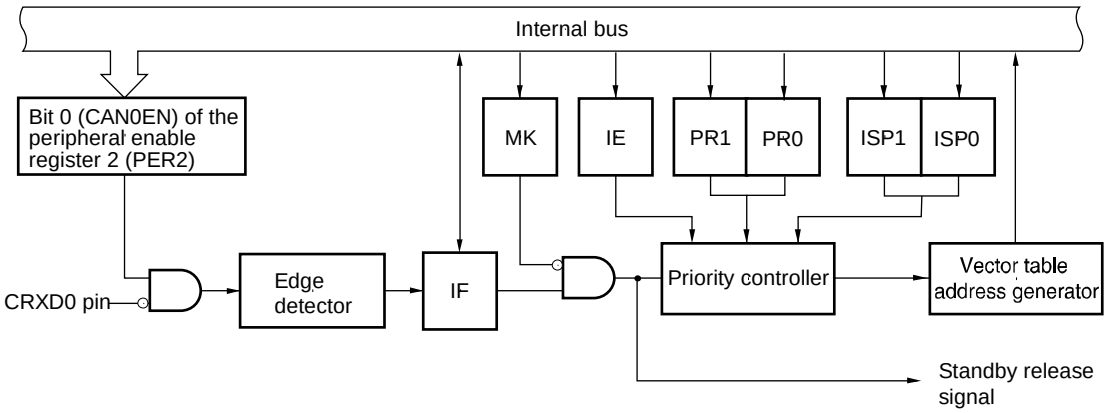
**Notes 1.** Group A products: m = 0 to 7
**2.** Products of Groups B to D: m = 0 to 11

**Figure 21-1. Basic Configuration of Interrupt Function (2/3)**

**(C) External maskable interrupt (INTKR)**




**(D) External maskable interrupt (CAN wake-up)**

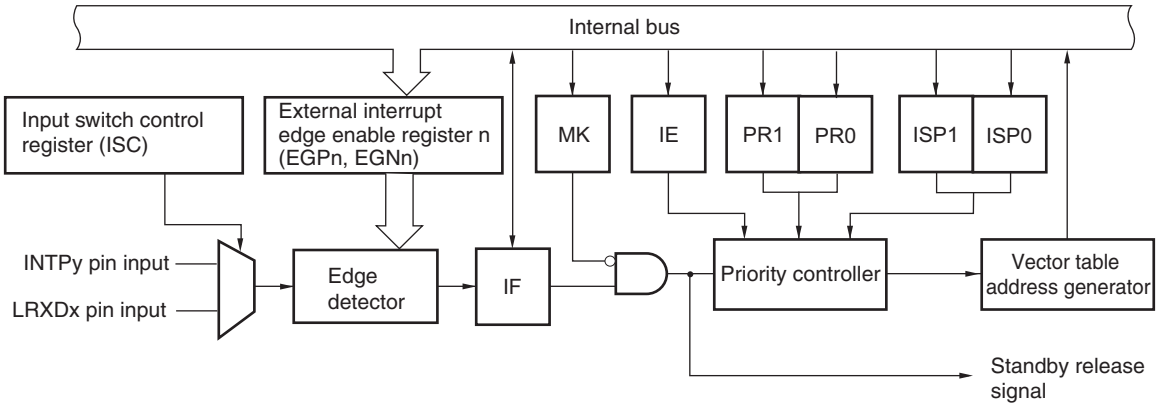



IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

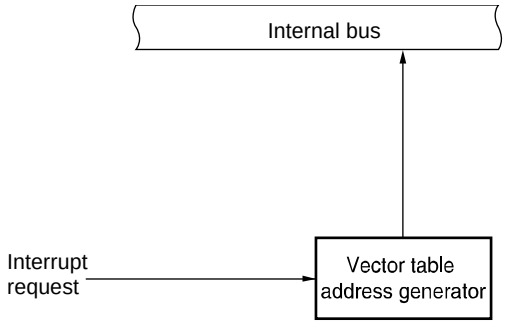
**Remark** 20-pin: n = 0, 1
30-pin: n = 0 to 7
32-pin: n = 0 to 5
48-, 64-, 80-, 100-pin: n = 0 to 7

**Figure 21-1. Basic Configuration of Interrupt Function (3/3)**

**(E) External maskable interrupt (LINx wake-up)**




**(F) Software interrupt**




IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

**Remark** n = 0, 1
x = 0, 1 **Note**
y = 11, 12 **Note**

**Note** Except in Group E products, x = 0 and y = 11.

## 21.3 Registers Controlling Interrupt Functions

The following 9 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L)
- External interrupt rising edge enable registers (EGP0, EGP1)
- External interrupt falling edge enable registers (EGN0, EGN1)
- Interrupt source determination flag register 0 (INTFLG0)
- Interrupt mask register (INTMSK)
- Input switch control register (ISC)
- Program status word (PSW)

Table 21-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

**Table 21-2. Flags Corresponding to Interrupt Request Sources (1/4)**

| Interrupt Source | Interrupt Request Flag | | Interrupt Mask Flag | | Priority Specification Flag | | 100-pin | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Register | | Register | | Register | | | | | | | |
| INTWDTI | WDTIIF | IF0L | WDTIMK | MK0L | WDTIPR0, WDTIPR1 | PR00L, PR10L | √ | √ | √ | √ | √ | √ | √ |
| INTLVI | LVIIF | | LVIMK | | LVIPR0, LVIPR1 | | √ | √ | √ | √ | √ | √ | √ |
| INTP0 | PIF0 | | PMK0 | | PPR00, PPR10 | | √ | √ | √ | √ | √ | √ | √ |
| INTP1 | PIF1 | | PMK1 | | PPR01, PPR11 | | √ | √ | √ | √ | √ | √ | √ |
| INTP2 | PIF2 | | PMK2 | | PPR02, PPR12 | | √ | √ | √ | √ | √ | √ | √ |
| INTP3 | PIF3 | | PMK3 | | PPR03, PPR13 | | √ | √ | √ | √ | √ | √ | √ |
| INTP4 | PIF4 | | PMK4 | | PPR04, PPR14 | | √ | √ | √ | √ | √ | √ | √ |
| INTSPM | SPMIF | | SPMMK | | SPMPR0, SPMPR1 | | √ | √ | √ | √ | √ | √ | √ |
| INTP5 | PIF5 | | PMK5 | | PPR05, PPR15 | | √ | √ | √ | √ | √ | √ | – |
| INTCMP0 | CMPIF0 | | CMPMK0 | | CMPPR00, CMPPR10 | | √ Note | √ Note | √ Note | √ Note | √ Note | √ Note | – |

**Note** Provided only in products of Groups D and E.

**Table 21-2. Flags Corresponding to Interrupt Request Sources (2/4)**

| Interrupt Source | Interrupt Request Flag | | Interrupt Mask Flag | | Priority Specification Flag | | 100-pin | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Register | | Register | | Register | | | | | | | |
| INTP13 | PIF13 | IF0H | PMK13 | MK0H | PPR013, PPR113 | PR00H, PR10H | √ Note | √ Note | – | – | – | – | – |
| INTCLM | CLMIF | | CLMMK | | CLMPR0, CLMPR1 | | √ | √ | √ | √ | √ | √ | √ |
| INTST0 | STIF0 | | STMK0 | | STPR00, STPR10 | | √ | √ | √ | √ | √ | √ | √ |
| INTCSI00 | IICIF00 | | CSIMK00 | | CSIPR000, CSIPR100 | | √ | √ | √ | √ | √ | √ | √ |
| INTIIC00 | IICIF00 | | IICMK00 | | IICPR000, IICPR100 | | √ | √ | √ | √ | √ | √ | √ |
| INTSR0 | SRIF0 | | SRMK0 | | SRPR00, SRPR10 | | √ | √ | √ | √ | √ | √ | √ |
| INTCSI01 | CSIIF01 | | CSIMK01 | | CSIPR001, CSIPR101 | | √ | √ | √ | √ | √ | √ | √ |
| INTIIC01 | IICIF01 | | IICMK01 | | IICPR001, IICPR101 | | √ | √ | √ | √ | √ | √ | √ |
| INTTRD0 | TRDIF0 | | TRDMK0 | | TRDPR00, TRDPR10 | | √ | √ | √ | √ | √ | √ | √ |
| INTTRD1 | TRDIF1 | | TRDMK1 | | TRDPR01, TRDPR11 | | √ | √ | √ | √ | √ | √ | √ |
| INTTRJ0 | TRJIF0 | | TRJMK0 | | TRJPR00, TRJPR10 | | √ | √ | √ | √ | √ | √ | √ |
| INTRAM | RAMIF | | RAMKK | | RMAMPR0, RAMPR1 | | √ | √ | √ | √ | √ | √ | √ |
| INTLIN0TRM | LIN0TRMIF | | LIN0TRMMK | | LIN0TRMPR0, LIN0TRMPR1 | | √ | √ | √ | √ | √ | √ | √ |

**Cautions 1. If one of the interrupt sources INTST0, INTCSI00, and INTIIC00 is generated, bit 1 of the IF0H register is set to 1. Bit 1 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.**

**2. If one of the interrupt sources INTSR0, INTCSI01, and INTIIC01 is generated, bit 2 of the IF0H register is set to 1. Bit 2 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.**

**Note** Provided only in Group E products.

**Table 21-2. Flags Corresponding to Interrupt Request Sources (3/4)**

| Interrupt Source | Interrupt Request Flag | Register | Interrupt Mask Flag | Register | Priority Specification Flag | Register | 100-pin | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INTLIN0RVC | LIN0RVCIF | IF1L | LIN0RVCMK | MK1L | LIN0RVCPR0, LIN0RVCPR1 | PR01L, PR11L | √ | √ | √ | √ | √ | √ | √ |
| INTLIN0STA | LIN0STAIF | | LIN0STAMK | | LIN0STAPR0, LIN0STAPR1 | | √ | √ | √ | √ | √ | √ | √ |
| INTLIN0 | LIN0IF | | LIN0MK | | LIN0PR0, LIN0PR1 | | √ | √ | √ | √ | √ | √ | √ |
| INTIICA0 | IICAIF0 | | IICAMK0 | | IICAPR00, IICAPR10 | | √ | √ | √ Note | √ Note | √ Note | – | – |
| INTP8 | PIF8 | | PMK8 | | PPR08, PPR18 | | √ | √ | √ Note | √ Note | – | – | – |
| INTRTC | RTCIF | | RTCMK | | RTCPR0, RTCPR1 | | √ | √ | √ | √ | √ | √ | √ |
| INTRM00 | TMIF00 | | TMMK00 | | TMPR000, TMPR100 | | √ | √ | √ | √ | √ | √ | √ |
| INTTM01 | TMIF01 | | TMMK01 | | TMPR001, TMPR101 | | √ | √ | √ | √ | √ | √ | √ |
| INTTM02 | TMIF02 | | TMMK02 | | TMPR002, TMPR102 | | √ | √ | √ | √ | √ | √ | √ |
| INTTM03 | TMIF03 | | TMMK03 | | TMPR003, TMPR103 | | √ | √ | √ | √ | √ | √ | √ |
| INTAD | ADIF | IF1H | ADMK | MK1H | ADPR0, ADPR1 | PR01H, PR11H | √ | √ | √ | √ | √ | √ | √ |
| INTP6 | PIF6 | | PMK6 | | PPR06, PPR16 | | √ | √ | √ | √ | – | – | – |
| INTTM11H | TMIF11H | | TMMK11H | | TMPR011H, TMPR111H | | √ | √ | √ Note | √ Note | √ Note | √ Note | – |
| INTP7 | PIF7 | | PMK7 | | PPR07, PPR17 | | √ | √ | √ | √ | – | – | – |
| INTTM13H | TMIF13H | | TMMK13H | | TMPR013H, TMPR113H | | √ | √ | √ Note | √ Note | √ Note | √ Note | – |
| INTP9 | PIF9 | | PMK9 | | PPR09, PPR19 | | √ | √ | √ Note | √ Note | – | – | – |
| INTTM01H | TMIF01H | | TMMK01H | | TMPR001H, TMPR101H | | √ | √ | √ | √ | √ | √ | √ |
| INTP10 | PIF10 | | PMK10 | | PPR10, PPR110 | | √ | √ | √ Note | – | – | – | – |
| INTTM03H | TMIF03H | | TMMK03H | | TMPR003H, TMPR103H | | √ | √ | √ | √ | √ | √ | √ |
| INTST1 | STIF1 | | STMK1 | | STPR01, STPR11 | | √ | √ | √ Note | √ Note | √ | √ | – |
| INTCSI10 | CSIIF10 | | CSIMK10 | | CSIPR010, CSIPR110 | | √ | √ | √ Note | √ Note | √ | √ | – |
| INTIIC10 | IICIF10 | | IICMK10 | | IICPR010, IICPR110 | | √ | √ | √ Note | √ Note | √ | √ | – |
| INTSR1 | SRIF1 | | SRMK1 | | SRPR01, SRPR11 | | √ | √ | √ Note | √ Note | √ | √ | – |
| INTCSI11 | CSIIF11 | | CSIMK11 | | CSIPR011, CSIPR111 | | √ | √ | √ Note | √ Note | – | – | – |
| INTIIC11 | IICIF11 | | IICMK11 | | IICPR011, IICPR111 | | √ | √ | √ Note | √ Note | – | – | – |
| INTTM04 | TMIF04 | | TMMK04 | | TMPR004, TMPR104 | | √ | √ | √ | √ | √ | √ | √ |

(**Cautions** and **Note** are listed on the next page.)

**Cautions 1. Do not use INTP6 and channel 1 of TAU1 (at 8-bit timer operation) at the same time because they share flags for the interrupt request sources. Whether the interrupt source condition that is satisfied is INTP6 or channel 1 of TAU1, bit 1 of the IF1H register is set to 1. Bit 1 of the MK1H, PR01H, and PR11H registers supports these two interrupt sources.**

**2. Do not use INTP7 and channel 3 of TAU1 (at 8-bit timer operation) at the same time because they share flags for the interrupt request sources. Whether the interrupt source condition that is satisfied is INTP7 or channel 3 of TAU1, bit 2 of the IF1H register is set to 1. Bit 2 of the MK1H, PR01H, and PR11H registers supports these two interrupt sources.**

**3. Do not use INTP9 and channel 1 of TAU0 (at 8-bit timer operation) at the same time because they share flags for the interrupt request sources. Whether the interrupt source condition that is satisfied is INTP6 or channel 1 of TAU0, bit 3 of the IF1H register is set to 1. Bit 3 of the MK1H, PR01H, and PR11H registers supports these two interrupt sources.**

**4. Do not use INTP10 and channel 3 of TAU0 (at 8-bit timer operation) at the same time because they share flags for the interrupt request sources. Whether the interrupt source condition that is satisfied is INTP6 or channel 3 of TAU0, bit 4 of the IF1H register is set to 1. Bit 4 of the MK1H, PR01H, and PR11H registers supports these two interrupt sources.**

**5. If one of the interrupt sources INTST1, INTCSI10, and INTIIC10 is generated, bit 5 of the IF1H register is set to 1. Bit 5 of the MK1H, PR01H, and PR11H registers supports these three interrupt sources.**

**6. If one of the interrupt sources INTSR1, INTCSI11, and INTIIC11 is generated, bit 6 of the IF1H register is set to 1. Bit 6 of the MK1H, PR01H, and PR11H registers supports these three interrupt sources.**

**Note** Not provided in Group A products.

**Table 21-2. Flags Corresponding to Interrupt Request Sources (4/4)**

| Interrupt Source | Interrupt Request Flag | | Interrupt Mask Flag | | Priority Specification Flag | | 100-pin | 80-pin | 64-pin | 48-pin | 32-pin | 30-pin | 20-pin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Register | | Register | | Register | | | | | | | |
| INTTM05 | TMIF05 | IF2L | TMMK05 | MK2L | TMPR005, TMPR105 | PR02L, PR12L | √ | √ | √ | √ | √ | √ | √ |
| INTTM06 | TMIF06 | | TMMK06 | | TMPR006, TMPR106 | | √ | √ | √ | √ | √ | √ | √ |
| INTTM07 | TMIF07 | | TMMK07 | | TMPR007, TMPR107 | | √ | √ | √ | √ | √ | √ | √ |
| INTP11 | PIF11 | | PMK11 | | PPR11, PPR111 | | √ | √ | √ Note 1 | – | – | – | – |
| INTLIN0WUP | LIN0WUPIF | | LIN0WUPMK | | LIN0WUPPR0, LIN0WUPPR1 | | √ | √ | √ | √ | √ | √ | √ |
| INTKR | KRIF | | KRMK | | KRPR0, KRPR1 | | √ | √ | √ | √ | √ | √ | √ |
| INTCAN0ERR | CAN0ERRIF | | CAN0ERMK | | CAN0ERRPR0, CA0ERRPR1 | | √ | √ Note 2 | √ Note 2 | √ Note 2 | √ Note 2 | √ Note 2 | – |
| INTCAN0WUP | CAN0WUPIF | | CAN0WUPMK | | CAN0WUPPR0, CAN0WUPPR1 | | √ | √ Note 2 | √ Note 2 | √ Note 2 | √ Note 2 | √ Note 2 | – |
| INTCAN0CFR | CAN0CFRIF | | CAN0CFRMK | | CAN0CFRPR0, CAN0CFRPR1 | | √ | √ Note 2 | √ Note 2 | √ Note 2 | √ Note 2 | √ Note 2 | – |
| INTCAN0TRM | CAN0TRMIF | IF2H | CAN0TRMMK | MK2H | CAN0TRMPR0, CAN0TRMPR1 | PR02H, PR12H | √ | √ Note 2 | √ Note 2 | √ Note 2 | √ Note 2 | √ Note 2 | – |
| INTCANGRFR | CANGRFRIF | | CANGRFRMK | | CANGRFRPR0, CANGRFRPR1 | | √ | √ Note 2 | √ Note 2 | √ Note 2 | √ Note 2 | √ Note 2 | – |
| INTCANGERR | CANGERRIF | | CANGERRMK | | CANGERRPR0, CANGERRPR1 | | √ | √ Note 2 | √ Note 2 | √ Note 2 | √ Note 2 | √ Note 2 | – |
| INTTM10 | TMIF10 | | TMMK10 | | TMPR010, TMPR110 | | √ | √ | √ Note 1 | √ Note 1 | √ Note 2 | √ Note 2 | – |
| INTTM11 | TMIF11 | | TMMK11 | | TMPR011, TMPR111 | | √ | √ | √ Note 1 | √ Note 1 | √ Note 2 | √ Note 2 | – |
| INTTM12 | TMIF12 | | TMMK12 | | TMPR012, TMPR112 | | √ | √ | √ Note 1 | √ Note 1 | √ Note 2 | √ Note 2 | – |
| INTTM13 | TMIF13 | | TMMK13 | | TMPR013, TMPR113 | | √ | √ | √ Note 1 | √ Note 1 | √ Note 2 | √ Note 2 | – |
| INTFL | FLIF | | FLMK | | FLPR0, FLPR1 | | √ | √ | √ | √ | √ | √ | √ |
| INTP12 | PIF12 | IF3L | PMK12 | MK3L | PPR012, PPR112 | PR03L, PR13L | √ | √ Note 3 | √ Note 3 | – | – | – | – |
| INTLIN1WUP | LIN1WUPIF | | LIN1WUPMK | | LIN1WUPPR0, LIN1WUPPR1 | | √ | √ Note 3 | √ Note 3 | √ Note 3 | – | – | – |
| INTLIN1TRM | LIN1TRMIF | | LIN1TRMMK | | LIN1TRMPR0, LIN1TRMPR1 | | √ | √ Note 3 | √ Note 3 | √ Note 3 | – | – | – |
| INTLIN1RVC | LIN1RVCIF | | LIN1RVCMK | | LIN1RVCPR0, LIN1RVCPR1 | | √ | √ Note 3 | √ Note 3 | √ Note 3 | – | – | – |
| INTLIN1STA | LIN1STAIF | | LIN1STAMK | | LIN1STAPR0, LIN1STAPR1 | | √ | √ Note 3 | √ Note 3 | √ Note 3 | – | – | – |
| INTLIN1 | LIN1IF | | LIN1MK | | LIN1PR0, LIN1PR1 | | √ | √ Note 3 | √ Note 3 | √ Note 3 | – | – | – |
| INTTM14 | TMIF14 | | TMMK14 | | TMPR014, TMPR114 | | √ | √ Note 3 | √ Note 3 | √ Note 3 | – | – | – |
| INTTM15 | TMIF15 | | TMMK15 | | TMPR015, TMPR115 | | √ | √ Note 3 | √ Note 3 | √ Note 3 | – | – | – |
| INTTM16 | TMIF16 | | TMMK16 | | TMPR016, TMPR116 | | √ | √ Note 3 | √ Note 3 | √ Note 3 | – | – | – |
| INTTM17 | TMIF17 | | TMMK17 | | TMPR017, TMPR117 | | √ | √ Note 3 | √ Note 3 | √ Note 3 | – | – | – |

(**Notes** are listed on the next page.)

**Notes 1.** Not provided in Group A products.

**2.** Not provided in products of Groups A and B.

**3.** Provided only in Group E products.

### 21.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

The IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, and IF3L registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the IF0L and IF0H registers, the IF1L and IF1H registers, and the IF2L and IF2H registers are combined to form 16-bit registers IF0, IF1, and IF2, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Remark** If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

**Figure 21-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L) (1/2)**

Address: FFFE0H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IF0L | PIF5<br>CMPIF0 | PIF4<br>SPMIF | PIF3 | PIF2 | PIF1 | PIF0 | LVIIF | WDTIIF |

Address: FFFE1H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IF0H | LIN0TRMIF | RAMIF | TRJIIF0 | TRDIF1 | TRDIF0 | SRIF0<br>CSIIF01<br>IICIF01 | STIF0<br>CSIIF00<br>IICIF00 | CLMIF<br>PIF13 |

Address: FFFE2H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IF1L | TMIF03 | TMIF02 | TMIF01 | TMIF00 | PIF8<br>RTCIF | IICAIF0 | LIN0STAIF<br>LIN0IF | LIN0RVCIF |

Address: FFFE3H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IF1H | TMIF04 | SRIF1<br>CSIIF11<br>IICIF11 | STIF1<br>CSIIF10<br>IICIF10 | PIF10<br>TMIF03H | PIF9<br>TMIF01H | PIF7<br>TMIF13H | PIF6<br>TMIF11H | ADIF |

Address: FFFD0H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IF2L | CAN0CFRIF | CAN0WUPIF | CAN0ERRIF | KRIF | PIF11<br>LIN0WUPIF | TMIF07 | TMIF06 | TMIF05 |

**Figure 21-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L) (2/2)**

Address: FFFD1H After reset: 00H R/W

| Symbol | <7> | <6> | 5 | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IF2H | FLIF | TMIF13 | TMIF12 | TMIF11 | TMIF10 | CANGERRIF | CANGREFRIF | CAN0TRMIF |

Address: FFFD2H After reset: 00H R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| IF3L | TMIF17 | TMIF16 | TMIF15 | TMIF14 | LIN1STAIF<br>LIN1IF | LIN1RVCIF | LIN1TRMIF | PIF12<br>LIN1WUPIF |

| IFxx | Interrupt request flag |
|---|---|
| 0 | No interrupt request signal is generated |
| 1 | Interrupt request is generated, interrupt request status |

**Cautions**

1. **The above is the bit layout for the 100-pin. The available bits differ depending on the product. For details about the bits available for each product, see Table 21-2. Be sure to clear bits that are not available to 0.**
2. **When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.**
3. **When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).**
   **If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.**

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

**In this case, even if the request flag of the another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.**

### 21.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

The MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, and MK3L registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the MK0L and MK0H registers, the MK1L and MK1H registers, and the MK2L and MK2H registers are combined to form 16-bit registers MK0, MK1, and MK2, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

**Remark** If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

**Figure 21-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L)**

Address: FFFE4H After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| MK0L | PMK5<br>CMPMK0 | PMK4<br>SPMMK | PMK3 | PMK2 | PMK1 | PMK0 | LVIMK | WDTIMK |

Address: FFFE5H After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| MK0H | LIN0TRMMK | RAMMK | TRJMK0 | TRDMK1 | TRDMK0 | SRMK0<br>CSIMK01<br>IICMK01 | STMK0<br>CSIMK00<br>IICMK00 | CLMMK<br>PMK13 |

Address: FFFE6H After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| MK1L | TMMK03 | TMMK02 | TMMK01 | TMMK00 | PMK8<br>RTCMK | IICAMK0 | LIN0STAMK<br>LIN0MK | LIN0RVCMK |

Address: FFFE7H After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| MK1H | TMMK04 | SRMK1<br>CSIMK11<br>IICMK11 | STMK1<br>CSIMK10<br>IICMK10 | PMK10<br>TMMK03H | PMK9<br>TMMK01H | PMK7<br>TMMK13H | PMK6<br>TMMK11H | ADMK |

Address: FFFD4H After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| MK2L | CAN0CFRMK | CAN0WUPMK | CAN0ERRMK | KRMK | PMK11<br>LIN0WUPMK | TMMK07 | TMMK06 | TMMK05 |

Address: FFFD5H After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| MK2H | FLMK | TMMK13 | TMMK12 | TMMK11 | TMMK10 | CANGERRMK | CANGRFRMK | CAN0TRMMK |

Address: FFFD6H After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| MK3L | TMMK17 | TMMK16 | TMMK15 | TMMK14 | LIN1STAMK<br>LIN1MK | LIN1RVCMK | LIN1TRMMK | PMK12<br>LIN1WUPMK |

| MKxx | Interrupt servicing control |
|---|---|
| 0 | Interrupt servicing enabled |
| 1 | Interrupt servicing disabled |

**Caution The above is the bit layout for the 100-pin. The available bits differ depending on the product. For details about the bits available for each product, see Table 21-2. Be sure to set bits that are not available to 1.**

### 21.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L)

The priority specification flag registers are used to set the corresponding maskable interrupt priority level.

A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L, 2H, or 3L).

The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and the PR13L registers can be set by a 1-bit or 8-bit memory manipulation instruction. If the PR00L and PR00H registers, the PR01L and PR01H registers, the PR02L and PR02H registers, the PR10L and PR10H registers, the PR11L and PR11H registers, and the PR12L and PR12H registers are combined to form 16-bit registers PR00, PR01, PR02, PR10, PR11, and PR12, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

**Remark** If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

**Figure 21-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (1/3)**

Address: FFFE8H After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR00L | PPR05<br>CMPPR00 | PPR04<br>SPMPR0 | PPR03 | PPR02 | PPR01 | PPR00 | LVIPR0 | WDTIPR0 |

Address: FFFECH After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR10L | PPR15<br>CMPPR10 | PPR14<br>SPMPR1 | PPR13 | PPR12 | PPR11 | PPR10 | LVIPR1 | WDTIPR1 |

Address: FFFE9H After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR00H | LIN0TRMPR0 | RAMPR0 | TRJPR00 | TRDPR01 | TRDPR00 | SRPR00<br>CSIPR001<br>IICPR001 | STPR00<br>CSIPR000<br>IICPR000 | CLMPR0<br>PPR013 |

Address: FFFEDH After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR10H | LIN0TRMPR1 | RAMPR1 | TRJPR10 | TRDPR11 | TRDPR10 | SRPR10<br>CSIPR101<br>IICPR101 | STPR10<br>CSIPR100<br>IICPR100 | CLMPR1<br>PPR113 |

Address: FFFEAH After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR01L | TMPR003 | TMPR002 | TMPR001 | TMPR000 | PPR08<br>RTCPR0 | IICAPR00 | LIN0STAPR0<br>LIN0PR0 | LIN0RVCPR0 |

**Figure 21-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR3L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (2/3)**

Address: FFFEEH After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR11L | TMPR103 | TMPR102 | TMPR101 | TMPR100 | PPR18<br>RTCPR1 | IICAPR10 | LIN0STAPR1<br>LIN0PR1 | LIN0RVCPR1 |

Address: FFFEBH After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR01H | TMPR004 | SRPR01<br>CSIPR011<br>IICPR011 | STPR01<br>CSIPR010<br>IICPR010 | PPR010<br>TMPR003H | PPR09<br>TMPR001H | PPR07<br>TMPR013H | PPR06<br>TMPR011H | ADPR0 |

Address: FFFEFH After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR11H | TMPR104 | SRPR11<br>CSIPR111<br>IICPR111 | STPR11<br>CSIPR110<br>IICPR110 | PPR110<br>TMPR103H | PPR19<br>TMPR101H | PPR17<br>TMPR113H | PPR16<br>TMPR111H | ADPR1 |

Address: FFFD8H After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR02L | CAN0CFRPR0 | CAN0WUPPR0 | CAN0ERRPR0 | KRPR0 | PPR011<br>LIN0WUPPR0 | TMPR007 | TMPR006 | TMPR005 |

Address: FFFDCH After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR12L | CAN0CFRPR1 | CAN0WUPPR1 | CAN0ERRPR1 | KRPR1 | PPR111<br>LIN0WUPPR1 | TMPR107 | TMPR106 | TMPR105 |

Address: FFFD9H After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR02H | FLPR0 | TMPR013 | TMPR012 | TMPR011 | TMPR010 | CANGERRPR0 | CANGRFRPR0 | CAN0TRMPR0 |

Address: FFFDDH After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR12H | FLPR1 | TMPR113 | TMPR112 | TMPR111 | TMPR110 | CANGERRPR 1 | CANGRFRPR1 | CAN0TRMPR1 |

**Figure 21-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR3L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L) (3/3)**

Address: FFFDAH After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR03L | TMPR017 | TMPR016 | TMPR015 | TMPR014 | LIN1STAPR0<br>LIN1PR0 | LIN1RVCPR0 | LIN1TRMPR0 | PPR012<br>LIN1WUPPR0 |

Address: FFFDEH After reset: FFH R/W

| Symbol | <7> | <6> | <5> | <4> | <3> | <2> | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| PR13L | TMPR117 | TMPR116 | TMPR115 | TMPR114 | LIN1STAPR1<br>LIN1PR1 | LIN1RVCPR1 | LIN1TRMPR1 | PPR112<br>LIN1WUPPR1 |

| XXPR1X | XXPR0X | Priority level selection |
|---|---|---|
| 0 | 0 | Specify level 0 (high priority level) |
| 0 | 1 | Specify level 1 |
| 1 | 0 | Specify level 2 |
| 1 | 1 | Specify level 3 (low priority level) |

**Caution The above is the bit layout for the 100-pin. The available bits differ depending on the product. For details about the bits available for each product, see Table 21-2. Be sure to set bits that are not available to 1.**

### 21.3.4 External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)

These registers specify the valid edge for INTP0 to INTP13.

The EGP0, EGP1, EGN0, and EGN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Figure 21-5. Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1)**

Address: FFF38H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| EGP0 | EGP7 | EGP6 | EGP5 | EGP4 | EGP3 | EGP2 | EGP1 | EGP0 |

Address: FFF39H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| EGN0 | EGN7 | EGN6 | EGN5 | EGN4 | EGN3 | EGN2 | EGN1 | EGN0 |

Address: FFF3AH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| EGP1 | 0 | 0 | EGP13 | EGP12 | EGP11 | EGP10 | EGP9 | EGP8 |

Address: FFF3BH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| EGN1 | 0 | 0 | EGN13 | EGN12 | EGN11 | EGN10 | EGN9 | EGN8 |

| EGPn | EGNn | INTPn pin valid edge selection (n = 0 to 13) |
|---|---|---|
| 0 | 0 | Edge detection disabled |
| 0 | 1 | Falling edge |
| 1 | 0 | Rising edge |
| 1 | 1 | Both rising and falling edges |

Table 21-3 shows the ports corresponding to the EGPn and EGNn bits.

**Table 21-3. Ports Corresponding to EGPn and EGNn bits**

| Detection Enable Bit | | Edge Detection Port | Interrupt Request Signal | 80, 100-pin | 64-pin | 48-pin | 30, 32-pin | 20-pin |
|---|---|---|---|---|---|---|---|---|
| EGP0 | EGN0 | P137 | INTP0 | √ | √ | √ | √ | √ |
| EGP1 | EGN1 | P125 | INTP1 | √ | √ | √ | √ | √ |
| EGP2 | EGN2 | P30(P31) | INTP2 | √ | √ | √ | √ | √ |
| EGP3 | EGN3 | P17 (P50) | INTP3 | √ | √ | √ | √ | √ |
| EGP4 | EGN4 | P120 | INTP4 | √ | √ | √ | √ | √ |
| EGP5 | EGN5 | P12 | INTP5 | √ | √ | √ | √ | – |
| EGP6 | EGN6 | P71 | INTP6 | √ | √ | √ | – | – |
| EGP7 | EGN7 | P32 | INTP7 | √ | √ | √ | – | – |
| EGP8 | EGN8 | P70 | INTP8 | √ | √ Note 1 | √ Note 1 | – | – |
| EGP9 | EGN9 | P00 | INTP9 | √ | √ Note 1 | √ Note 1 | – | – |
| EGP10 | EGN10 | P53 | INTP10 | √ | √ Note 1 | – | – | – |
| EGP11 | EGN11 | P51 | INTP11/ INTLIN0WUP Note 2 | √ | √ | √ | √ | √ |
| EGP12 | EGN12 | P77 | INTP12/ INTLIN1WUP Note 2 | √ Note 3 | √ Note 3 | – | – | – |
| EGP13 | EGN13 | P47 | INTP13 | √ Note 3 | – | – | – | – |

**Notes 1.** Not provided in Group A products.

**2.** Set the EGP1 and EGN1 registers before the INTLIN0WUP and INTLIN1WUP interrupts are generated.

**3.** Provided only in Group E products.

**Caution When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge. When switching the input port pins to the output mode, set the port mode register (PMxx) to 0 after disabling the edge detection (by setting EGPn and EGNn to 0).**

**Remarks 1.** For edge detection ports, see **2.1 Pin Function List**.

**2.** n = 0 to 13

### 21.3.5 Interrupt source determination flag register 0 (INTFLG0)

This register determines which of interrupt sources causes an interrupt, an external interrupt source (INTP4, 5, 8, 13) or other interrupt source that are allocated to the same vector table address as the comparator detection 0 interrupt source.

The flag in this register cannot be set by software.

The flags are cleared by software.

To clear the flag, write 1 to the bits other than the bit desired to be cleared.

Use an 8-bit data transfer instruction as a write instruction.

**Figure 21-6. Format of Interrupt Source Determination Flag Register 0 (INTFLG0)**

Address: F0079H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| INTFLG0 | INTFLG07 Note 4, 6 | INTFLG06 Notes 3, 5 | 0 | 0 | 0 | INTFLG02 Notes 1, 2, 6 | INTFLG01 Note 1, 6 | INTFLG00 Note 6 |

| INTFLG07 Note 4, 6 | Interrupt source determination flag at vector table address 00014h |
|---|---|
| 0 | An INTP13 interrupt has not been generated. |
| 1 | An INTP13 interrupt has been generated. |

| INTFLG06 Notes 3, 5 | Interrupt source determination flag at vector table address 00012h |
|---|---|
| 0 | A comparator detection 0 interrupt has not been generated. |
| 1 | A comparator detection 0 interrupt has been generated. |

| INTFLG02 Notes 1, 2, 6 | Interrupt source determination flag at vector table address 0002Ah |
|---|---|
| 0 | An INTP8 interrupt has not been generated. |
| 1 | An INTP8 interrupt has been generated. |

| INTFLG01 Note 1, 6 | Interrupt source determination flag at vector table address 00012h |
|---|---|
| 0 | An INTP5 interrupt has not been generated. |
| 1 | An INTP5 interrupt has been generated. |

| INTFLG00 Note 6 | Interrupt source determination flag at vector table address 00010h |
|---|---|
| 0 | An INTP4 interrupt has not been generated. |
| 1 | An INTP4 interrupt has been generated. |

**Notes 1.** Not provided in Group A products.

**2.** Not provided in the RL78/F13 (CAN and LIN incorporated) products with 30 or 32 pins and the RL78/F14 products with 30 or 32 pins.

**3.** Only provided in products of Groups D and E.

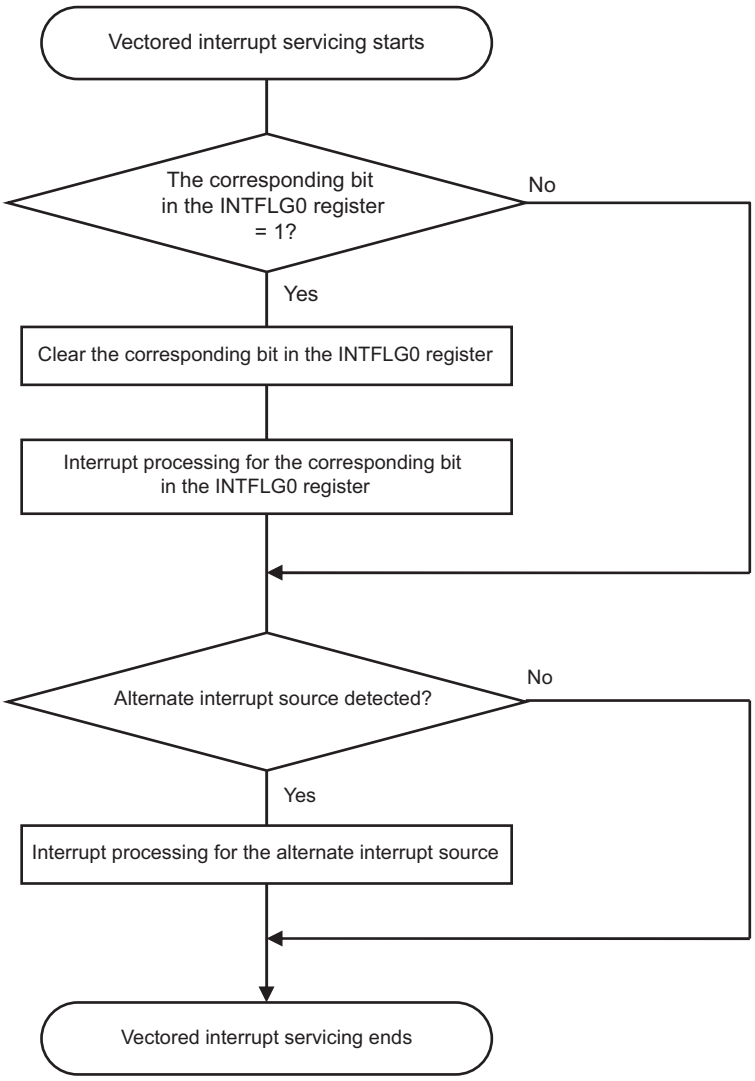
**4.** Only provided in 80-pin products of the RL78/F14 with 128 Kbytes to 256 Kbytes of code flash memory and 100-pin products of the RL78/F14 with 64 Kbytes to 256 Kbytes of code flash memory.

**5.** Even if the RPTINT bit in the DTCCRj register (j = 0 to 23) is set to 0 (disabling the interrupt while the DTC module is in repeat mode), when the comparator detection 0 interrupt source is generated, the INTFLG06 bit is set to 1. For details, see (A) Internal maskable interrupt (only comparator detection 0 interrupt) in **Figure 21-1. Basic Configuration of Interrupt Function**.

**6.** If an INTPn interrupt is generated, the bit m in the interrupt source determination flag register 0 (INTFLG0) is set regardless of the settings of the bits in the interrupt mask flag register (MKxx) and interrupt mask register (INTMSK).
m: Bit number (m = 0, 1, 2, 7), n: INTP interrupt number (n = 4, 5, 8, 13)

The interrupt sources INTP4, INTP5, INTP8, and INTP13 are function-multiplexed with other interrupts. INTFLG0 register is used to determine which of the interrupt sources causes an interrupt. Figure 21-7 shows the flowchart of interrupt processing using the interrupt source determination flag.

**Figure 21-7. Flowchart of Interrupt Processing Using the Interrupt Source Determination Flag**



### 21.3.6 Interrupt mask register (INTMSK)

The interrupt mask register in the interrupt control circuit is used to mask interrupt requests corresponding to an INTPn interrupt that is to be used as an event signal for the ELC or a source for DTC activation.

Set this register when the INTP4 to INTP6 signals are used not as interrupt sources for the interrupt control circuit but as only ELC event signals or DTC activation sources.

This register can be set by an 8-bit memory manipulation instruction

Reset signal generation sets this register to FFH.

**Figure 21-8. Format of Interrupt Mask Register (INTMSK)**

Address: F007CH After reset: FFH R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| INTMSK | 1 | 1 | 1 | 1 | 1 | INTMSK2 Notes 1, 2 | INTMSK1 Note 1 | INTMSK0 |

| INTMSK2 Notes 1, 2 | Setting masking for INTP6 interrupt source to the interrupt control circuit |
|---|---|
| 0 | Requests to the interrupt control circuit and DTC are enabled. |
| 1 | Requests to the interrupt control circuit are disabled, and requests to DTC are enabled. |

| INTMSK1 Note 1 | Setting masking for INTP5 interrupt source to the interrupt control circuit Note 3 |
|---|---|
| 0 | Requests to the interrupt control circuit, ELC, and DTC are enabled. |
| 1 | Requests to the interrupt control circuit are disabled, and requests to ELC and DTC are enabled. |

| INTMSK0 | Setting masking for INTP4 interrupt source to the interrupt control circuit Note 3 |
|---|---|
| 0 | Requests to the interrupt control circuit, ELC, and DTC are enabled. |
| 1 | Requests to the interrupt control circuit are disabled, and requests to ELC and DTC are enabled. |

**Notes 1.** Not provided in the 20-pin products.

**2.** Not provided in the products with 30 or 32 pins.

**3.** If an INTPn interrupt is generated, the bit m in the interrupt source determination flag register 0 (INTFLG0) is set regardless of the setting of the bit in the interrupt mask register.
n: INTP interrupt number (n = 4, 5), m: Bit number (m = 0,1)

### 21.3.7 Input switch control register (ISC)

The ISC0 bit of the ISC register is used for the LIN-bus communication with UART0. The ISC2 and ISC3 bits are used for the LIN/UART module (RLIN3). When the ISC0 bit is set to 1, set the TIS17 and TIS16 bits in the TIS1 register (timer input select register 1) at the same time.

Setting bit 0 to 1 selects the input signal of the serial data input pin (RxD0) as the external interrupt input (INTP0), which allows detection of the wakeup signal by the INTP0 interrupt.

Setting bits 2 and 3 to 1 select the input signal of the serial data input pin (RxD0) for the LIN/UART module as the external interrupt input.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

**Figure 21-9. Format of Input Switch Control Register (ISC)**

Address: F0073H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | <3> | <2> | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| ISC | 0 | 0 | 0 | 0 | ISC3 | ISC2 | 0 | ISC0 |

| ISC3 | Input selection for external interrupt INTP12 |
|---|---|
| 0 | INTP12 pin input signal is selected as external interrupt input. |
| 1 | LRxD1 pin input signal is selected as external interrupt input. |

| ISC2 | Input selection for external interrupt INTP11 |
|---|---|
| 0 | INTP11 pin input signal is selected as external interrupt input. |
| 1 | LRxD0 pin input signal is selected as external interrupt input. |

| ISC0 | Input selection for external interrupt INTP0 |
|---|---|
| 0 | INTP0 pin input signal is selected as external interrupt input. (normal operation) |
| 1 | RxD0 pin input signal is selected as external interrupt input. (wake-up signal detection) |

**Caution Bits 7 to 4 and 1 should always be set to 0.**

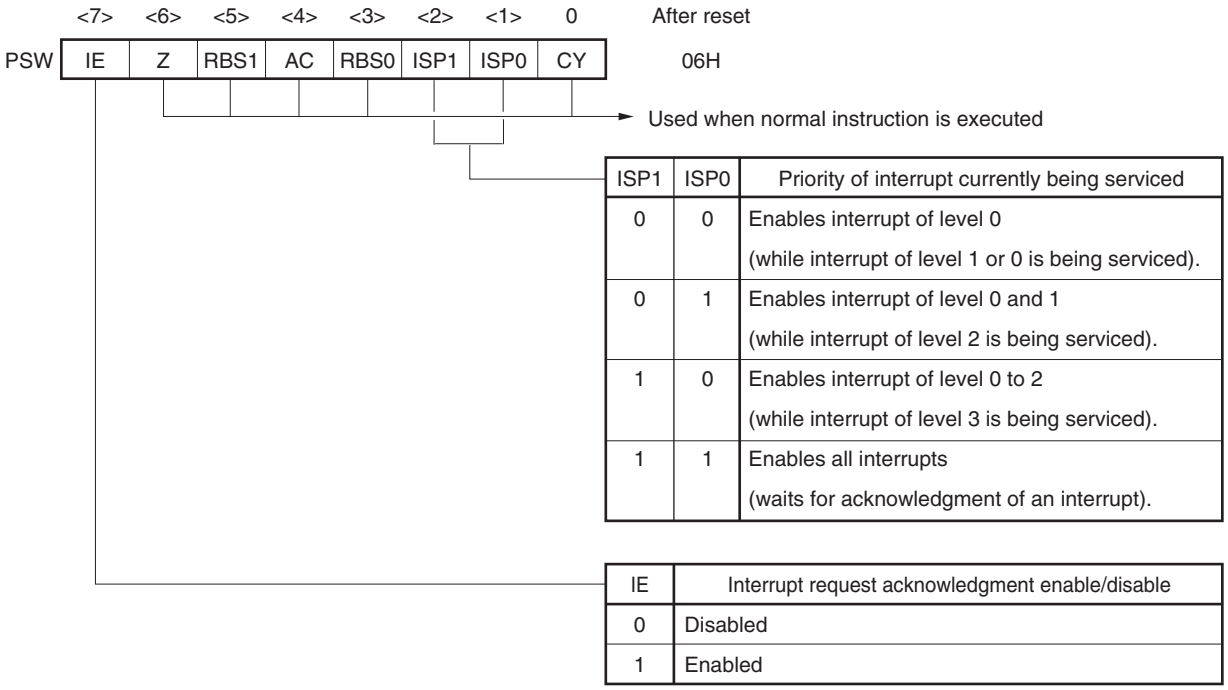
### 21.3.8 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that control multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. When a maskable interrupt request is acknowledged, if the value of the bits in the priority specification flag register which correspond to that interrupt is not 00, the value minus 1 is transferred to the ISP0 and ISP1 flags. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset signal generation sets PSW to 06H.

**Figure 21-10. Configuration of Program Status Word**




| ISP1 | ISP0 | Priority of interrupt currently being serviced |
|---|---|---|
| 0 | 0 | Enables interrupt of level 0<br>(while interrupt of level 1 or 0 is being serviced). |
| 0 | 1 | Enables interrupt of level 0 and 1<br>(while interrupt of level 2 is being serviced). |
| 1 | 0 | Enables interrupt of level 0 to 2<br>(while interrupt of level 3 is being serviced). |
| 1 | 1 | Enables all interrupts<br>(waits for acknowledgment of an interrupt). |

| IE | Interrupt request acknowledgment enable/disable |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

## 21.4 Interrupt Servicing Operations

### 21.4.1 Maskable interrupt request acknowledgment

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request.

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 21-4 below.

For the interrupt request acknowledgment timing, see **Figures 21-12** and **21-13**.

**Table 21-4. Time from Generation of Maskable Interrupt Until Servicing**

| | Minimum Time | Maximum Time[Note] |
|---|---|---|
| Servicing time | 9 clocks | 16 clocks |

**Note** Maximum time does not apply when an instruction from the internal RAM area is executed.

**Remark** 1 clock: $1/f_{CLK}$ ($f_{CLK}$: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

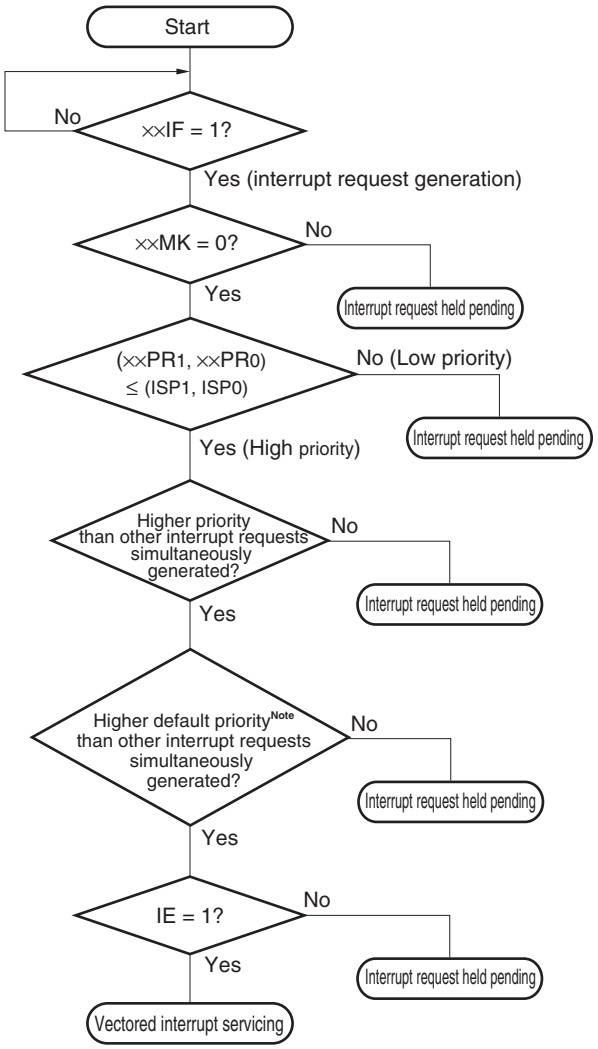
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 21-11 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

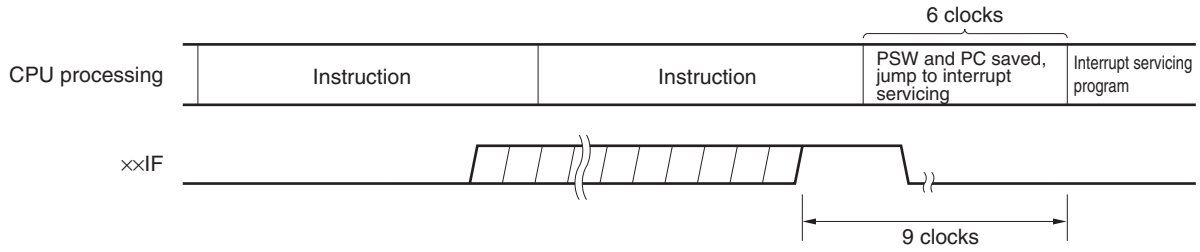
**Figure 21-11. Interrupt Request Acknowledgment Processing Algorithm**




××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR0: Priority specification flag 0
××PR1: Priority specification flag 1
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1: Flag that indicates the priority level of the interrupt currently being serviced

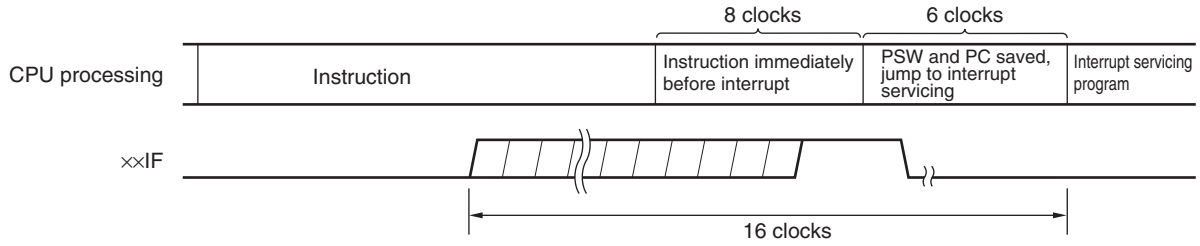
**Note** For the default priority, see **Table 21-1 Interrupt Source List**.

**Figure 21-12. Interrupt Request Acknowledgment Timing (Minimum Time)**




**Remark** 1 clock: 1/$f_{CLK}$ ($f_{CLK}$: CPU clock)

**Figure 21-13. Interrupt Request Acknowledgment Timing (Maximum Time)**




**Remark** 1 clock: 1/$f_{CLK}$ ($f_{CLK}$: CPU clock)

### 21.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (0007EH, 0007FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

**Caution Can not use the RETI instruction for restoring from the software interrupt.**

### 21.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority equal to or lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. However, when setting the IE flag to 1 during the interruption at level 0, other level 0 interruptions can be allowed.

Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 21-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 21-14 shows multiple interrupt servicing examples.

**Table 21-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing**

| Multiple Interrupt Request / Interrupt Being Serviced | | Maskable Interrupt Request | | | | | | | | Software Interrupt Request |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Priority Level 0 (PR = 00) | | Priority Level 1 (PR = 01) | | Priority Level 2 (PR = 10) | | Priority Level 3 (PR = 11) | | |
| | | IE = 1 | IE = 0 | IE = 1 | IE = 0 | IE = 1 | IE = 0 | IE = 1 | IE = 0 | |
| Maskable interrupt | ISP1 = 0, ISP0 = 0 | ○ | × | × | × | × | × | × | × | ○ |
| | ISP1 = 0, ISP0 = 1 | ○ | × | ○ | × | × | × | × | × | ○ |
| | ISP1 = 1, ISP0 = 0 | ○ | × | ○ | × | ○ | × | × | × | ○ |
| | ISP1 = 1, ISP0 = 1 | ○ | × | ○ | × | ○ | × | ○ | × | ○ |
| Software interrupt | | ○ | × | ○ | × | ○ | × | ○ | × | ○ |

**Remarks 1.** ○: Multiple interrupt servicing enabled

**2.** ×: Multiple interrupt servicing disabled

**3.** ISP0, ISP1, and IE are flags contained in the PSW.

ISP1 = 0, ISP0 = 0: Enables interrupt of level 0 (an interrupt of level 1 or level 0 is being serviced).

ISP1 = 0, ISP0 = 1: Enables interrupt of level 0 and 1 (an interrupt of level 2 is being serviced).

ISP1 = 1, ISP0 = 0: Enables interrupt of level 0 to 2 (an interrupt of level 3 is being serviced).

ISP1 = 1, ISP0 = 1: Enables all interrupts (wait for an interrupt acknowledgment).

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

**4.** PR is a flag contained in the PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L registers.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)

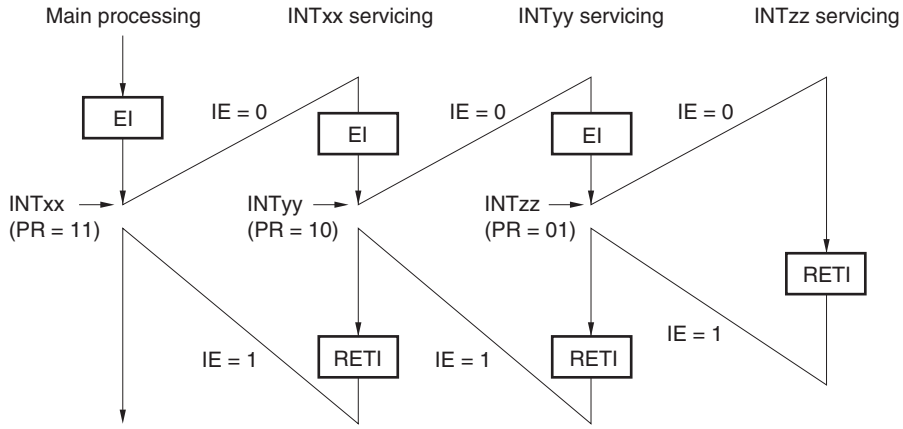
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

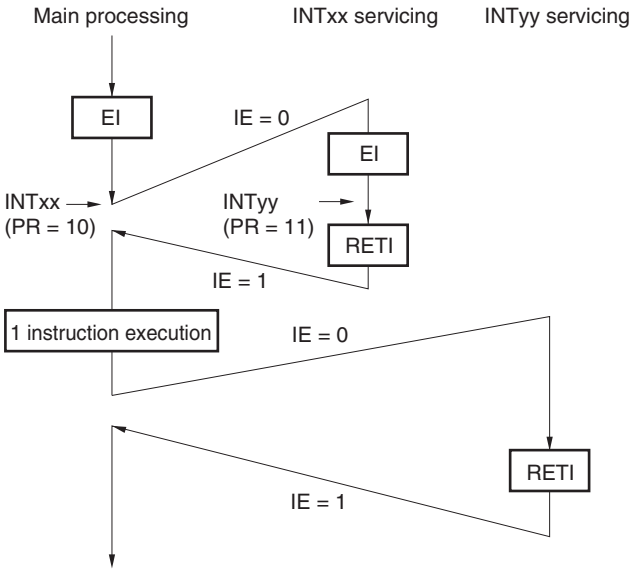
**Figure 21-14. Examples of Multiple Interrupt Servicing (1/2)**

**Example 1. Multiple interrupt servicing occurs twice**




During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

**Example 2. Multiple interrupt servicing does not occur due to priority control**

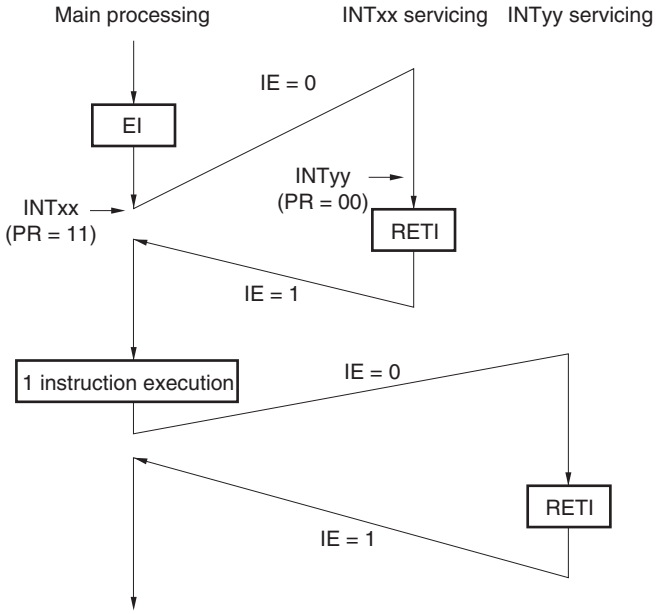



Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

**Figure 21-14. Examples of Multiple Interrupt Servicing (2/2)**

**Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled**




Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

### 21.4.4 Interrupt servicing during division instruction

The RL78/F13 and RL78/F14 handle interrupts during the DIVHU/DIVWU instruction in order to enhance the interrupt response when a division instruction is executed.

- When an interrupt is generated while the DIVHU/DIVWU instruction is executed, the instruction is suspended
- After the instruction is suspended, the PC indicates the next instruction after DIVHU/DIVWU
- An interrupt is generated by the next instruction
- PC-3 is saved in the stack memory to execute the DIVHU/DIVWU instruction again

**Table 21-6. Normal Interrupt Processing and Interrupt Processing while Executing Division Instructions**

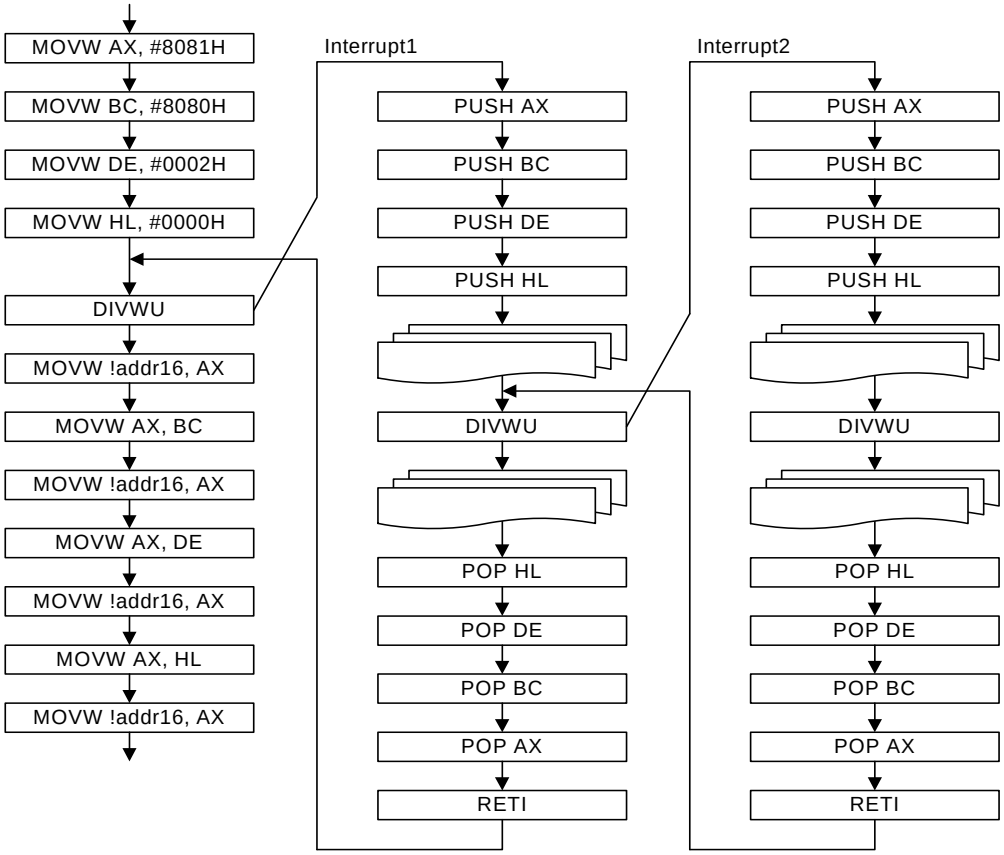
| Normal interrupt | Interrupts while Executing DIVHU/DIVWU Instruction |
|---|---|
| (SP-1) ← PSW | (SP-1) ← PSW |
| (SP-2) ← $(PC)_S$ | (SP-2) ← $(PC-3)_S$ |
| (SP-3) ← $(PC)_H$ | (SP-3) ← $(PC-3)_H$ |
| (SP-4) ← $(PC)_L$ | (SP-4) ← $(PC-3)_L$ |
| $PC_S$ ← 0000 | $PC_S$ ← 0000 |
| $PC_H$ ← (Vector) | $PC_H$ ← (Vector) |
| $PC_L$ ← (Vector) | $PC_L$ ← (Vector) |
| SP ← SP-4 | SP ← SP-4 |
| IE ← 0 | IE ← 0 |

<R> **Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine. Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.**

- **V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.6 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

The AX, BC, DE, and HL registers are used for DIVHU/DIVWU. For the interrupt processing, save these registers in the stack memory.

**Figure 21-15. Example of Interrupt during Division Instruction**



### 21.4.5 Interrupt request hold

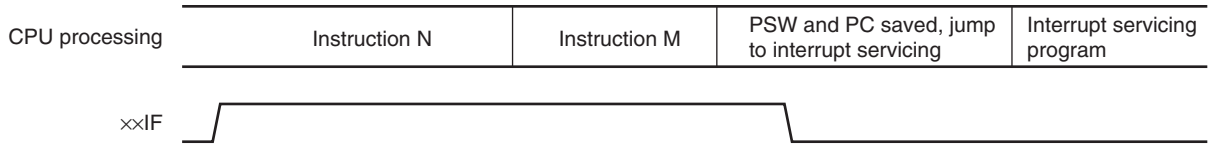
There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- <R> MULHU
- MULH
- MACHU
- MACH
- Manipulation instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PR13L registers

**Caution The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged.**

Figure 21-16 shows the timing at which interrupt requests are held pending.

**Figure 21-16. Interrupt Request Hold**




**Remarks 1.** Instruction N: Interrupt request hold instruction

**2.** Instruction M: Instruction other than interrupt request hold instruction

# CHAPTER 22 KEY INTERRUPT FUNCTION

The number of key interrupt input channels differs, depending on the product.

| | 20-pin | 32-pin | 30, 48, 64, 80, 100-pin |
|---|---|---|---|
| Key interrupt input channels | 2 ch | 6 ch | 8 ch |

## 22.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by setting the key return mode register (KRM) and inputting a falling edge to the key interrupt input pins (KR0 to KR7).

**Table 22-1. Assignment of Key Interrupt Detection Pins**

| Flag | Description |
|---|---|
| KRM0 | Controls KR0 signal in 1-bit units. |
| KRM1 | Controls KR1 signal in 1-bit units. |
| KRM2 | Controls KR2 signal in 1-bit units. |
| KRM3 | Controls KR3 signal in 1-bit units. |
| KRM4 | Controls KR4 signal in 1-bit units. |
| KRM5 | Controls KR5 signal in 1-bit units. |
| KRM6 | Controls KR6 signal in 1-bit units. |
| KRM7 | Controls KR7 signal in 1-bit units. |

**Caution The pin assignment differs depending on the products. The PIOR50 bit can specify which I/O port is assigned to each KRn function. Inputs to the A/D converter are multiplexed with P80 to P87 and P90 to P92, to which the function can be assigned. These pins are used as analog input pins in their initial state. Use the PIOR50 bit and the ADPC register to make the pins operate as digital input pins before using the key interrupt function. For details of the PIOR50 bit and the ADPC register, refer to 4.3.14 Peripheral I/O redirection register 5 (PIOR5) and 12.3.11 A/D port configuration register (ADPC).**

**Remarks 1.** n = 0 to 7

**2.** The available number of interrupts depends on the setting of the PIOR50 bit as shown below.

- When PIOR50 is set to 0
  KR0 to KR3: 48-pin products
  KR0 to KR7: 64-pin, 80-pin, 100-pin products
- When PIOR50 is set to 1
  KR0, KR1: 20-pin products
  KR0 to KR7: 30-pin, 48-pin, 64-pin products
  KR0 to KR5: 32-pin products

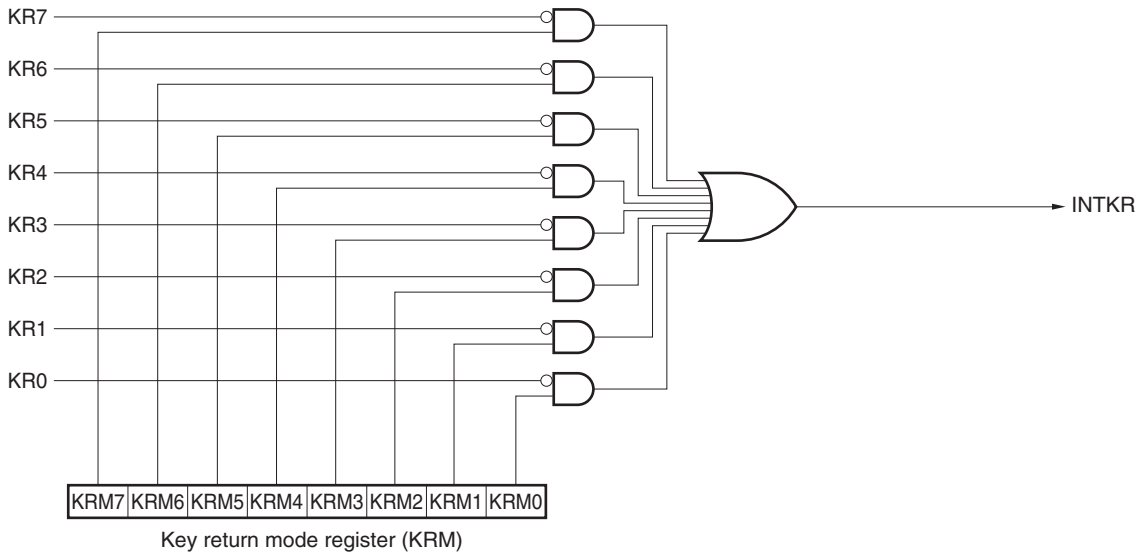
## 22.2 Configuration of Key Interrupt

Table 22-2 shows the configuration of the key interrupt. Figure 22-1 is the block diagram of the key interrupt.

**Table 22-2. Configuration of Key Interrupt**

| Item | Configuration |
|---|---|
| Control register | Key return mode register (KRM) |

**Figure 22-1. Block Diagram of Key Interrupt**



## 22.3 Register Controlling Key Interrupt

### 22.3.1 Key return mode register (KRM)

The KRM0 to KRM7 bits control signals KR0 to KR7.

The KRM register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 22-2. Format of Key Return Mode Register (KRM)**

Address: FFF37H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| KRM | KRM7 | KRM6 | KRM5 | KRM4 | KRM3 | KRM2 | KRM1 | KRM0 |

| KRMn | Key interrupt mode control |
|---|---|
| 0 | Does not detect key interrupt signal |
| 1 | Detects key interrupt signal |

**Cautions 1. An interrupt will be generated if the target bit of the KRM register is set to 1 while the KRn pin is at low level. To ignore this interrupt, set the KRM register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag after waiting for the key interrupt input low-level width ($t_{KR}$).**

**2. The pins not used in the key interrupt can be used as normal ports.**

**3. When the assignment of the key interrupt input pin is changed by using the PIOR50 bit, an interrupt may be generated. The pin assignment must be changed while the KRM register is 00H or while the key input interrupt is prohibited.**

**4. Set the bits of the KRM register to 0 for pins to which the key interrupt function is not to be allocated.**

**Remark** n = 0 to 7

# CHAPTER 23 STANDBY FUNCTION

## 23.1 Standby Function and Configuration

### 23.1.1 Standby function

The standby function reduces the operating current of the system, and the following three modes are available.

**(1) HALT mode**

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock, high-speed on-chip oscillator, subsystem clock, or low-speed on-chip oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

**(2) STOP mode**

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock and high-speed on-chip oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

The output from a port pin can be inverted in response to the generation of a source condition for release from the STOP mode.

**(3) SNOOZE mode**

In response to a request for A/D conversion by a timer trigger signal (INTRTC), ELE event input **Note 1**, or signal generated by the A/D converter trigger select register 0 (ADTRGS0) **Note 2**, a signal for data reception by the LIN/UART module (RLIN3) in the UART mode, or a DTC activation signal, this LSI is released from the STOP mode and A/D conversion, data reception, or DTC operation proceed without the CPU operating. This can only be specified when the high-speed on-chip oscillator is selected for the CPU/peripheral hardware clock ($f_{CLK}$).

There is a function for the output of a signal to indicate whether the LSI is in the SNOOZE mode or not on a specified pin when this LSI enters and is released from the SNOOZE mode.

In any mode, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

**Notes** **1.** Only in the RL78/F14.
**2.** Only in the RL78/F13.

(**Cautions** are listed on the next page.)

**Cautions 1. The STOP mode can be used only when the CPU is operating on the main system clock. The STOP mode cannot be set while the CPU operates with the PLL clock or the subsystem/low-speed on-chip oscillator select clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem/low-speed on-chip oscillator select clock.**

**2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction (except for the setting unit of SNOOZE mode).**

**3. When using the A/D converter and the LIN/UART module in the SNOOZE mode, set up the A/D converter mode register 2 (ADM2) and the UART standby control register (LUSCn) before switching to the STOP mode. For details, see 17.2 Register Descriptions.**

**4. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.**

**5. It can be selected by the option byte whether the WDT-dedicated low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 29 OPTION BYTE.**

## 23.2 Registers controlling standby function

Oscillation stabilization time when this LSI is released from the STOP mode is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Output from the port when a source condition for release from the STOP mode is generated is inverted by the following register.

- STOP status output control register (STPSTC)

**Remark** For the registers that start, stop, or select the clock, see **CHAPTER 5 CLOCK GENERATOR**.

### 23.2.1 Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.

The X1 clock oscillation stabilization time can be checked in the following case.

- If the X1 clock starts oscillation while the high-speed on-chip oscillator clock or sub/low-speed on-chip oscillator select clock is being used as the CPU clock.
- If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.

When reset is released (reset by $\overline{\text{RESET}}$ input, POR, LVD, WDT, and executing an illegal instruction), the STOP instruction and MSTOP bit (bit 7 of clock operation status control register (CSC)) = 1 clear this register to 00H.

**Figure 23-1. Format of Oscillation Stabilization Time Counter Status Register (OSTC)**

Address: FFFA2H After reset: 00H R

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSTC | MOST8 | MOST9 | MOST10 | MOST11 | MOST13 | MOST15 | MOST17 | MOST18 |

| MOST 8 | MOST 9 | MOST 10 | MOST 11 | MOST 13 | MOST 15 | MOST 17 | MOST 18 | Oscillation stabilization time status | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | $f_X$ = 10 MHz | $f_X$ = 20 MHz |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $2^8/f_X$ max. | 25.6 μs max. | 12.8 μs max. |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $2^8/f_X$ min. | 25.6 μs min. | 12.8 μs min. |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | $2^9/f_X$ min. | 51.2 μs min. | 25.6 μs min. |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | $2^{10}/f_X$ min. | 102.4 μs min. | 51.2 μs min. |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | $2^{11}/f_X$ min. | 204.8 μs min. | 102.4 μs min. |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | $2^{13}/f_X$ min. | 819.2 μs min. | 409.6 μs min. |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | $2^{15}/f_X$ min. | 3.27 ms min. | 1.63 ms min. |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | $2^{17}/f_X$ min. | 13.10 ms min. | 6.55 ms min. |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | $2^{18}/f_X$ min. | 26.21 ms min. | 13.10 ms min. |

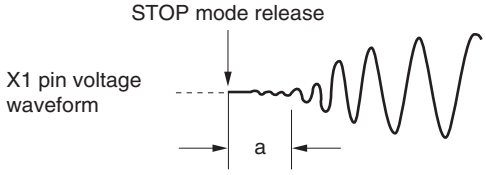
**Cautions 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.**

**2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS). In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register.**

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.**

**(Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)**

**3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).**




**Remark** $f_X$: X1 clock oscillation frequency

### 23.2.2 Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time when the STOP mode is released.

When the X1 clock is made to oscillate, the operation automatically waits for the time set using the OSTS register.

After the X1 clock starts oscillating, confirm with the oscillation stabilization time counter status register (OSTC) that the desired oscillation stabilization time has elapsed. The oscillation stabilization time can be checked up to the time set using the OSTC register.

The OSTS register can be set by an 8-bit memory manipulation instruction.

Writing to the OSTS register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation sets this register to 07H.

**Figure 23-2. Format of Oscillation Stabilization Time Select Register (OSTS)**

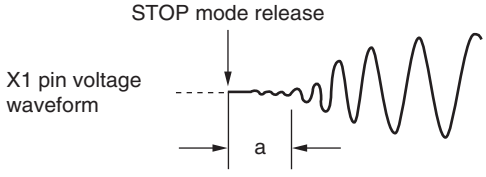
Address: FFFA3H After reset: 07H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| OSTS | 0 | 0 | 0 | 0 | 0 | OSTS2 | OSTS1 | OSTS0 |

| OSTS2 | OSTS1 | OSTS0 | Oscillation stabilization time selection | | |
|---|---|---|---|---|---|
| | | | | $f_X$ = 10 MHz | $f_X$ = 20 MHz |
| 0 | 0 | 0 | $2^8/f_X$ | 25.6 μs | 12.8 μs |
| 0 | 0 | 1 | $2^9/f_X$ | 51.2 μs | 25.6 μs |
| 0 | 1 | 0 | $2^{10}/f_X$ | 102.4 μs | 51.2 μs |
| 0 | 1 | 1 | $2^{11}/f_X$ | 204.8 μs | 102.4 μs |
| 1 | 0 | 0 | $2^{13}/f_X$ | 819.2 μs | 409.6 μs |
| 1 | 0 | 1 | $2^{15}/f_X$ | 3.27 ms | 1.63 ms |
| 1 | 1 | 0 | $2^{17}/f_X$ | 13.10 ms | 6.55 ms |
| 1 | 1 | 1 | $2^{18}/f_X$ | 26.21 ms | 13.10 ms |

1. **To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTS register before executing the STOP instruction.**
2. **Before changing the setting of the OSTS register, confirm that the count operation of the OSTC register is completed.**
3. **Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.**
4. **The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS).**
   **In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register.**
   - **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem clock is being used as the CPU clock.**
   - **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.**

   **(Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)**
5. **The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).**




**Remark** $f_X$: X1 clock oscillation frequency

### 23.2.3 STOP status output control register (STPSTC)

The port latch of P31 or P52 can be inverted in response to a source condition for release from the STOP mode being generated or a transition from SNOOZE mode to normal mode.

Set the STPSTC register by a 1-bit or 8-bit memory manipulation instruction.

Writing to the STPSTC register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation clears this register to 00H.

**Cautions 1. As the 20-, 30-, and 32-pin products do not have the STOPST function, they do not have the STPSTC register.**

**2. When the STOP status output control register is to be used, the target port pin should be placed in the output mode and the port latch should be set to 0 beforehand.**

**Figure 23-3. Format of STOP Status Output Control Register (STPSTC)**

Address: F02CAH After reset: 00H R/W

| Symbol | <7> | 6 | 5 | <4> | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| STPSTC | STPOEN | 0 | 0 | STPLV Note 1 | 0 | 0 | 0 | STPSEL Note 2 |

| STPOEN | Enabling or disabling of STOPST output |
|---|---|
| 0 | Nothing is done when this LSI is released from the STOP mode. |
| 1 | The STPLV value is output on the pin selected by STPSEL when this LSI is released from the STOP mode. |

| STPLV Note 1 | Control of STOPST output level |
|---|---|
| 0 | Output low |
| 1 | Output high |

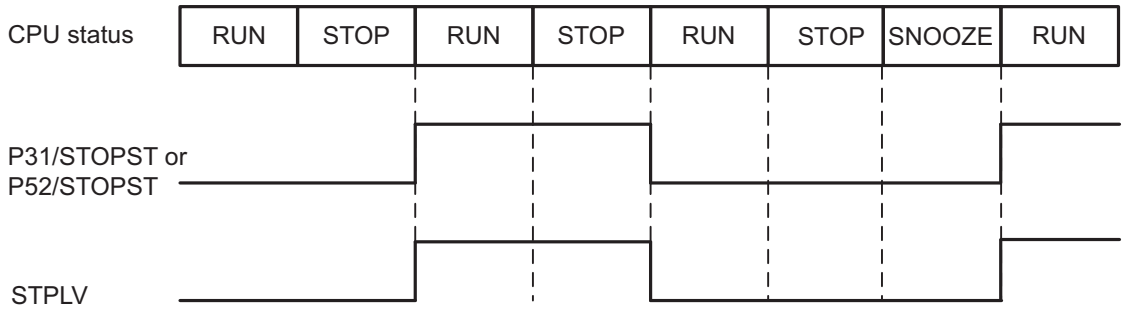
| STPSEL Note 2 | Control of STOPST pin selection |
|---|---|
| 0 | Selects P31 |
| 1 | Selects P52 |

**Notes 1.** The STPLV bit is inverted when this LSI is released from the STOP mode and when this LSI makes a transition from SNOOZE mode to normal mode.

**2.** Bit 0 is a read-only reserved bit in 48-pin products. When setting the register, write the initial value, 0, to this bit.

**Caution Be sure to set bits 1 to 3, 5, and 6 of the STPSTC register to 0.**

The following figure shows the timing of the STOPST pin and STPLV bit during CPU operation status.



## 23.3 Standby Function Operation

### 23.3.1 HALT mode

**(1) HALT mode**

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock, high-speed on-chip oscillator clock, PLL clock, low-speed on-chip oscillator clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

**Caution If the interrupt mask flag is 0 (interrupt servicing enabled) and the interrupt request flag is 1 (interrupt request signal is generated), the HALT mode is released even if the HALT instruction is executed (Because the interrupt request signal is used to release the HALT mode).**

**Table 23-1. Operating Statuses in HALT Mode (1/2)**

| HALT Mode Setting / Item | | | When HALT Instruction Is Executed While CPU Is Operating on Main System Clock | | | |
|---|---|---|---|---|---|---|
| | | | When CPU Is Operating on High-speed On-chip Oscillator Clock ($f_{IH}$) | When CPU Is Operating on X1 Clock ($f_X$) | When CPU Is Operating on External Main System Clock ($f_{EX}$) | When CPU Is Operating on PLL Clock ($f_{PLL}$) |
| System clock | | | Clock supply to the CPU is stopped | | | |
| | Main system clock | $f_{IH}$ | Operation continues (cannot be stopped) | Operation disabled | | Only operation of the PLL clock continues (and cannot be stopped). Clocks other than the PLL clock do not operate. |
| | | $f_X$ | Operation disabled | Operation continues (cannot be stopped) | Cannot operate | |
| | | $f_{EX}$ | | Cannot operate | Operation continues (cannot be stopped) | |
| | | $f_{PLL}$ | | Operation disabled | Operation disabled | Operation continues (cannot be stopped) |
| | Subsystem clock | $f_{XT}$ | Status before HALT mode was set is retained | | | |
| | | $f_{EXS}$ | | | | |
| | $f_{IL}$ | | Set by bit 1 (HPIEN) of on-chip debug option byte (000C3H/020C3H), bit 0 (SELLOSC) of the CKSEL register, and bit 4 (WUTMMCK0) of the OSMC register.<br>• WUTMMCK0 = 1: Oscillates<br>• WUTMMCK0 = 0 and SELLOSC = 1: Oscillates<br>• WUTMMCK0 = 0, SELLOSC = 0, and HPIEN = 1: Oscillates<br>• WUTMMCK0 = 0, SELLOSC = 0, and HPIEN = 0: Stops | | | |
| $f_{WDT}$ | | | Set by bits 0 (WDSTBYON) and 4 (WDTON) of user option byte (000C0H/020C0H)<br>• WDTON = 0: Stops<br>• WDTON = 1 and WDSTBYON = 1: Oscillates<br>• WDTON = 1 and WDSTBYON = 0: Stops | | | |
| CPU | | | Operation stopped | | | |
| Code flash memory | | | | | | |
| Data flash memory | | | | | | |
| RAM | | | Operation stopped (operation can continue during DTC transfer) | | | |
| Port (latch) | | | Status before HALT mode was set is retained | | | |
| Timer array unit | | | Operable | | | |
| Real-time clock (RTC) | | | | | | |
| Watchdog timer | | | See **CHAPTER 11 WATCHDOG TIMER** | | | |
| Clock monitor | | | Operable ($f_{IL}$ operates) | | | |
| Timer RJ | | | Operable | | | |
| Timer RD | | | | | | |
| Clock output/buzzer output | | | | | | |
| A/D converter | | | | | | |
| D/A converter | | | | | | |
| Comparator | | | | | | |
| Serial array unit (SAU) | | | | | | |
| Serial interface (IICA) | | | | | | |
| DTC | | | | | | |
| ELC | | | Linking between operational function blocks is possible. | | | |
| LIN/UART module (RLIN3) | | | Operable | | | |
| CAN interface (RS-CAN lite) | | | | | | |
| Power-on-reset function | | | | | | |
| Voltage detection function | | | | | | |
| External interrupt | | | | | | |

| HALT Mode Setting / Item | | When HALT Instruction Is Executed While CPU Is Operating on Main System Clock | | | |
|---|---|---|---|---|---|
| | | When CPU Is Operating on High-speed On-chip Oscillator Clock ($f_{IH}$) | When CPU Is Operating on X1 Clock ($f_X$) | When CPU Is Operating on External Main System Clock ($f_{EX}$) | When CPU Is Operating on PLL Clock ($f_{PLL}$) |
| Key interrupt function | | Operable | | | |
| CRC operation function | High-speed CRC | | | | |
| | General-purpose CRC | Operation stopped (operation can continue during DTC transfer) | | | |
| Illegal-memory access detection function | | | | | |
| RAM2 bit error detection function | | | | | |
| RAM guard function | | | | | |
| SFR guard function | | | | | |
| CPU stack pointer monitor function | | Operation stopped (operation can continue during vectored interrupt servicing) | | | |

**Remark** Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
Cannot operate: Operation is not possible regardless of switching to the HALT mode.
$f_{IH}$: High-speed on-chip oscillator clock $f_{IL}$: Low-speed on-chip oscillator clock
$f_X$: X1 clock $f_{EX}$: External main system clock
$f_{XT}$: XT1 clock $f_{EXS}$: External subsystem clock
$f_{PLL}$: PLL clock
$f_{WDT}$: WDT-dedicated low-speed on-chip oscillator clock

**Table 23-1. Operating Statuses in HALT Mode (2/2)**

| HALT Mode Setting / Item | | | When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock | | When HALT Instruction Is Executed While CPU Is Operating on Low-speed On-chip Oscillator Clock ($f_{IL}$) |
|---|---|---|---|---|---|
| | | | When CPU Is Operating on XT1 Clock ($f_{XT}$) | When CPU Is Operating on External Subsystem Clock ($f_{EXS}$) | |
| System clock | | | Clock supply to the CPU is stopped | | |
| | Main system clock | $f_{IH}$ | Operation disabled | | |
| | | $f_X$ | | | |
| | | $f_{EX}$ | | | |
| | | $f_{PLL}$ | | | |
| | Subsystem clock | $f_{XT}$ | Operation continues (cannot be stopped) | Cannot operate | Cannot operate |
| | | $f_{EXS}$ | Cannot operate | Operation continues (cannot be stopped) | |
| | $f_{IL}$ | | Set by bit 1 (HPIEN) of on-chip debug option byte (000C3H/020C3H), bit 0 (SELLOSC) of the CKSEL register, and bit 4 (WUTMMCK0) of the OSMC register.<br>• WUTMMCK0 = 1: Oscillates<br>• WUTMMCK0 = 0 and SELLOSC = 1: Oscillates<br>• WUTMMCK0 = 0, SELLOSC = 0, and HPIEN = 1: Oscillates<br>• WUTMMCK0 = 0, SELLOSC = 0, and HPIEN = 0: Stops | | |
| $f_{WDT}$ | | | Set by bits 0 (WDSTBYON) and 4 (WDTON) of user option byte (000C0H/020C0H)<br>• WDTON = 0: Stops<br>• WDTON = 1 and WDSTBYON = 1: Oscillates<br>• WDTON = 1 and WDSTBYON = 0: Stops | | |
| CPU | | | Operation stopped | | |
| Code flash memory | | | | | |
| Data flash memory | | | | | |
| RAM | | | Operation stopped (operation can continue during DTC transfer) | | |
| Port (latch) | | | Status before HALT mode was set is retained | | |
| Timer array unit | | | Operable (Operation is disabled while in the low consumption RTC mode) | | |
| Real-time clock (RTC) | | | Operable | | |
| Watchdog timer | | | See **CHAPTER 11 WATCHDOG TIMER** | | |
| Clock monitor | | | Operation stopped | | |
| Timer RJ | | | Operable (Operation is disabled while in the low consumption RTC mode) | | |
| Timer RD | | | | | |
| Clock output/buzzer output | | | | | |
| A/D converter | | | Operation disabled | | |
| D/A converter | | | | | |
| Comparator | | | | | |
| Serial array unit (SAU) | | | Operable (Operation is disabled while in the low consumption RTC mode) | | |
| Serial interface (IICA) | | | Operation disabled | | |
| DTC | | | Operable | | |
| ELC | | | Linking between operational function blocks is possible. | | |
| LIN/UART module (RLIN3) | | | Operation disabled | | |
| CAN interface (RS-CAN lite) | | | | | |
| Power-on-reset function | | | Operable | | |
| Voltage detection function | | | | | |
| External interrupt | | | | | |
| Key interrupt function | | | | | |

| Item \ HALT Mode Setting | | When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock | | When HALT Instruction Is Executed While CPU Is Operating on Low-speed On-chip Oscillator Clock ($f_{IL}$) |
|---|---|---|---|---|
| | | When CPU Is Operating on XT1 Clock ($f_{XT}$) | When CPU Is Operating on External Subsystem Clock ($f_{EXS}$) | |
| CRC operation function | High-speed CRC | Operation disabled | | |
| | General-purpose CRC | Operation stopped (operation can continue during DTC transfer) | | |
| Illegal-memory access detection function | | | | |
| RAM2 bit error detection function | | | | |
| RAM guard function | | | | |
| SFR guard function | | | | |
| CPU stack pointer monitor function | | Operation stopped (operation can continue during vectored interrupt servicing) | | |

**Remark** Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
Cannot operate: Operation is not possible regardless of switching to the HALT mode.
$f_{IH}$: High-speed on-chip oscillator clock
$f_{IL}$: Low-speed on-chip oscillator clock
$f_X$: X1 clock
$f_{EX}$: External main system clock
$f_{XT}$: XT1 clock
$f_{EXS}$: External subsystem clock
$f_{PLL}$: PLL clock
$f_{WDT}$: WDT-dedicated low-speed on-chip oscillator clock

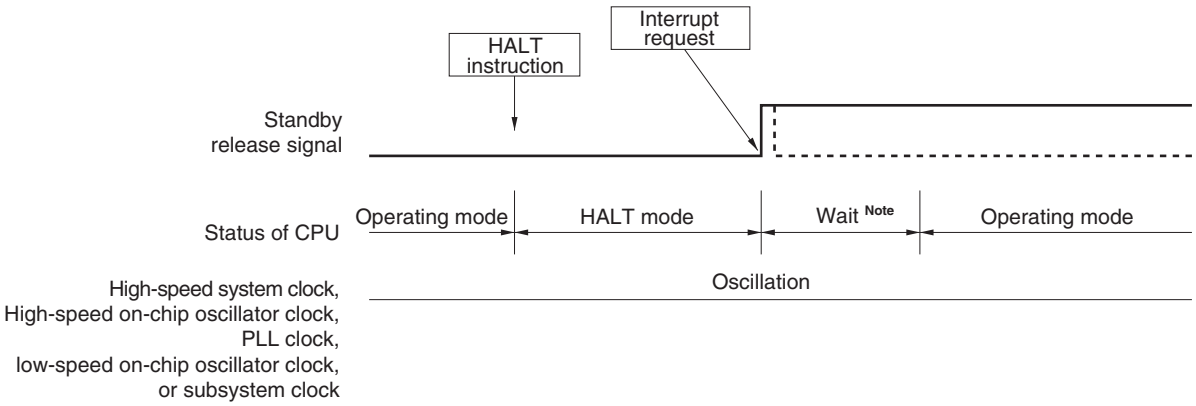
### (2) HALT mode release

The HALT mode can be released by interrupt and reset signal generation.

#### (a) Release by unmasked interrupt request

When an interrupt request with an interrupt mask flag set to 0 (interrupt servicing enabled) is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction of the HALT instruction is executed.

**Figure 23-4. HALT Mode Release by Interrupt Request Generation**




**Note** Wait time for HALT mode release

- When vectored interrupt servicing is carried out
  - Main/PLL select clock: 15 to 16 clocks
  - Subsystem/low-speed on-chip oscillator select clock (RTCLPC = 0): 10 to 11 clocks
  - Subsystem/low-speed on-chip oscillator select clock (RTCLPC = 1): 11 to 12 clocks
- When vectored interrupt servicing is not carried out
  - Main/PLL select clock: 9 to 10 clocks
  - Subsystem/low-speed on-chip oscillator select clock (RTCLPC = 0): 4 to 5 clocks
  - Subsystem/low-speed on-chip oscillator select clock (RTCLPC = 1): 5 to 6 clocks

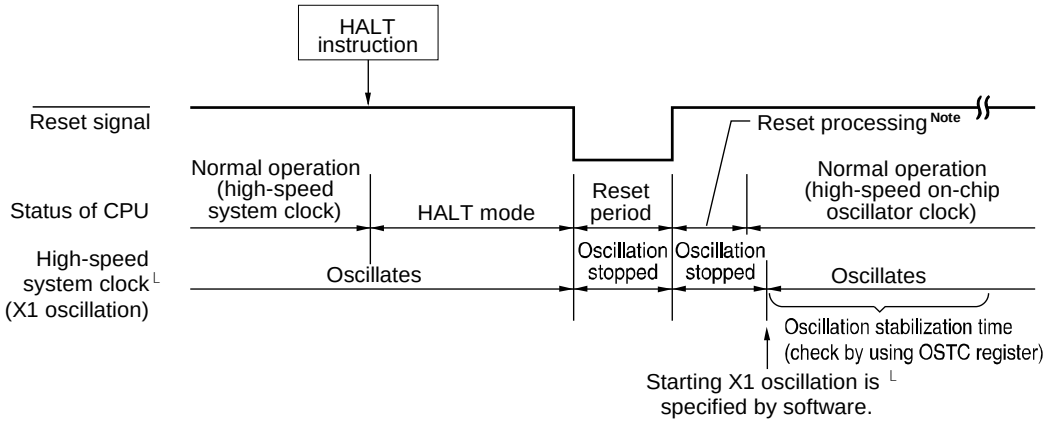
**Remark** The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

**(b) Release by reset signal generation**

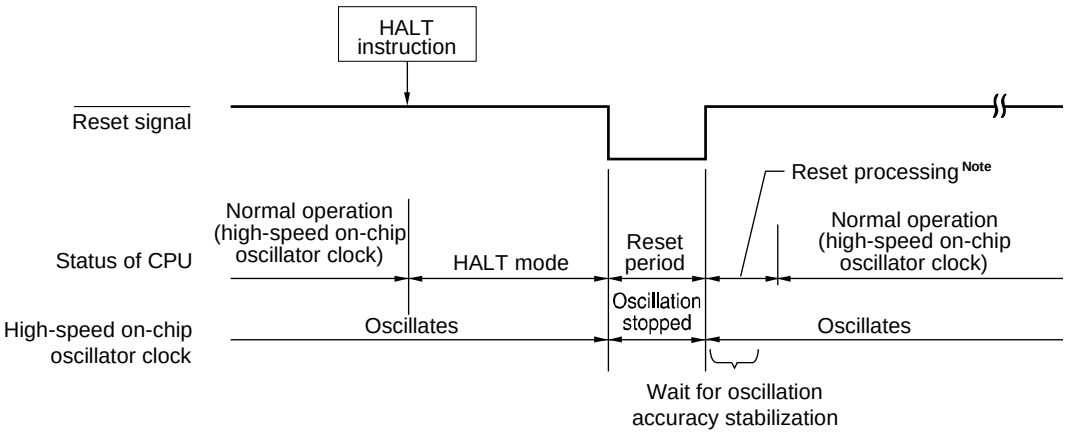
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program branches.

**Figure 23-5. HALT Mode Release by Reset**

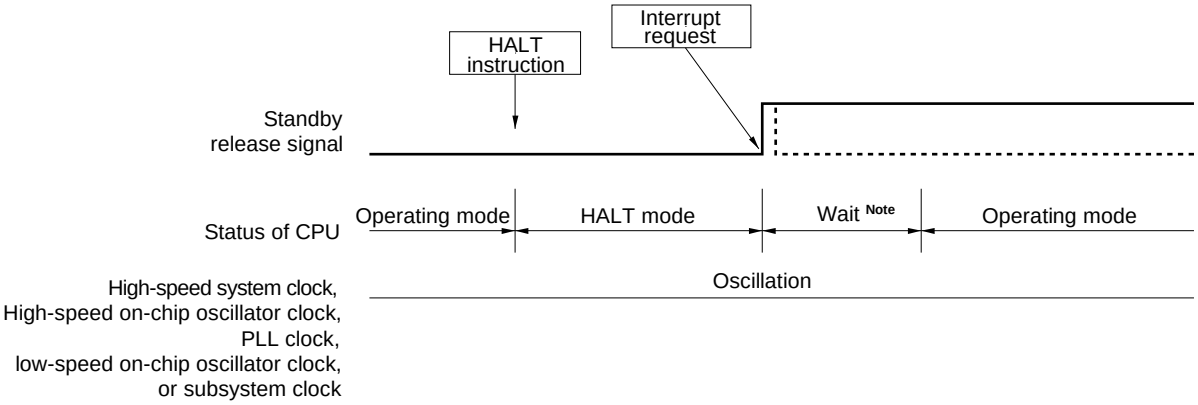
**(1) When high-speed system clock is used as CPU clock**




**(2) When high-speed on-chip oscillator clock is used as CPU clock**




**(3) When subsystem clock or low-speed on-chip oscillator clock is used as CPU clock**




**Note** For the reset processing time, see **CHAPTER 25 POWER-ON-RESET CIRCUIT**.

### 23.3.2 STOP mode

**(1) STOP mode setting and operating statuses**

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the high-speed on-chip oscillator clock, X1 clock, or external main system clock.

**Caution Because the interrupt request signal is used to clear the STOP mode, if the interrupt mask flag is 0 (interrupt servicing enabled) and the interrupt request flag is 1 (interrupt request signal is generated), the STOP mode is immediately cleared when the STOP instruction is executed. Accordingly, once the STOP instruction is executed, the system returns to its normal operating mode after the elapse of release time from the STOP mode.**

The operating statuses in the STOP mode are shown below.

**Table 23-2. Operating Statuses in STOP Mode**

| Item \ STOP Mode Setting | | When STOP Instruction Is Executed While CPU Is Operating on Main System Clock | | | |
|---|---|---|---|---|---|
| | | When CPU Is Operating on High-speed On-chip Oscillator clock ($f_{IH}$) | When CPU Is Operating on X1 Clock ($f_X$) | When CPU Is Operating on External Main System Clock ($f_{EX}$) | When CPU Is Operating on PLL Clock ($f_{PLL}$) |
| System clock | | Clock supply to the CPU is stopped | | | |
| Main system clock | $f_{IH}$ | Stopped | | | |
| | $f_X$ | | | | |
| | $f_{EX}$ | | | | |
| | $f_{PLL}$ | Operation disabled | | | |
| Subsystem clock | $f_{XT}$ | Status before STOP mode was set is retained | | | |
| | $f_{EXS}$ | | | | |
| $f_{IL}$ | | Set by bit 0 (SELLOSC) of the CKSEL register and bit 4 (WUTMMCK0) of the OSMC register.<br>• WUTMMCK0 = 1: Oscillates<br>• WUTMMCK0 = 0 and SELLOSC = 1: Oscillates<br>• WUTMMCK0 = 0 and SELLOSC = 0: Stops | | | |
| $f_{WDT}$ | | Set by bits 0 (WDSTBYON) and 4 (WDTON) of user option byte (000C0H/020C0H)<br>• WDTON = 0: Stops<br>• WDTON = 1 and WDSTBYON = 1: Oscillates<br>• WDTON = 1 and WDSTBYON = 0: Stops | | | |
| CPU | | Operation stopped | | | |
| Code flash memory | | | | | |
| Data flash memory | | Operation stopped (the STOP instruction is not executed during data flash programming) | | | |
| RAM | | Operation stopped | | | |
| Port (latch) | | Status before STOP mode was set is retained | | | |
| Timer array unit | | Operation disabled | | | |
| Real-time clock (RTC) | | Operable (when the subsystem clock is selected as an input clock ($f_{RTC}$)) | | | |
| Watchdog timer | | See **CHAPTER 11 WATCHDOG TIMER** | | | |
| Clock monitor | | Operation stopped | | | |
| Timer RJ | | Operable<br>• In the event counting mode when no TRJIO0 input filters are selected.<br>• If the subsystem/low-speed on-chip oscillator select clock is selected as the clock source for counting and the RTCLPC bit of the OSMC register is 0.<br>• If the low-speed on-chip oscillator is selected as the clock source for counting. | | | |
| Timer RD | | Operable (can only operate for output of the SNOOZE status signal when the subsystem/low-speed on-chip oscillator select clock is selected) | | | |
| Clock output/buzzer output | | Operable only when the subsystem/low-speed on-chip oscillator select clock is selected as the count clock | | | |
| A/D converter | | Wakeup operation is enabled (switching to the SNOOZE mode) | | | |
| D/A converter | | Operable (the state before the STOP mode was set is retained) | | | |
| Comparator | | Operable (if the settings allow release from the STOP mode and the digital filters are not in use) | | | |
| Serial array unit (SAU) | | Operation disabled | | | |
| Serial interface (IICA) | | Wakeup operation by address match is enabled | | | |
| DTC | | Reception of trigger signals from sources for DTC activation is enabled (switching to the SNOOZE mode) | | | |
| ELC | | Linking between operational function blocks is possible. | | | |
| LIN/UART module (RLIN3) | | Only wakeup operation of the UART is possible (switching to the SNOOZE mode). | | | |
| CAN interface (RS-CAN lite) | | Operation disabled | | | |
| Power-on-reset function | | Operable | | | |
| Voltage detection function | | | | | |
| External interrupt | | | | | |
| Key interrupt function | | | | | |

| Item \ STOP Mode Setting | | When STOP Instruction Is Executed While CPU Is Operating on Main System Clock | | | |
|---|---|---|---|---|---|
| | | When CPU Is Operating on High-speed On-chip Oscillator clock ($f_{IH}$) | When CPU Is Operating on X1 Clock ($f_X$) | When CPU Is Operating on External Main System Clock ($f_{EX}$) | When CPU Is Operating on PLL Clock ($f_{PLL}$) |
| CRC operation function | High-speed CRC | Operation stopped | | | |
| | General-purpose CRC | | | | |
| Illegal-memory access detection function | | | | | |
| RAM2 bit error detection function | | | | | |
| RAM guard function | | | | | |
| SFR guard function | | | | | |
| CPU stack pointer monitor function | | Operation stopped (operation can continue during vectored interrupt servicing) | | | |

**Remark** Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.

$f_{IH}$: High-speed on-chip oscillator clock
$f_{IL}$: Low-speed on-chip oscillator clock
$f_X$: X1 clock
$f_{EX}$: External main system clock
$f_{XT}$: XT1 clock
$f_{EXS}$: External subsystem clock
$f_{PLL}$: PLL clock
$f_{WDT}$: WDT-dedicated low-speed on-chip oscillator clock

**Cautions 1. To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.**

**2. To stop the watchdog timer clock in the STOP mode, set bit 0 (WDSTBYON) of a user option byte (000C0H/020C0H) to 0 (stop the watchdog timer operation in the HALT/STOP/SNOOZE mode).**

**3. To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the high-speed on-chip oscillator clock before the execution of the STOP instruction. Before changing the CPU clock from the high-speed on-chip oscillator clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).**

### (2) STOP mode release

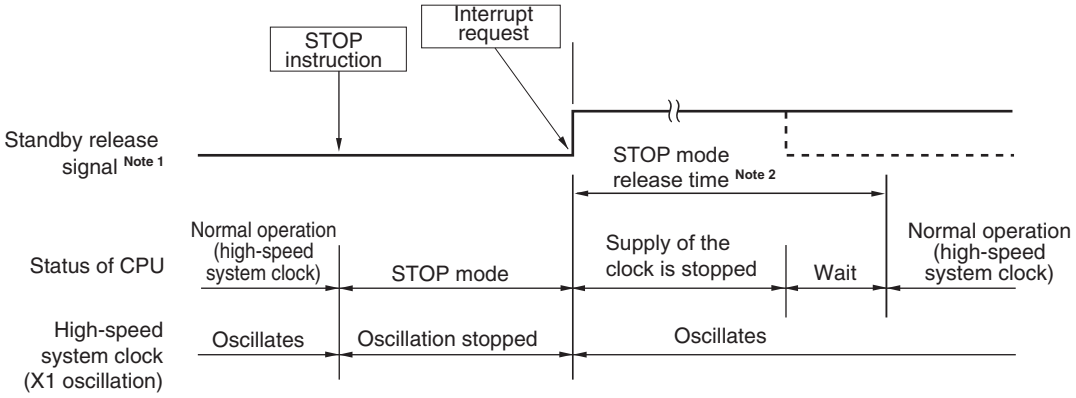
The STOP mode can be released by interrupt and reset signal generation.

#### (a) Release by unmasked interrupt request

When an interrupt request with an interrupt mask set to 0 (Interrupt servicing enabled) is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction of the STOP instruction is executed.

**Figure 23-6. STOP Mode Release by Interrupt Request Generation (1/2)**

**(1) When high-speed system clock (X1 oscillation) is used as CPU clock**




**Notes**

**1.** For details of the standby release signal, see **Figure 21-1. Basic Configuration of Interrupt Function**.

**2.** STOP mode release time

Supply of the clock is stopped

- When FRQSEL4 = 1 in the user option byte (000C2H/020C2H):
  18 μs to "whichever is longer 105 μs and the oscillation stabilization time (set by STS)"
- When FRQSEL4 = 0 in the user option byte (000C2H/020C2H):
  18 μs to "whichever is longer 65 μs and the oscillation stabilization time (set by OSTS)"

Wait

- When vectored interrupt servicing is carried out: 10 to 11 clocks
- When vectored interrupt servicing is not carried out: 4 to 5 clocks

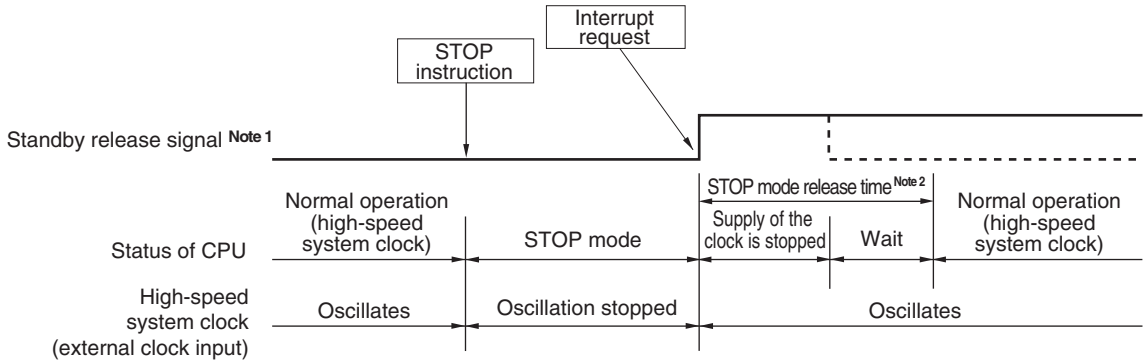
**Remarks**

**1.** The clock supply stop time varies depending on the temperature conditions and STOP mode period.

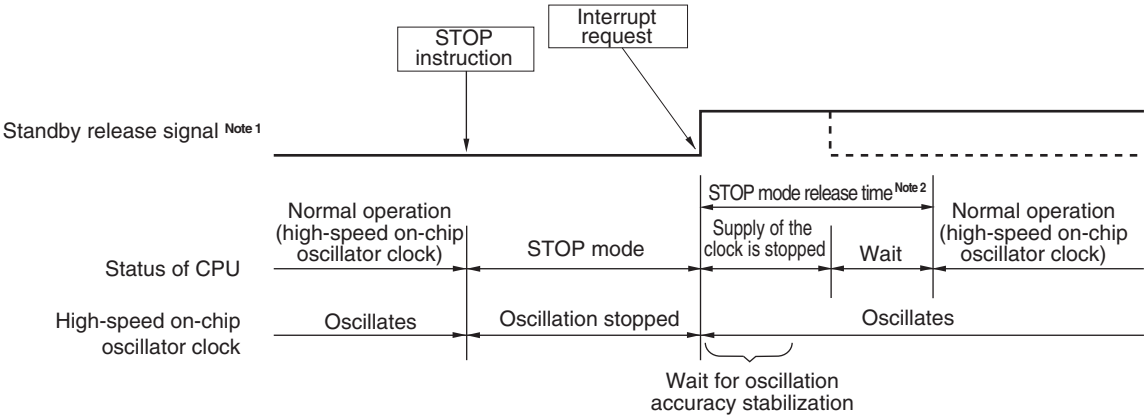
**2.** The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

**Figure 23-6. STOP Mode Release by Interrupt Request Generation (2/2)**

**(2) When high-speed system clock (external clock input) is used as CPU clock**




**(3) When high-speed on-chip oscillator clock is used as CPU clock**




**Notes**

**1.** For details of the standby release signal, see **Figure 21-1. Basic Configuration of Interrupt Function**.

**2.** STOP mode release time

Supply of the clock is stopped

- When FRQSEL4 = 1 in the user option byte (000C2H/020C2H):
  18 μs to 105 μs
- When FRQSEL4 = 0 in the user option byte (000C2H/020C2H):
  18 μs to 65 μs

Wait

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

**Remarks**

**1.** The clock supply stop time varies depending on the temperature conditions and STOP mode period.

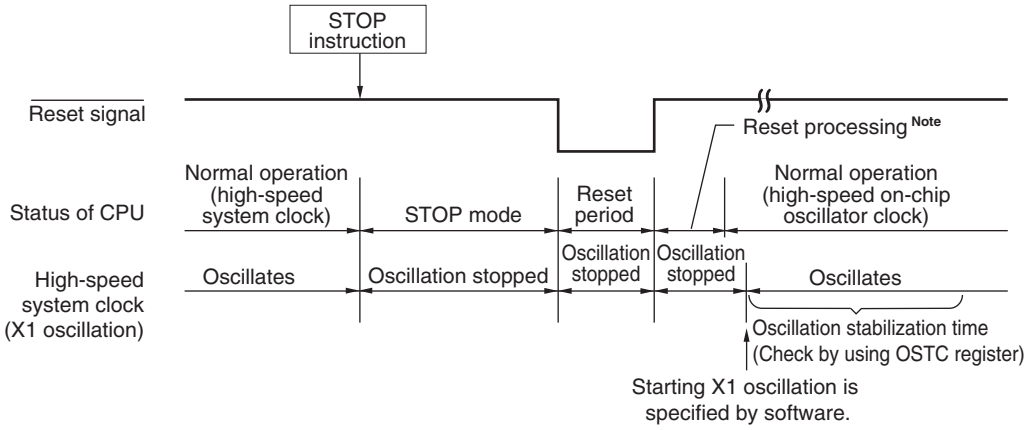
**2.** The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

**(b) Release by reset signal generation**

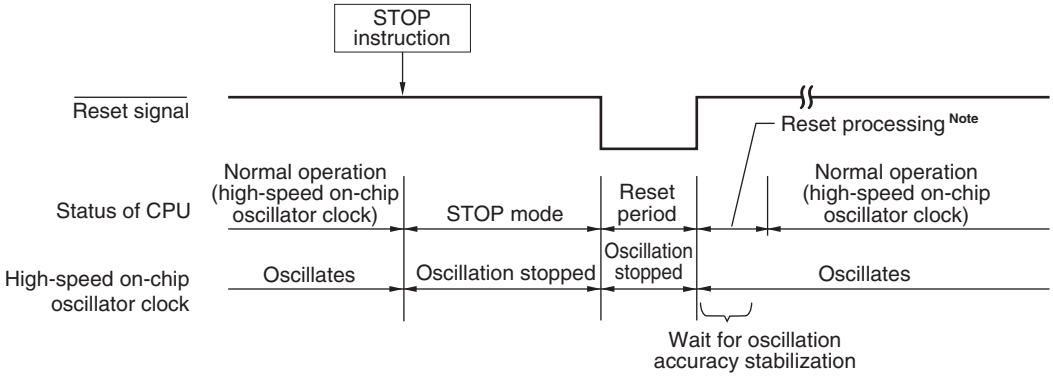
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

**Figure 23-7. STOP Mode Release by Reset**

**(1) When high-speed system clock is used as CPU clock**




**(2) When high-speed on-chip oscillator clock is used as CPU clock**




**Note** For the reset processing time, see **CHAPTER 25 POWER-ON-RESET CIRCUIT**.

### 23.3.3 SNOOZE mode

**(1) SNOOZE mode setting and operating statuses**

The SNOOZE mode can only be specified for the A/D converter, LIN/UART module, or DTC. Note that this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock.

When using the A/D converter in the SNOOZE mode, set up A/D converter mode register 2 (ADM2) before entering the STOP mode. For details, see **12.3 Registers Used in A/D Converter**.

When using the UART function of the LIN/UART module in the SNOOZE mode, set up the LUSCn register before entering the STOP mode.

When DTC transfer is used in SNOOZE mode, before entering the STOP mode, allow DTC activation by interrupt to be used. During STOP mode, detecting DTC activation by interrupt enables DTC transit to SNOOZE mode, automatically. For details, see **19.2 Registers**.

In SNOOZE mode transition, wait status to be only following time.

Transition time from STOP mode to SNOOZE mode

- When FRQSEL4 = 1 in the user option byte (000C2H/010C2H): 18 μs to 105 μs
- When FRQSEL4 = 0 in the user option byte (000C2H/010C2H): 18 μs to 65 μs

**Remark** Transition time from STOP mode to SNOOZE mode varies depending on the temperature conditions and the STOP mode period.

Transition time from SNOOZE mode to normal operation:

- When vectored interrupt servicing is carried out: "4.99 μs to 9.44 μs" + 7 clocks
- When vectored interrupt servicing is not carried out: "4.99 μs to 9.44 μs" + 1 clock

The operating statuses in the SNOOZE mode are shown below.

**Table 23-3. Operating Statuses in SNOOZE Mode**

| Item \ STOP Mode Setting | | During a period in STOP mode, input of a timer trigger signal for the A/D converter, reception of a data signal for the LIN/UART module in the UART mode, or the generation of an interrupt signal for DTC activation |
|---|---|---|
| | | When CPU Is Operating on High-speed On-chip Oscillator Clock ($f_{IH}$) |
| System clock | | Clock supply to the CPU is stopped |
| Main system clock | $f_{IH}$ | Operation started |
| | $f_X$ | Stopped |
| | $f_{EX}$ | |
| | $f_{PLL}$ | Operation disabled |
| Subsystem clock | $f_{XT}$ | Use of the status while in the STOP mode continues |
| | $f_{EXS}$ | |
| $f_{IL}$ | | Set by bit 0 (SELLOSC) of the CKSEL register and bit 4 (WUTMMCK0) of the OSMC register.<br>• WUTMMCK0 = 1: Oscillates<br>• WUTMMCK0 = 0 and SELLOSC = 1: Oscillates<br>• WUTMMCK0 = 0 and SELLOSC = 0: Stops |
| $f_{WDT}$ | | Set by bits 0 (WDSTBYON) and 4 (WDTON) of user option byte (000C0H/020C0H)<br>• WDTON = 0: Stops<br>• WDTON = 1 and WDSTBYON = 1: Oscillates<br>• WDTON = 1 and WDSTBYON = 0: Stops |
| CPU | | Operation stopped |
| Code flash memory | | |
| Data flash memory | | |
| RAM | | |
| Port (latch) | | Use of the status while in the STOP mode continues |
| Timer array unit | | Operation disabled |
| Real-time clock (RTC) | | Operable (when the subsystem clock is selected as an input clock ($f_{RTC}$)) |
| Watchdog timer | | See **CHAPTER 11 WATCHDOG TIMER** |
| Clock monitor | | Operation stopped |
| Timer RJ | | Operable<br>• In the event counting mode when no TRJIO0 input filters are selected.<br>• If the subsystem/low-speed on-chip oscillator select clock is selected as the clock source for counting and the RTCLPC bit of the OSMC register is 0.<br>• If the low-speed on-chip oscillator is selected as the clock source for counting. |
| Timer RD | | Operable (can only operate for output of the SNOOZE status signal when the subsystem/low-speed on-chip oscillator select clock is selected) |
| Clock output/buzzer output | | Operable only when the subsystem/low-speed on-chip oscillator select clock is selected as the count clock |
| A/D converter | | Operable |
| D/A converter | | Operable (the state before the STOP mode was set is retained) |
| Comparator | | Operable (if the settings allow release from the STOP mode and the digital filters are not in use) |
| Serial array unit (SAU) | | Operation disabled |
| Serial interface (IICA) | | |
| DTC | | Operable |
| ELC | | Linking between operation function blocks is possible. |
| LIN/UART module (RLIN3) | | Operable (only in the UART mode) |
| CAN interface (RS-CAN lite) | | Operation disabled |

| Item \ STOP Mode Setting | | During a period in STOP mode, input of a timer trigger signal for the A/D converter, reception of a data signal for the LIN/UART module in the UART mode, or the generation of an interrupt signal for DTC activation |
|---|---|---|
| | | When CPU Is Operating on High-speed On-chip Oscillator Clock ($f_{IH}$) |
| Power-on-reset function | | Operable |
| Voltage detection function | | |
| External interrupt | | |
| Key interrupt function | | |
| CRC operation function | High-speed CRC | Operation stopped |
| | General-purpose CRC | |
| Illegal-memory access detection function | | |
| RAM2 bit error detection function | | |
| RAM guard function | | |
| SFR guard function | | |
| CPU stack pointer monitor function | | Operation stopped (operation can continue during vectored interrupt servicing) |

**Remark** Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.

$f_{IH}$: High-speed on-chip oscillator clock
$f_{IL}$: Low-speed on-chip oscillator clock
$f_X$: X1 clock
$f_{EX}$: External main system clock
$f_{XT}$: XT1 clock
$f_{EXS}$: External subsystem clock
$f_{PLL}$: PLL clock
$f_{WDT}$: WDT-dedicated low-speed on-chip oscillator clock

**(2) SNOOZE mode status output (SNOOZEST)**

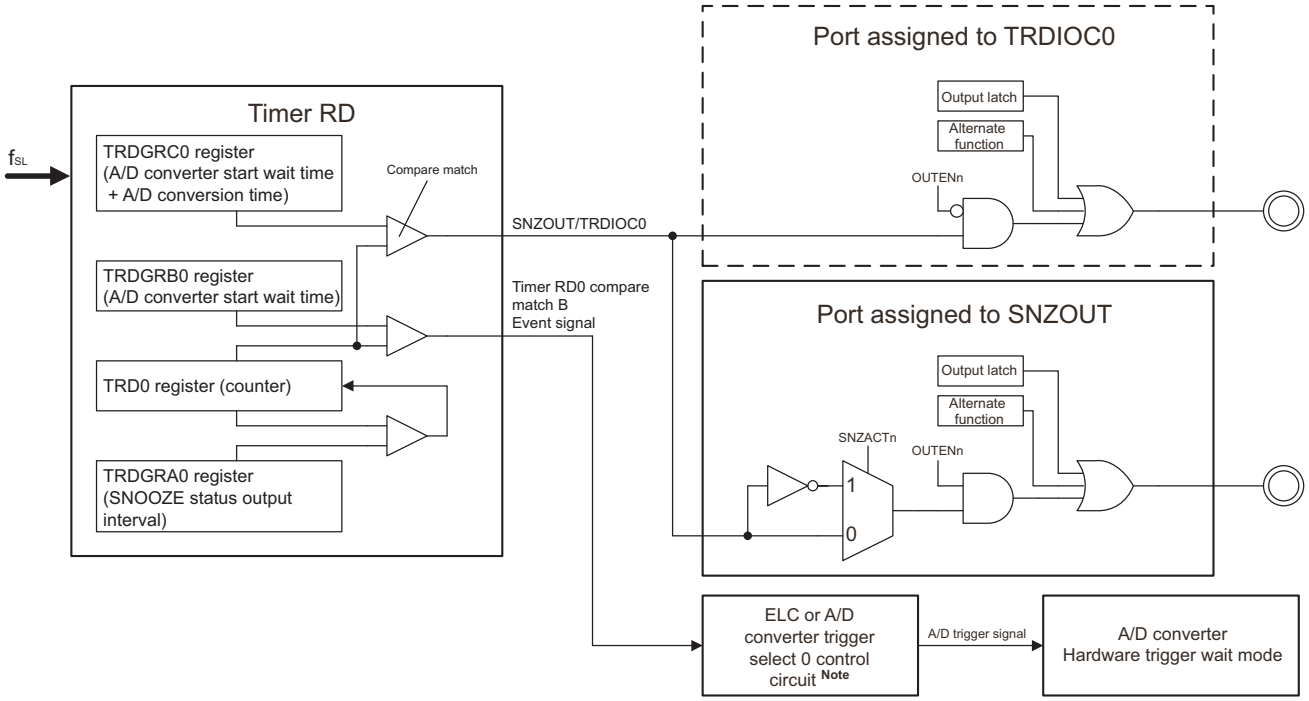
This function is used to output the status of the SNOOZE mode (in any mode other than the SNOOZE mode or in the SNOOZE mode) on the specified pin.

This function is executed by $f_{SL}$ (sub/low-speed on-chip oscillator select clock), timer RD0, event link controller (ELC) or A/D converter trigger select 0 control circuit **Note**, A/D converter, and ports simultaneously. $f_{SL}$ is used as the count source for the timer RD0. By using the PWM function of the timer mode, the timer RD0 sets a period in the TRDGRA0 register and generates a compare match signal in the TRDGRB0 and TRDGRC0 registers. The compare match signal of the TRDGRB0 register is used as the operation trigger of the A/D converter via the ELC or A/D converter trigger select 0 control circuit **Note**. After receiving this operation trigger, the A/D converter performs A/D conversion in SNOOZE mode. After the compare match signal of the TRDGRC0 register is received, SNOOZE status is output from the SNZOUTn (n = 0 to 7) pin selected by the PSNZCNT0 to PSNZCNT3 registers.

**Note** The RL78/F14 uses the ELC and the RL78/F13 uses the A/D converter trigger select 0 control circuit.

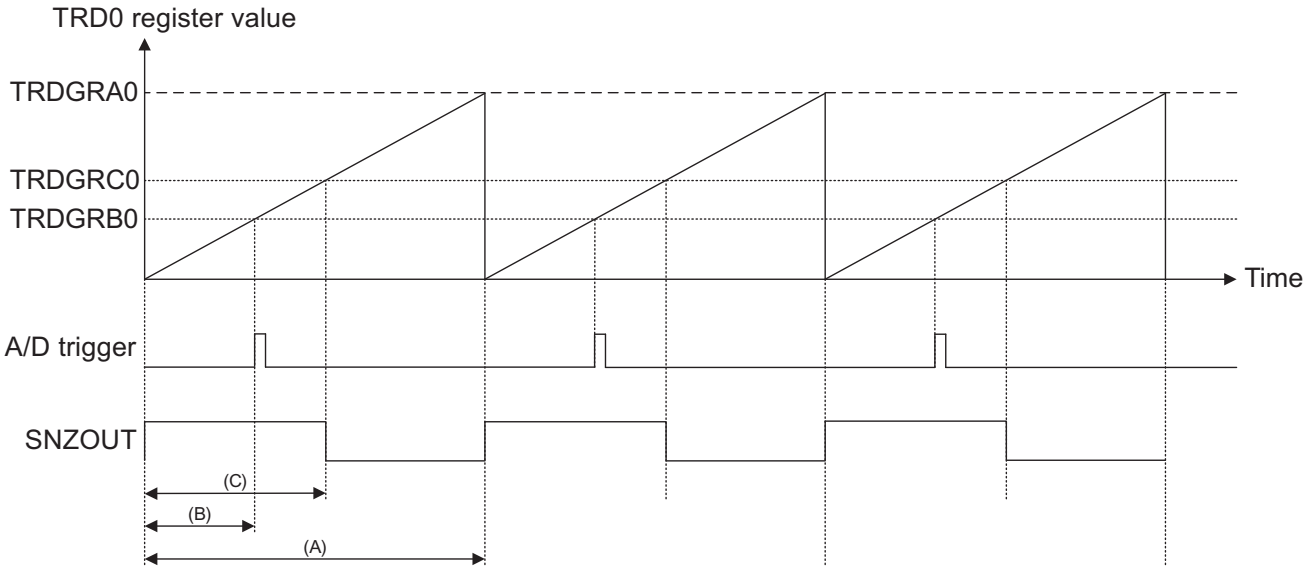
Figure 23-8 shows the configuration of the circuit for SNOOZE mode status output. Figure 23-9 shows the timing of SNOOZE mode status output.

**Figure 23-8. Configuration of Circuit for SNOOZE Mode Status Output**



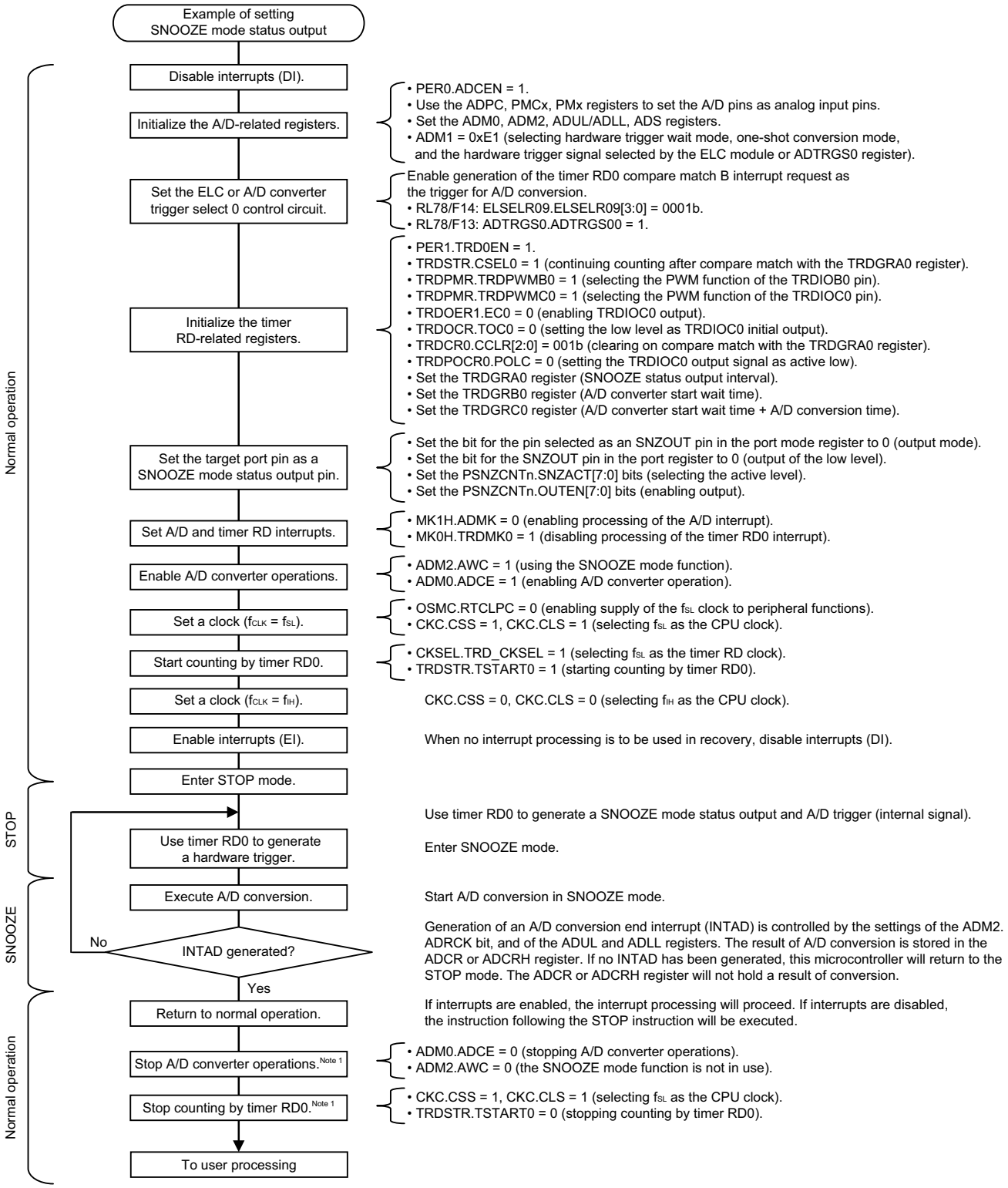

**Note** The RL78/F14 uses the ELC and the RL78/F13 uses the A/D converter trigger select 0 control circuit.

**Figure 23-9. Timing of SNOOZE Mode Status Output**




(A): TRDGRA0 = SNOOZE status output interval
(B): TRDGRB0 = A/D converter start wait time
(C): TRDGRC0 = A/D converter start wait time + A/D conversion time

**Figure 23-10. Example of Settings for SNOOZE Mode Status Output**




**Note 1.** On the transition from STOP (or SNOOZE) mode to normal operation, stop A/D converter operations and then stop counting by timer RD0 (after changing the CPU clock to $f_{SL}$) as shown in the figure above.

**Caution For details on the settings of registers for the A/D converter, timer RD, port functions, etc., see the corresponding chapters.**

# CHAPTER 24 RESET FUNCTION

The following seven operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-reset (POR) circuit
(4) Internal reset by comparison of supply voltage of the voltage detector (LVD) and detection voltage
(5) Internal reset by execution of illegal instruction **Note**
(6) Internal reset in response to the clock monitor detecting that oscillation of the main clock has stopped
(7) Internal reset by illegal-memory access

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.

When a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer detects an overflow, a voltage is detected on the POR and LVD circuits, an illegal instruction is executed **Note**, the clock monitor detects that oscillation of the main clock has stopped, or memory is accessed illegally, the device is reset and the hardware is set to the status shown in **Table 24-1**.

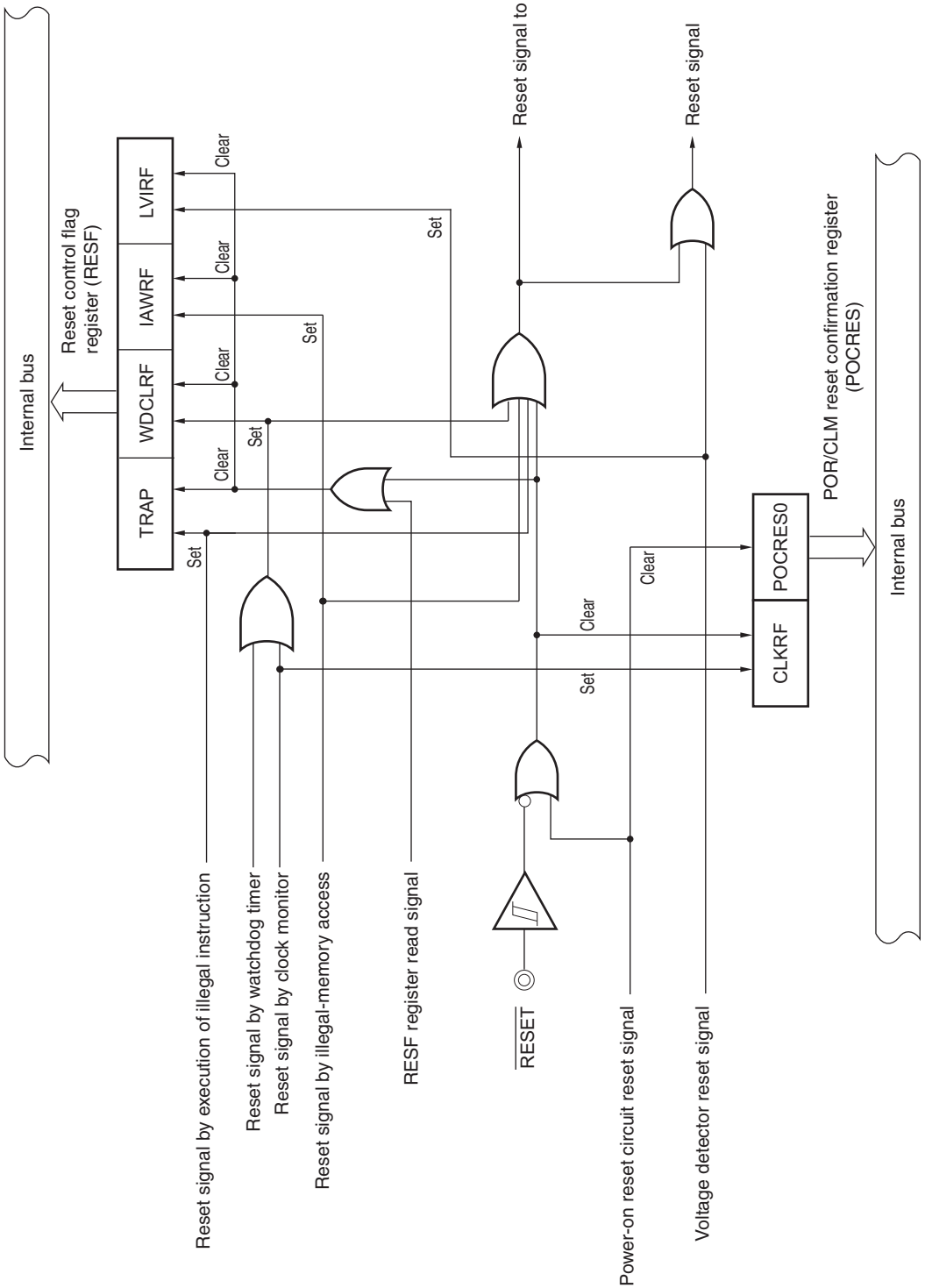
When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the high-speed on-chip oscillator clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the high-speed on-chip oscillator clock (see **Figures 24-2** to **24-4**) after reset processing. Reset by POR and LVD circuit supply voltage detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD}$ is detected after the reset, and program execution starts using the high-speed on-chip oscillator clock (see **CHAPTER 25 POWER-ON-RESET CIRCUIT** and **CHAPTER 26 VOLTAGE DETECTOR**) after reset processing.

**Note** The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

**Cautions 1. For an external reset, input the low level to the $\overline{\text{RESET}}$ pin for at least 10 μs.**
**When an external reset is applied while the power supply voltage is rising, the period over which the voltage is below the range of operating voltage ($V_{DD}$ < 2.7V) is not included in the 10 μs. However, continuing the input of a low level before release from the power-on reset state does not create a problem.**
**2. During reset input, the X1 clock, XT1 clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating. External main system clock input and external subsystem clock input become invalid.**
**3. When reset is effected, port pin P130 is set to low-level output and other port pins become high-impedance, because each SFR and 2nd SFR are initialized.**

**Remark** $V_{POR}$: POR power supply rise detection voltage

**Figure 24-1. Block Diagram of Reset Function**




**Caution An LVD circuit internal reset does not reset the LVD circuit.**

**Remarks 1.** LVIM: Voltage detection register

**2.** LVIS: Voltage detection level register

**Figure 24-2. Timing of Reset by $\overline{\text{RESET}}$ Input**

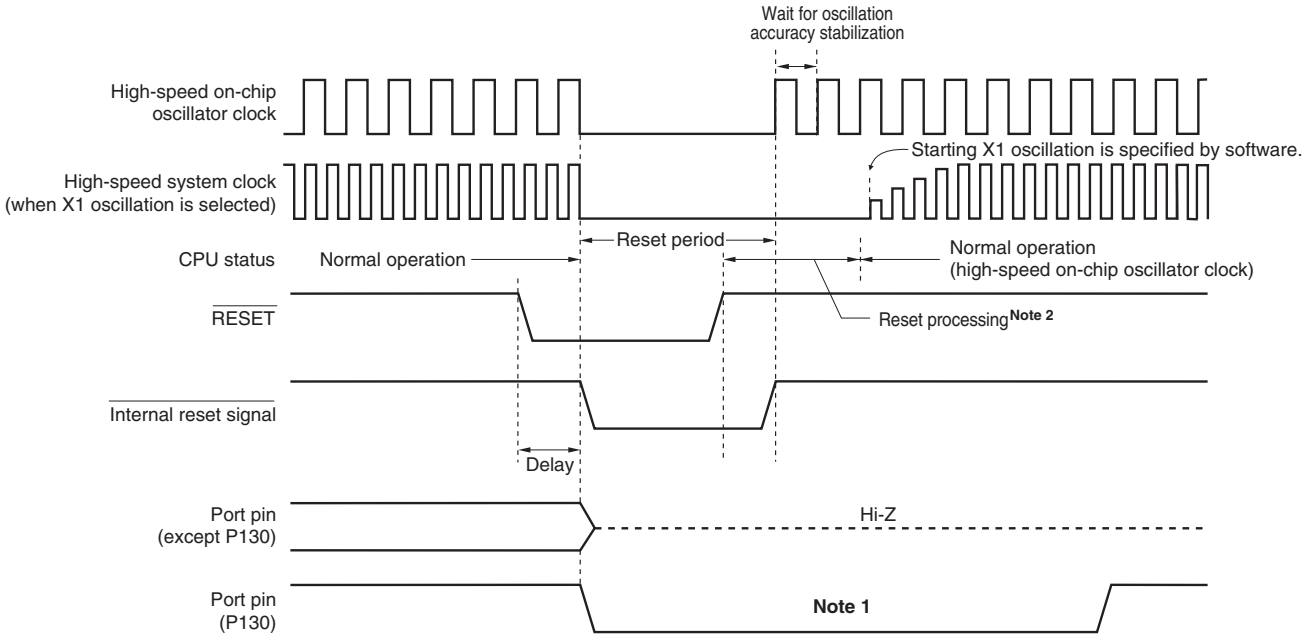
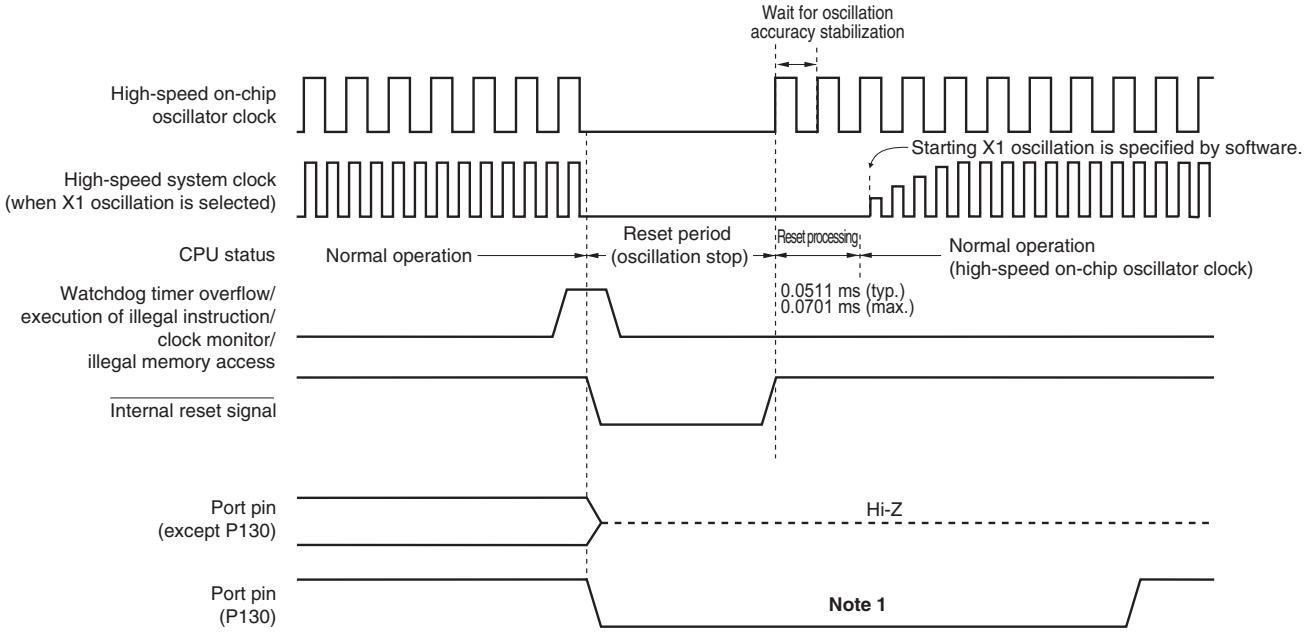
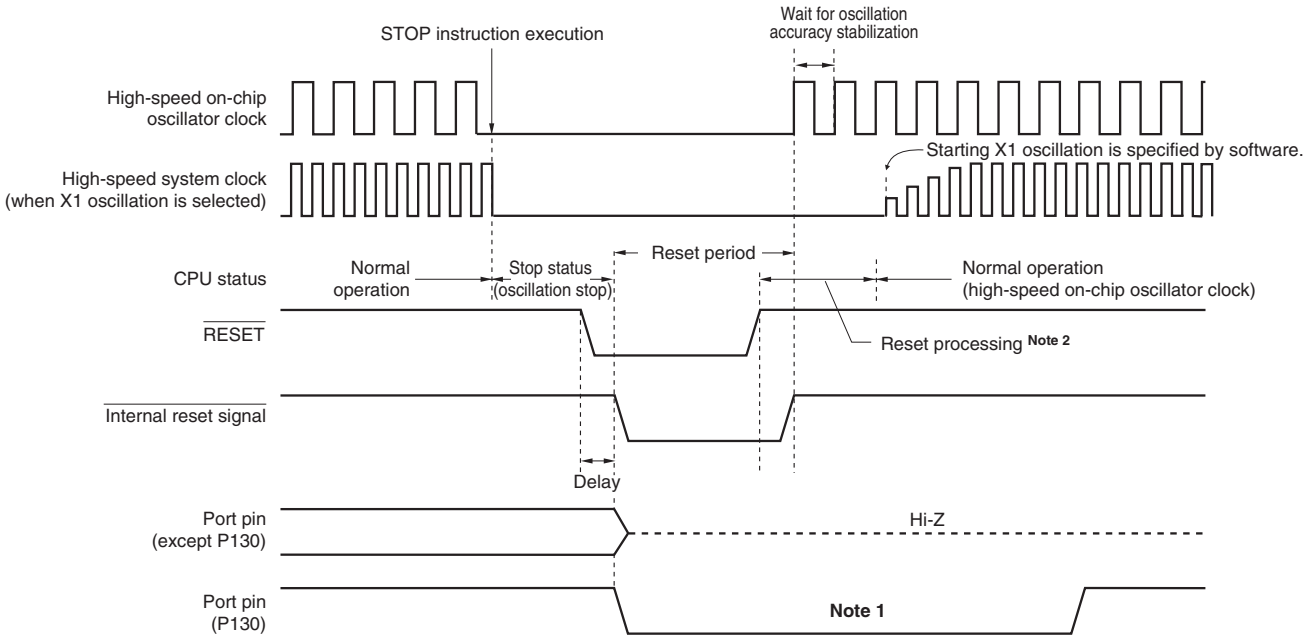



**Figure 24-3. Timing of Reset by Watchdog Timer Overflow, Execution of Illegal Instruction, Clock Monitor, or Illegal-Memory Access**




(**Notes**, **Caution**, and **Remark** are listed on the next page.)

**Figure 24-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input**




**Notes 1.** When P130 is set to high-level output before reset is effected, the output signal of P130 can be dummy-output as a reset signal to an external device, because P130 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 to high-level output by software.

**2.** Reset processing time when the external reset is released is shown below.

After the first release of POR: 0.672 ms (typ.), 0.832 ms (max.) (when the LVD is in use)
0.399 ms (typ.), 0.519 ms (max.) (when the LVD is off)

After the second release of POR: 0.531 ms (typ.), 0.675 ms (max.) (when the LVD is in use)
0.259 ms (typ.), 0.362 ms (max.) (when the LVD is off)

After power is supplied, a voltage stabilization waiting time of about 0.99 ms (typ.) and up to 2.30 ms (max.) is required before reset processing starts after release of the external reset.

**Remark** For the reset timing of the power-on-reset circuit and voltage detector, see **CHAPTER 25 POWER-ON-RESET CIRCUIT** and **CHAPTER 26 VOLTAGE DETECTOR**.

**Table 24-1. Operation Statuses During Reset Period**

| Item | | | During Reset Period |
|---|---|---|---|
| System clock | | | Clock supply to the CPU is stopped. |
| | Main system clock | $f_{IH}$ | Operation stopped |
| | | $f_X$ | Operation stopped (the X1 and X2 pins are input port mode) |
| | | $f_{EX}$ | Clock input invalid (the pin is input port mode) |
| | Subsystem clock | $f_{XT}$ | Operation stopped (the XT1 and XT2 pins are input port mode) |
| | | $f_{EXS}$ | Clock input invalid (the pin is input port mode) |
| | $f_{IL}$ | | Operation stopped |
| | $f_{PLL}$ | | |
| | $f_{WDT}$ | | |
| CPU | | | Operation stopped |
| Code flash memory | | | |
| Data flash memory | | | |
| RAM | | | |
| Port (latch) | P130 | | Outputs the low level |
| | P40 | | High impedance (by an external reset or POR reset)<br>Pulled up (by reset other than external reset or POR reset) |
| | Other than P130 and P40 | | High impedance |
| Timer array unit | | | Operation stopped |
| Timer RJ | | | |
| Timer RD | | | |
| Real-time clock (RTC) | | | |
| Watchdog timer | | | |
| Clock monitor | | | |
| Clock output/buzzer output | | | |
| A/D converter | | | |
| D/A converter **Note** | | | |
| Comparator **Note** | | | |
| Serial array unit (SAU) | | | |
| Serial interface (IICA) | | | |
| LIN/UART module (RLIN3) | | | |
| CAN interface (RS-CAN lite) | | | |
| Multiplier and Divider/Multiply-Accumulator | | | |
| DTC | | | |
| ELC **Note** | | | |
| Power-on-reset function | | | Detection operation possible |
| Low-voltage detection function | | | Operation stopped |
| External interrupt | | | |
| Key interrupt function | | | |
| CRC operation function | High-speed CRC | | |
| | General-purpose CRC | | |
| Illegal-memory access detection function | | | |
| RAM guard function | | | |
| SFR guard function | | | |

**Note** RL78/F14 only.

**Remark** $f_{IH}$: High-speed on-chip oscillator clock
$f_{EX}$: External main system clock
$f_{EXS}$: External subsystem clock
$f_{PLL}$: PLL clock
$f_X$: X1 oscillation clock
$f_{XT}$: XT1 oscillation clock
$f_{IL}$: Low-speed on-chip oscillator clock
$f_{WDT}$: WDT-dedicated low-speed on-chip oscillator clock

**Table 24-2. States of Hardware After Acceptance of a Reset (1/4)**

| Hardware | | After Acceptance of a Reset Note 1 |
|---|---|---|
| Program counter (PC) | | The contents of the reset vector table (0000H, 0001H) are set. |
| Stack pointer (SP) | | Undefined |
| Program status word (PSW) | | 06H |
| RAM | Data memory | Undefined |
| | General-purpose registers | Undefined |
| Port mode select register (PMS) | | 00H |
| Noise filter enable registers 0 to 2 (NFEN0 to NFEN2) | | 00H |
| Peripheral enable registers 0 to 2 (PER0 to PER2) | | 00H |
| High-speed on-chip oscillator frequency select register (HOCODIV) | | Undefined |
| High-speed on-chip oscillator trimming register (HIOTRM) | | **Note 2** |
| Operation speed mode control register (OSMC) | | 00H |
| Port register 0 (P0) | | 00H |
| Port register 1 (P1) | | 00H |
| Port register 3 (P3) | | 00H |
| Port register 4 (P4) | | 00H |
| Port register 5 (P5) | | 00H |
| Port register 6 (P6) | | 00H |
| Port register 7 (P7) | | 00H |
| Port register 8 (P8) | | 00H |
| Port register 9 (P9) | | 00H |
| Port register 10 (P10) | | 00H |
| Port register 12 (P12) | | Undefined |
| Port register 13 (P13) | | Undefined |
| Port register 14 (P14) | | 00H |
| Port register 15 (P15) | | 00H |
| Serial data register 00 (SDR00) | | 0000H |
| Serial data register 01 (SDR01) | | 0000H |
| Timer data register 00 (TDR00) | | 0000H |
| Timer data register 01 (TDR01/L/H) | | 00H |
| 10-bit A/D conversion result register (ADCR) | | 0000H |
| 8-bit A/D conversion result register (ADCRH) | | 00H |
| Port mode register 0 (PM0) | | FFH |
| Port mode register 1 (PM1) | | FFH |
| Port mode register 3 (PM3) | | FFH |
| Port mode register 4 (PM4) | | FFH |
| Port mode register 5 (PM5) | | FFH |
| Port mode register 6 (PM6) | | FFH |
| Port mode register 7 (PM7) | | FFH |
| Port mode register 8 (PM8) | | FFH |
| Port mode register 9 (PM9) | | FFH |
| Port mode register 10 (PM10) | | FFH |
| Port mode register 12 (PM12) | | FFH |
| Port mode register 14 (PM14) | | FFH |
| Port mode register 15 (PM15) | | FFH |
| A/D converter mode register 0 (ADM0) | | 00H |
| Analog input channel specification register (ADS) | | 00H |
| A/D converter mode register 1 (ADM1) | | 00H |
| D/A conversion value setting register 0 (DACS0) | | 00H |
| D/A converter mode register (DAM) | | 00H |
| Key return mode register (KRM) | | 00H |
| External interrupt rising edge enable register 0 (EGP0) | | 00H |

(**Notes** and **Remark** are given below Table 24-2, States of Hardware After Acceptance of a Reset (4/4).)

**Table 24-2. States of Hardware After Acceptance of a Reset (2/4)**

| Hardware | After Acceptance of a Reset[Note 1] |
|---|---|
| External interrupt falling edge enable register 0 (EGN0) | 00H |
| External interrupt rising edge enable register 1 (EGP1) | 00H |
| External interrupt falling edge enable register 1 (EGN0) | 00H |
| Serial data register 10 (SDR10) | 0000H |
| Serial data register 11 (SDR11) | 0000H |
| IICA shift register 0 (IICA0) | 00H |
| IICA status register 0 (IICS0) | 00H |
| IICA flag register 0 (IICF0) | 00H |
| 16-bit watch error correction register (SUBCUDW) | 0000H |
| Timer RD general register C0 (TRDGRC0) | FFFFH[Note 3] |
| Timer RD general register D0 (TRDGRD0) | FFFFH[Note 3] |
| Timer RD general register C1 (TRDGRC1) | FFFFH[Note 3] |
| Timer RD general register D1 (TRDGRD1) | FFFFH[Note 3] |
| Timer data register 02 (TDR02) | 0000H |
| Timer data register 03 (TDR03/L/H) | 00H |
| Timer data register 04 (TDR04) | 0000H |
| Timer data register 05 (TDR05) | 0000H |
| Timer data register 06 (TDR06) | 0000H |
| Timer data register 07 (TDR07) | 0000H |
| Timer data register 10 (TDR10) | 0000H |
| Timer data register 11 (TDR11/L/H) | 00H |
| Timer data register 12 (TDR12) | 0000H |
| Timer data register 13 (TDR13/L/H) | 00H |
| Timer data register 14 (TDR14) | 0000H |
| Timer data register 16 (TDR16) | 0000H |
| Timer data register 17 (TDR17) | 0000H |
| Second count register (SEC) | 00H |
| Minute count register (MIN) | 00H |
| Hour count register (HOUR) | 12H[Note 4] |
| Week count register (WEEK) | 00H |
| Day count register (DAY) | 01H |
| Month count register (MONTH) | 01H |
| Year count register (YEAR) | 00H |
| Watch error correction register (SUBCUD) | 00H |
| Alarm minute register (ALARMWM) | 00H |
| Alarm hour register (ALARMWH) | 12H |
| Alarm week register ALARMWW) | 00H |
| Real-time clock control register 0 (RTCC0) | 00H |
| Real-time clock control register 1 (RTCC1) | 00H |
| Clock operation mode control register (CMC) | 00H |
| Clock operation status control register (CSC) | C0H |
| Oscillation stabilization time counter status register (OSTC) | 00H |
| Oscillation stabilization time select register (OSTS) | 07H |
| System clock control register (CKC) | 00H |
| Clock output select register 0 (CKS0) | 00H |
| Reset control flag register (RESF) | Undefined[Note 5] |
| Voltage detection register (LVIM) | 00H[Note 5] |
| Voltage detection level register (LVIS) | 00H/01H/81H[Note 6] |
| Watchdog timer enable register (WDTE) | 1AH/9AH[Note 7] |
| CRC input register (CRCIN) | 00H |
| Interrupt request flag register 2 (IF2/L/H) | 00H |

(**Notes** and **Remark** are given below Table 24-2, States of Hardware After Acceptance of a Reset (4/4).)

**Table 24-2. States of Hardware After Acceptance of a Reset (3/4)**

| Hardware | After Acceptance of a Reset[Note 1] |
|---|---|
| Interrupt request flag register 3L (IF3L) | 00H |
| Interrupt mask flag register 2 (MK2/L/H) | FFH |
| Interrupt mask flag register 3 (MK3L) | FFH |
| Priority specification flag register 02 (PR02/L/H) | FFH |
| Priority specification flag register 03 (PR03L) | FFH |
| Priority specification flag register 12 (PR12/L/H) | FFH |
| Priority specification flag register 13 (PR13L) | FFH |
| Interrupt request flag register 0 (IF0/L/H) | 00H |
| Interrupt request flag register 1 (IF1/L/H) | 00H |
| Interrupt mask flag register 0 (MK0/L/H) | FFH |
| Interrupt mask flag register 1 (MK1/L/H) | FFH |
| Priority specification flag register 00 (PR00/L/H) | FFH |
| Priority specification flag register 01 (PR01/L/H) | FFH |
| Priority specification flag register 10 (PR10/L/H) | FFH |
| Priority specification flag register 11 (PR11/L/H) | FFH |
| Multiply and accumulation register (L) (MACRL) | 0000H |
| Multiply and accumulation register (H) (MACRH) | 0000H |
| Processor mode control register (PMC) | 00H |
| Comparator control register (CMPCTL) | 00H |
| Comparator input and output switching register (CMPSEL) | 00H |
| Comparator output monitor register (CMPMON) | 00H |
| CAN clock select register (CANCKSEL) | 00H |
| LIN clock select register (LINCKSEL) | 00H |
| Clock select register (CKSEL) | 00H |
| PLL control register (PLLCTL) | 00H |
| PLL status register (PLLSTS) | 00H |
| POR/CLM reset confirmation register (POCRES) | Undefined |
| High-speed DTC control register 0 (HDTCCR0) | 00H |
| High-speed DTC transfer count register 0 (HDTCCT0) | 00H |
| High-speed DTC transfer count reload register 0 (HDTRLD0) | 00H |
| High-speed DTC source address register 0 (HDTSAR0) | 0000H |
| High-speed DTC destination address register 0 (HDTDAR0) | 0000H |
| High-speed DTC control register 1 (HDTCCR1) | 00H |
| High-speed DTC transfer count register 1 (HDTCCT1) | 00H |
| High-speed DTC transfer count reload register 1 (HDTRLD1) | 00H |
| High-speed DTC source address register 1 (HDTSAR1) | 0000H |
| High-speed DTC destination address register 1 (HDTDAR1) | 0000H |
| DTC base address register (DTCBAR) | FDH |
| High-speed DTC channel select register 0 (SELHS0) | 3FH |
| High-speed DTC channel select register 1 (SELHS1) | 3FH |
| DTC activation enable register 0 (DTCEN0) | 00H |
| DTC activation enable register 1 (DTCEN1) | 00H |
| DTC activation enable register 2 (DTCEN2) | 00H |
| DTC activation enable register 3 (DTCEN3) | 00H |
| DTC activation enable register 4 (DTCEN4) | 00H |
| DTC activation enable register 5 (DTCEN5) [Note 8] | 00H |
| CRC operation mode control register (CRCMD) | 00H |
| CRC data register (CRCD) | 0000H |
| LIN wakeup baud rate select register (LWBR0/LWBR1) | 00H |

(**Notes** and **Remark** are given below Table 24-2, States of Hardware After Acceptance of a Reset (4/4).)

**Table 24-2. States of Hardware After Acceptance of a Reset (4/4)**

| Hardware | After Acceptance of a Reset Note 1 |
|---|---|
| LIN channel select register (LCHSEL) | 00H |
| LIN/UART baud rate prescaler 0 register (LBRP00/LBRP10) | 00H |
| LIN/UART baud rate prescaler 1 register (LBRP01/LBRP11) | 00H |
| LIN self test control register (LSTC0/LSTC1) | 00H |
| UART standby control register (LUSC0/LUSC1) | 00H |
| LIN/UART mode register (LMD0/LMD1) | 00H |
| LIN break field configuration register/UART configuration register (LBFC0/LBFC1) | 00H |
| LIN/UART space configuration register (LSC0/LSC1) | 00H |
| LIN wakeup configuration register (LWUP0/LWUP1) | 00H |
| LIN interrupt enable register (LIE0/LIE1) | 00H |
| LIN/UART error detection enable register (LEDE0/LEDE1) | 00H |
| LIN/UART control register (LCUC0/LCUC1) | 00H |
| LIN/UART transmit control register (LTRC0/LTRC1) | 00H |
| LIN/UART mode status register (LMST0/LMST1) | 00H |
| LIN/UART status register (LST0/LST1) | 00H |
| LIN/UART error status register (LEST0/LEST1) | 00H |
| LIN/UART data field configuration register (LDFC0/LDFC1) | 00H |
| LIN/UART ID buffer register (LIDB0/LIDB1) | 00H |
| LIN checksum buffer register (LCBR0/LCBR1) | 00H |
| UART data buffer 0 register (LUDB00/LUDB10) | 00H |
| LIN/UART data buffer 1 register (LDB01/LDB11) | 00H |
| LIN/UART data buffer 2 register (LDB02/LDB12) | 00H |
| LIN/UART data buffer 3 register (LDB03/LDB13) | 00H |
| LIN/UART data buffer 4 register (LDB04/LDB14) | 00H |
| LIN/UART data buffer 5 register (LDB05/LDB15) | 00H |
| LIN/UART data buffer 6 register (LDB06/LDB16) | 00H |
| LIN/UART data buffer 7 register (LDB07/LDB17) | 00H |
| LIN/UART data buffer 8 register (LDB08/LDB18) | 00H |
| UART operation enable register (LUOER0/LUOER1) | 00H |
| UART option register 1 (LUOR01/LUOR11) | 00H |
| UART transmit data register (LUTDR0/LUTDR1) | 0000H |
| UART receive data register (LURDR0/LURDR1) | 0000H |
| UART wait transmission data register (LUWTDR0/LUWTDR1) | 0000H |
| CAN-related registers | See **Table 3-6**. |

**Notes 1.** During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

**2.** The reset value differs for each chip.

**3.** The timer RD SFRs are undefined when FRQSEL4 = 1 in the user option byte (000C2H/010C2H) and TRD0EN = 0 in the PER1 register. If it is necessary to read the initial value, set $f_{CLK}$ to $f_{IH}$ and TRD0EN = 1 before reading.

**4.** If the AMPM bit (bit 3 of the real-time clock control register 0 (RTCC0)) is set to 1 after a reset, the setting of the hour count register (HOUR) becomes 00H.

**5.** The values depend on the source of the reset as shown in Table 24-3.

**6.** The reset value of the LVIS register varies depending on the reset source and the setting of the option byte.

**7.** The reset value of WDTE is determined by the option byte setting.

**8.** Only in the RL78/F14.

**Remark** The special function register (SFR) mounted depends on the product. See **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

**Table 24-3. States of Bits in RESF/LVIM/LVIS Registers When Reset Requests are Generated**

| Register \ Reset Source | | $\overline{\text{RESET}}$ Input | Reset by POR | Reset by Execution of Illegal Instruction | Reading from RESF | Reset by WDT | Reset by clock monitor | Reset by illegal-memory access | Reset by LVD |
|---|---|---|---|---|---|---|---|---|---|
| RESF | TRAP | Cleared (0) | Cleared (0) | Set (1) | Cleared (0) | Held | Held | Held | Held |
| | WDCLRF | | | Held | | Set (1) | Set (1) | Held | Held |
| | IAWRF | | | Held | | Held | Held | Set (1) | Held |
| | LVIRF | | | Held | | Held | Held | Held | Set (1) |
| POCRES | POCRES0 | Held | Cleared (0) | Held | Held | Held | Held | Held | Held |
| | CLKRF | Cleared (0) | Cleared (0) | Held | Held | Held | Set (1) | Held | Held |
| LVIM | LVISEN | Cleared (0) | Cleared (0) | Cleared (0) | Held | Cleared (0) | Cleared (0) | Cleared (0) | Held |
| | LVIOMSK | Held | Held | Held | Held | Held | Held | Held | Held |
| | LVIF | Held | Held | Held | Held | Held | Held | Held | Held |
| LVIS | | Cleared (00H/01H/81H) | Cleared (00H/01H/81H) | Cleared (00H/01H/81H) | Held | Cleared (00H/01H/81H) | Cleared (00H/01H/81H) | Cleared (00H/01H/81H) | Held |

**Caution The generation of reset signal other than an LVD reset sets as follows.**

- **When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H**
- **When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H**
- **When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H**

**Remark** The special function register (SFR) mounted depends on the product. See **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

## 24.1 Register for Confirming Reset Source

### 24.1.1 Reset control flag register (RESF)

Many internal reset generation sources exist in the RL78/F13 and RL78/F14. The reset control flag register (RESF) is used to store which source has generated the reset request.

The RESF register can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, reset by power-on-reset (POR) circuit, and reading the RESF register clear TRAP, WDCLRF, IAWRF, and LVIRF flags.

**Figure 24-5. Format of Reset Control Flag Register (RESF)**

Address: FFFA8H After reset: 00H Note 1 R

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| RESF | TRAP | 0 | 0 | WDCLRF | 0 | 0 | IAWRF | LVIRF |

| TRAP | Internal reset request by execution of illegal instruction Note 2 |
|---|---|
| 0 | Internal reset request is not generated, or the RESF register is cleared. |
| 1 | Internal reset request is generated. |

| WDCLRF | Internal reset request by watchdog timer (WDT) or clock monitor |
|---|---|
| 0 | Internal reset request is not generated, or the RESF register is cleared. |
| 1 | Internal reset request by the watchdog timer or the clock monitor is generated. |

| IAWRF | Internal reset request by illegal-memory access |
|---|---|
| 0 | Internal reset request is not generated, or the RESF register is cleared. |
| 1 | Internal reset request is generated. |

| LVIRF | Internal reset request by voltage detector (LVD) |
|---|---|
| 0 | Internal reset request is not generated, or the RESF register is cleared. |
| 1 | Internal reset request is generated. |

**Notes 1.** The value after reset varies depending on the reset source.

**2.** The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

**Caution Do not read data by a 1-bit memory manipulation instruction.**

The states of bits in the RESF register when reset requests are generated is shown in Table 24-4.

**Table 24-4. States of Bits in the RESF Register When Reset Requests are Generated**

| Reset Source / Flag | $\overline{\text{RESET}}$ Input | Reset by POR | Reset by Execution of Illegal Instruction | Reading from RESF | Reset by WDT | Reset by clock monitor | Reset by illegal-memory access | Reset by LVD |
|---|---|---|---|---|---|---|---|---|
| TRAP bit | Cleared (0) | Cleared (0) | Set (1) | Cleared (0) | Held | Held | Held | Held |
| WDCLRF bit | | | Held | | Set (1) | Set (1) | Held | Held |
| IAWRF bit | | | Held | | Held | Held | Set (1) | Held |
| LVIRF bit | | | Held | | Held | Held | Held | Set (1) |

### 24.1.2 POR/CLM reset confirmation register (POCRES)

The POR/CLM reset confirmation register (POCRES) is used to check whether a reset has been generated by a power-on reset or a clock-monitor reset.

When writing, the only effective value for the POCRES0 bit is 1. Writing 0 to this bit is ignored.

When writing, the only effective value for the CLKRF bit is 0. Writing 1 to this bit is ignored.

Set the POCRES register by a 1-bit or 8-bit memory manipulation instruction.

The POCRES0 bit only becomes 0 after a reset when the reset was generated by the power-on reset (POR) circuit.

The CLKRF bit becomes 0 after a reset when the reset was generated by the $\overline{\text{RESET}}$ input or power-on reset (POR) circuit.

**Remark** For confirming whether a reset was by the power-on reset (POR) circuit, the POCRES0 must be set to 1 beforehand.

**Figure 24-6. Format of POR Reset Register (POCRES)**

Address: F02C9H After reset: 00H Note R/W

| Symbol | 7 | 6 | 5 | <4> | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| POCRES | 0 | 0 | 0 | CLKRF | 0 | 0 | 0 | POCRES0 |

| POCRES0 | Internal reset request by POR reset |
|---|---|
| 0 | A POR request was generated or nothing has been written to this bit. |
| 1 | POR request is not generated. |

**Note** The value immediately before a reset is retained when a reset is from any source other than the POR circuit.

| CLKRF | Internal reset request by clock monitor |
|---|---|
| 0 | Internal reset request is not generated, or the CLKRF bit is cleared. |
| 1 | Internal reset request was generated. |

Table 24-5 shows the states of bits in the POCRES register when reset requests are generated.

**Table 24-5. States of Bits in the POCRES Register When Reset Requests are Generated**

| Flag \ Reset Source | $\overline{\text{RESET}}$ Input | Reset by POR | Reset by Execution of Illegal Instruction | Reading from RESF | Reset by WDT | Reset by clock monitor | Reset by illegal-memory access | Reset by LVD |
|---|---|---|---|---|---|---|---|---|
| POCRES0 | Held | Cleared (0) | Held | Held | Held | Held | Held | Held |
| CLKRF | Cleared (0) | Cleared (0) | Held | Held | Held | Set (1) | Held | Held |

# CHAPTER 25 POWER-ON-RESET CIRCUIT

## 25.1 Functions of Power-on-reset Circuit

The power-on-reset circuit (POR) has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage ($V_{DD}$) exceeds 1.56 V (typ.).

- Compares supply voltage ($V_{DD}$) and detection voltage ($V_{PDR}$ = 1.55 V (typ.)), generates internal reset signal when $V_{DD} < V_{PDR}$.

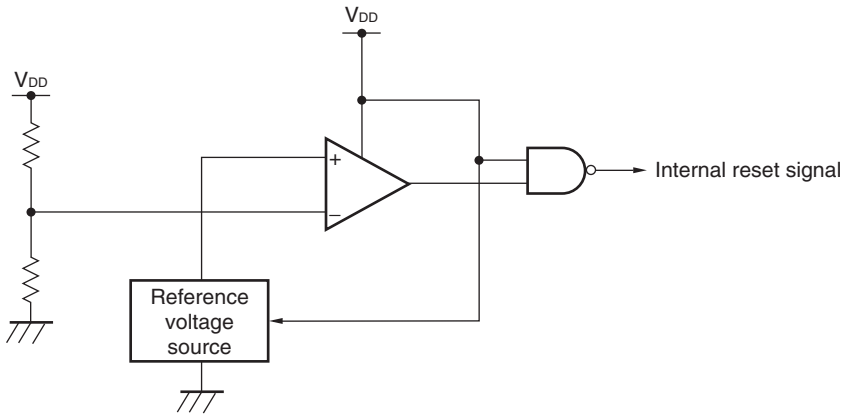
**Caution If an internal reset signal is generated in the POR circuit, the POCRES_0 and CLKRF flags of the POR/CLM reset confirmation register (POCRES) and the TRAP, WDCLRF, IAWRF, and LVIRF flags of the reset control flag register (RESF) are cleared (00H).**

**Remark** This product incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the RESF and POCRES registers for when an internal reset signal is generated by the power-on reset (POR), watchdog timer (WDT), clock monitor, voltage detector (LVD), illegal instruction execution, or illegal-memory access. The RESF register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the watchdog timer (WDT), voltage detector (LVD), illegal instruction execution, clock monitor, or illegal-memory access. The POCRES register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the clock monitor.
For details of the POCRES and RESF registers, see **CHAPTER 24 RESET FUNCTION**.

## 25.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 25-1.

**Figure 25-1. Block Diagram of Power-on-reset Circuit**




## 25.3 Operation of Power-on-reset Circuit

- An internal reset signal is generated on power application. When the supply voltage ($V_{DD}$) exceeds the detection voltage ($V_{POR}$ = 1.56 V (typ.)[Note]), the reset status is released.

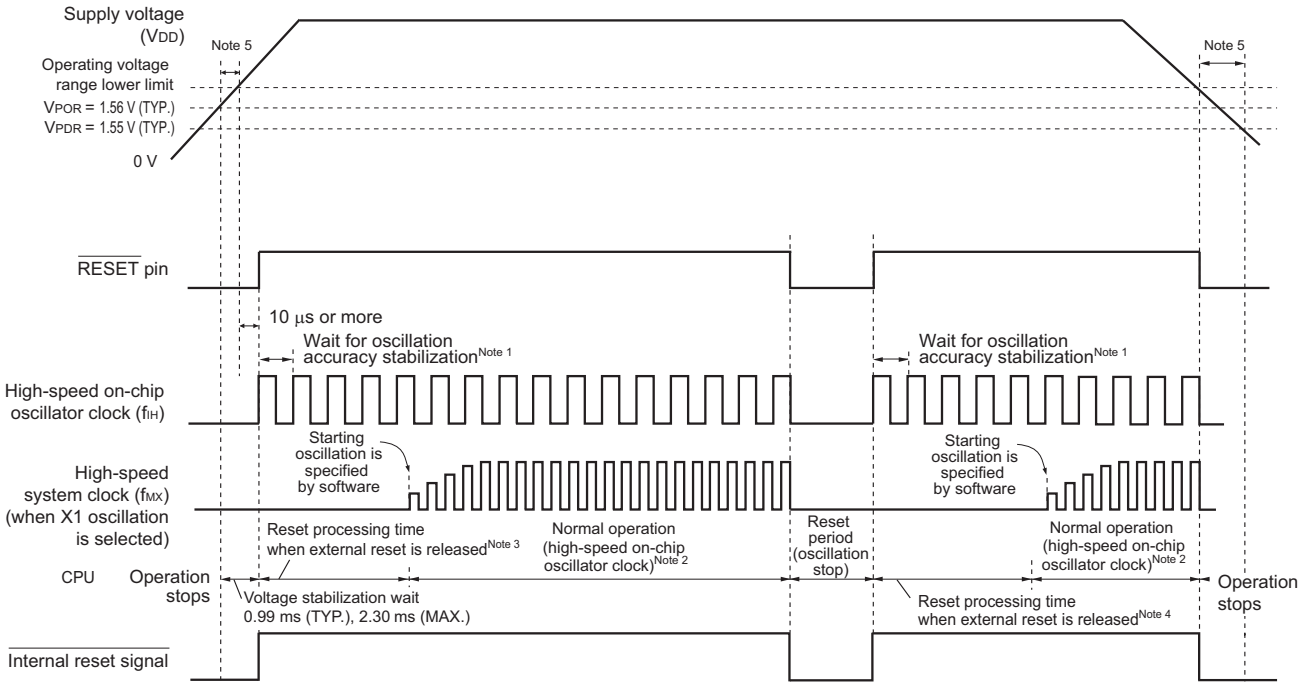
- The supply voltage ($V_{DD}$) and detection voltage ($V_{PDR}$ = 1.55 V (typ.)[Note]) are compared. When $V_{DD} < V_{PDR}$, the internal reset signal is generated.

**Note** When the user option byte function is used to enable the voltage detector by default, release from the reset state does not proceed until the value set in the option byte function is exceeded.

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown below.

**Figure 25-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/2)**

**(1) When the externally input reset signal on the $\overline{\text{RESET}}$ pin is used**




**Notes 1.** The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

**2.** The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

**3.** The time until normal operation starts includes the following reset processing time when the external reset is released (after the first release of POR) after the $\overline{\text{RESET}}$ signal is driven high (1) as well as the voltage stabilization wait time after $V_{POR}$ (1.56 V, typ.) is reached.
Reset processing time when the external reset is released is shown below.
After the first release of POR: 0.672 ms (typ.), 0.832 ms (max.) (when the LVD is in use)
0.399 ms (typ.), 0.519 ms (max.) (when the LVD is off)

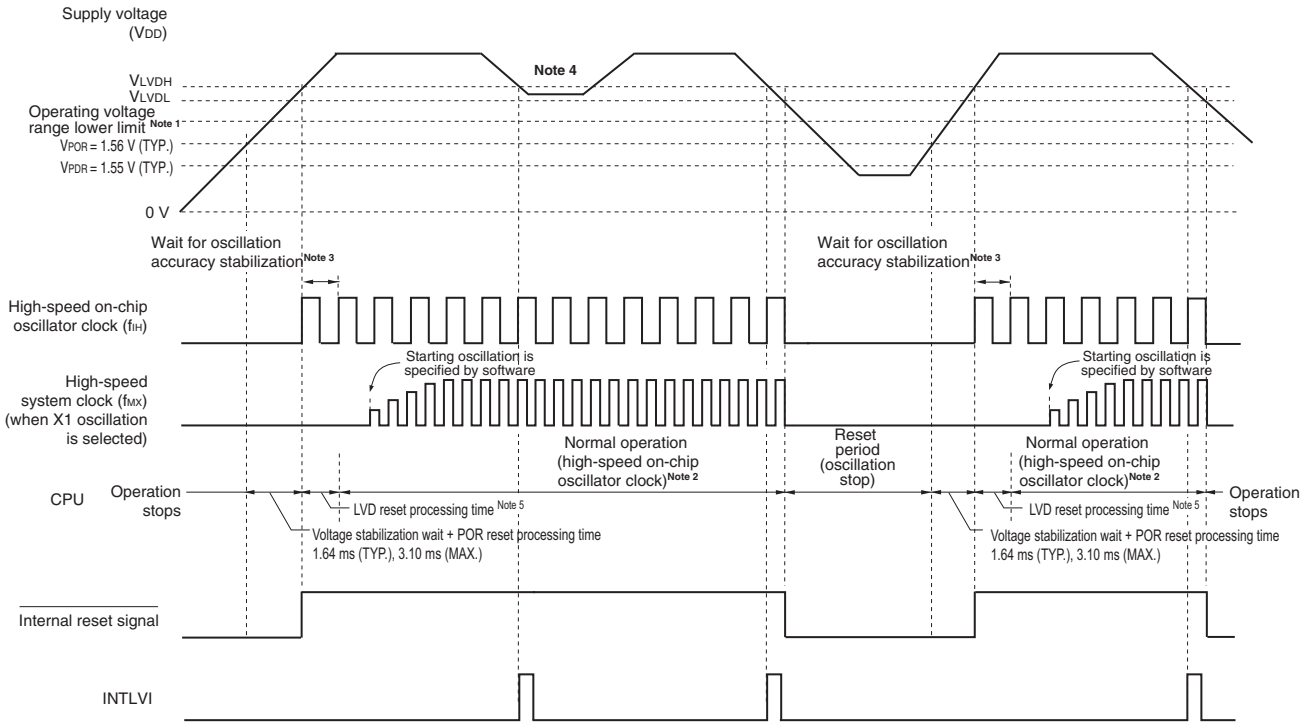
**4.** Reset processing time when the external reset is released after the second release of POR is shown below.
After the second release of POR: 0.531 ms (typ.), 0.675 ms (max.) (when the LVD is in use)
0.259 ms (typ.), 0.362 ms (max.) (when the LVD is off)

**5.** After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in the AC characteristics in **CHAPTER 34** to **CHAPTER 36 ELECTRICAL SPECIFICATIONS**. This is done by controlling the externally input reset signal. After power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the voltage falls below the operating range. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

**Remark** $V_{POR}$: POR power supply rise detection voltage
$V_{PDR}$: POR power supply fall detection voltage

**Figure 25-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/2)**

**(2) When LVD is interrupt & reset mode (option byte 000C1/020C1H: LVIMDS1, LVIMDS0 = 1, 0)**




**Notes 1.** The guaranteed range for operation is 2.7 V ≤ VDD ≤ 5.5 V. Only proceed with normal operations after VDD has reached or exceeded 2.7 V. If an operation may be generated at lower than 2.7V when the supply voltage falls or power-on, use the reset function of the voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

**2.** The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

**3.** The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

**4.** After the first interrupt request signal (INTLVI) is generated, the LVILV and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 26-8 Initial Setting of Interrupt and Reset Mode**, taking into consideration that the supply voltage might return to VLVDH or higher without falling below VLVDL.

**5.** LVD reset processing time: 0 to 0.0701 ms (MAX.)

**Remark** VLVDH, VLVDL: LVD detection voltage
VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

## 25.4 Cautions for Power-on-reset Circuit

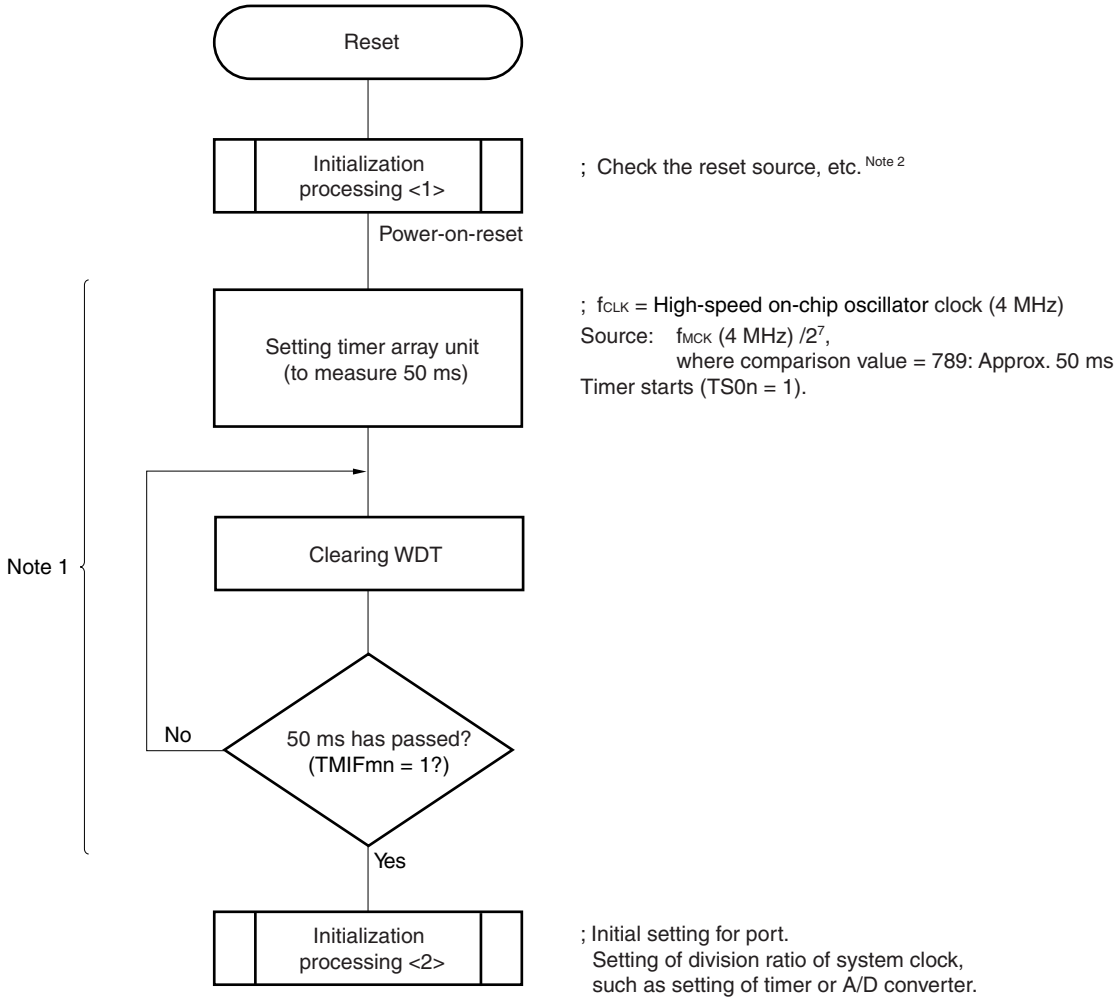
In a system where the supply voltage ($V_{DD}$) fluctuates for a certain period in the vicinity of the POR detection voltage ($V_{POR}$, $V_{PDR}$), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

**Figure 25-3. Example of Software Processing After Reset Release (1/2)**

(1) If supply voltage fluctuation is 50 ms or less in vicinity of POR detection voltage



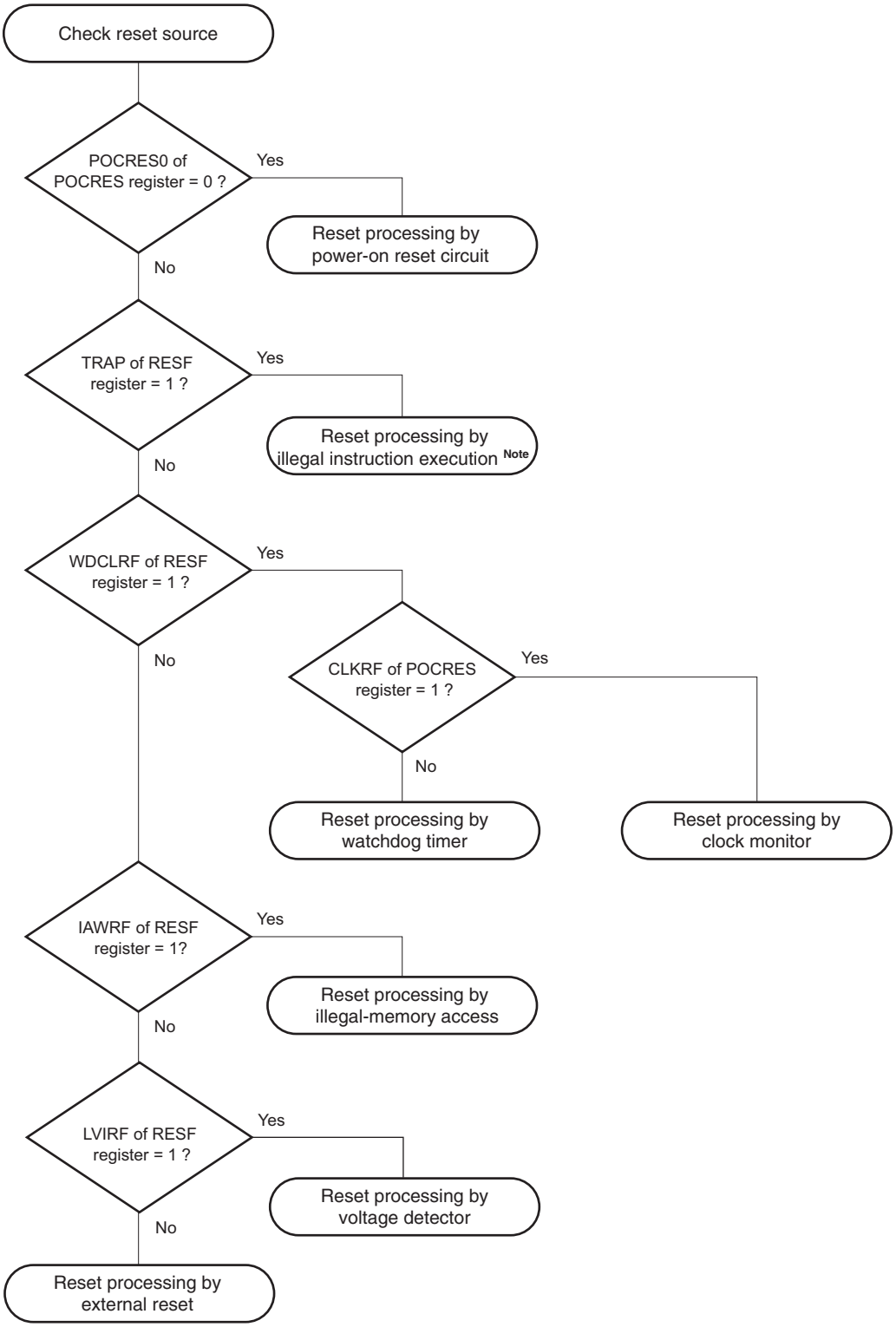

**Notes 1.** If reset is generated again during this period, initialization processing <2> is not started.

**2.** A flowchart is shown on the next page.

**Remark** m = 0, 1, n = 0 to 7

**Figure 25-3. Example of Software Processing After Reset Release (2/2)**

(2) Checking reset source




**Note** The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

# CHAPTER 26 VOLTAGE DETECTOR

## 26.1 Functions of Voltage Detector

The voltage detector (LVD) has the following functions.

- The LVD circuit compares the supply voltage ($V_{DD}$) with the detection voltage ($V_{LVDH}$, $V_{LVDL}$), and generates an internal reset or internal interrupt signal.
- The detection level for the power supply detection voltage ($V_{LVDH}$, $V_{LVDL}$) can be selected by using the option byte as one of 14 levels (For details, see **CHAPTER 29 OPTION BYTE**).
- Operable in STOP mode.
- The following three operation modes can be selected by using the option byte.

(a) Interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0)

For the two detection voltages selected by the option byte 000C1H/020C1H, the high-voltage detection level ($V_{LVDH}$) is used for generating interrupts and ending resets, and the low-voltage detection level ($V_{LVDL}$) is used for triggering resets.

(b) Reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1)

The detection voltage ($V_{LVD}$) selected by the option byte 000C1H/020C1H is used for triggering and ending resets.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)

The detection voltage ($V_{LVD}$) selected by the option byte 000C1H/020C1H is used for generating interrupts/reset release.

Two detection voltages ($V_{LVDH}$, $V_{LVDL}$) can be specified in the interrupt & reset mode, and one ($V_{LVD}$) can be specified in the reset mode and interrupt mode.

The reset and interrupt signals are generated as follows according to the option byte (LVIMDS0, LVIMDS1) selection.

| Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0) | Reset mode (LVIMDS1, LVIMDS0 = 1, 1) | Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1) |
|---|---|---|
| Generates an internal interrupt signal when $V_{DD} < V_{LVDH}$, and an internal reset when $V_{DD} < V_{LVDL}$.<br>Releases the reset signal when $V_{DD} \geq V_{LVDH}$. | Generates an internal reset signal when $V_{DD} < V_{LVD}$ and releases the reset signal when $V_{DD} \geq V_{LVD}$. | Generates an internal interrupt signal when $V_{DD}$ drops lower than $V_{LVD}$ ($V_{DD} < V_{LVD}$) or when $V_{DD}$ becomes $V_{LVD}$ or higher ($V_{DD} \geq V_{LVD}$).<br>Releases the reset signal when $V_{DD} \geq V_{LVD}$ at power on. |

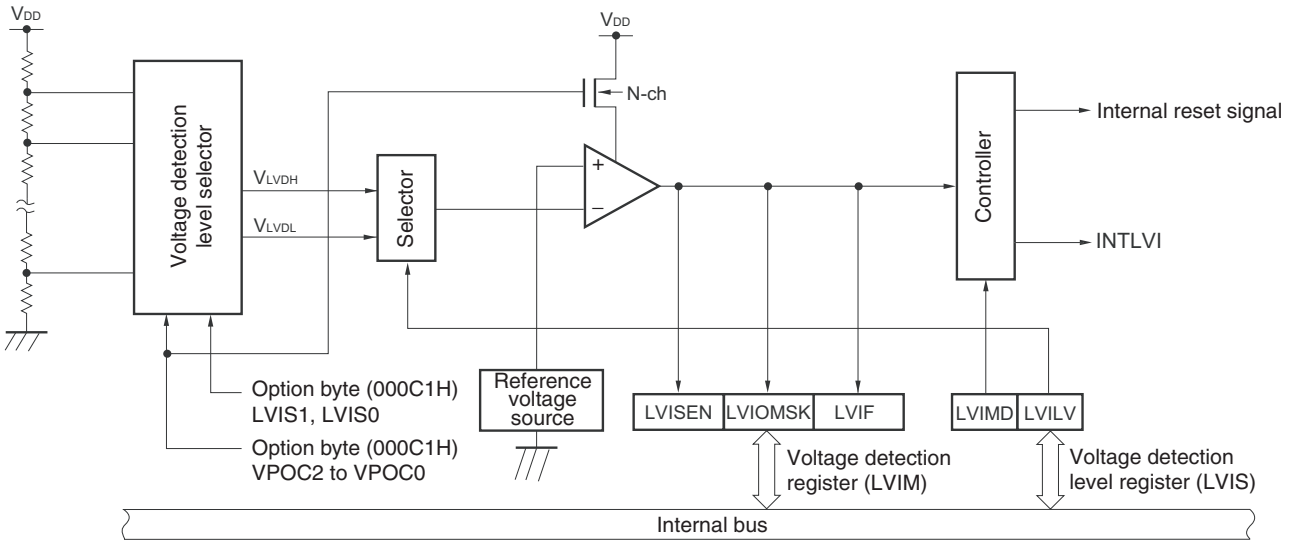
While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).

Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 24 RESET FUNCTION**.

## 26.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 26-1.

**Figure 26-1. Block Diagram of Voltage Detector**




## 26.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

### 26.3.1 Voltage detection register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the LVD output mask status.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H [Note 1].

**Figure 26-2. Format of Voltage Detection Register (LVIM)**

Address: FFFA9H After reset: 00H [Note 1] R/W [Note 2]

| Symbol | <7> | 6 | 5 | 4 | 3 | 2 | <1> | <0> |
|---|---|---|---|---|---|---|---|---|
| LVIM | LVISEN | 0 | 0 | 0 | 0 | 0 | LVIOMSK | LVIF |

| LVISEN | Specification of whether to enable or disable rewriting the voltage detection level register (LVIS) |
|---|---|
| 0 | Disabling rewriting |
| 1 | Enabling rewriting [Note 3] |

| LVIOMSK | Mask status flag of LVD output |
|---|---|
| 0 | Mask is invalid |
| 1 | Mask is valid [Note 4] |

| LVIF | Voltage detection flag |
|---|---|
| 0 | Supply voltage ($V_{DD}$) ≥ detection voltage ($V_{LVD}$), or when LVD operation is disabled |
| 1 | Supply voltage ($V_{DD}$) < detection voltage ($V_{LVD}$) |

**Notes 1.** The reset value changes depending on the reset source.
If the LVIS register is reset by LVD, it is not reset but holds the current value.
LVISEN is cleared to 0 by any reset other than one due to LVD.

**2.** Bits 0 and 1 are read-only.

**3.** This can only be set when LVIMDS1 and LVIMDS0 are set to 1 and 0 (interrupt and reset mode) by the option byte.

**4.** LVIOMSK bit is automatically set to 1 in the following periods and reset or interruption by LVD is masked.
- Period during LVISEN = 1
- Waiting period from the time when LVD interrupt is generated until LVD detection voltage becomes stable
- Waiting period from the time when the value of LVILV bit changes until LVD detection voltage becomes stable

### 26.3.2 Voltage detection level register (LVIS)

This register selects the voltage detection level.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation input sets this register to 00H/01H/81H Note1.

**Figure 26-3. Format of Voltage Detection Level Register (LVIS)**

Address: FFFAAH After reset: 00H/01H/81H Note 1 R/W

| Symbol | <7> | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| LVIS | LVIMD | 0 | 0 | 0 | 0 | 0 | 0 | LVILV |

| LVIMD Note 2 | Operation mode of voltage detection |
|---|---|
| 0 | Interrupt mode |
| 1 | Reset mode |

| LVILV Note 2 | LVD detection level |
|---|---|
| 0 | High-voltage detection level ($V_{LVDH}$) |
| 1 | Low-voltage detection level ($V_{LVDL}$ or $V_{LVD}$) |

**Notes 1.** The reset value changes depending on the reset source and the setting of the option byte.
This register is not cleared (00H) by LVD reset.
The generation of reset signal other than an LVD reset sets as follows.
- When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H
- When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H
- When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H

**2.** Writing 0 can only be allowed when LVIMDS1 and LVIMDS0 are set to 1 and 0 (interrupt and reset mode) by the option byte. In other cases, writing is not allowed and the value is switched automatically when reset or interrupt is generated.

**Cautions 1.** Only rewrite the value of the LVIS register after setting the LVISEN bit (bit 7 of the LVIM register) to 1.

**2.** Specify the LVD operation mode and detection voltage ($V_{LVDH}$, $V_{LVDL}$) by using the option byte (000C1H). Table 26-1 shows the option byte (000C1H) settings. For details about the option byte, see **CHAPTER 29 OPTION BYTE**.

**Table 26-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H/020C1H)**

- When used as interrupt & reset mode

| Detection voltage | | | Option byte Setting Value | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $V_{LVDH}$ | | $V_{LVDL}$ | LVIMDS1 | LVIMDS0 | VPOC2 | VPOC1 | VPOC0 | LVIS1 | LVIS0 |
| Rising edge | Falling edge | Falling edge | | | | | | | |
| 4.42 V | 4.32 V | 2.75 V | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 4.62 V | 4.52 V | 2.75 V | | | 0 | 1 | 0 | 0 | 0 |
| 3.32 V | 3.15 V | 2.75 V | | | 0 | 1 | 1 | 0 | 1 |
| 4.74 V | 4.64 V | | | | | | | 0 | 0 |
| Other than above | | | Setting prohibited | | | | | | |

- When used as reset mode

| Detection voltage | | Option byte Setting Value | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $V_{LVD}$ | | LVIMDS1 | LVIMDS0 | VPOC2 | VPOC1 | VPOC0 | LVIS1 | LVIS0 |
| Rising edge | Falling edge | | | | | | | |
| 2.81 V | 2.75 V | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 3.02 V | 2.96 V | | | 0 | 0 | 0 | 0 | 1 |
| 3.22 V | 3.15 V | | | 0 | 1 | 1 | 0 | 1 |
| 4.42 V | 4.32 V | | | 0 | 0 | 1 | 0 | 0 |
| 4.62 V | 4.52 V | | | 0 | 1 | 0 | 0 | 0 |
| 4.74 V | 4.64 V | | | 0 | 1 | 1 | 0 | 0 |
| Other than above | | Setting prohibited | | | | | | |

- When used as interrupt mode

| Detection voltage | | Option byte Setting Value | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $V_{LVD}$ | | LVIMDS1 | LVIMDS0 | VPOC2 | VPOC1 | VPOC0 | LVIS1 | LVIS0 |
| Rising edge | Falling edge | | | | | | | |
| 2.81 V | 2.75 V | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 3.02 V | 2.96 V | | | 0 | 0 | 0 | 0 | 1 |
| 3.22 V | 3.15 V | | | 0 | 1 | 1 | 0 | 1 |
| 4.42 V | 4.32 V | | | 0 | 0 | 1 | 0 | 0 |
| 4.62 V | 4.52 V | | | 0 | 1 | 0 | 0 | 0 |
| 4.74 V | 4.64 V | | | 0 | 1 | 1 | 0 | 0 |
| Other than above | | Setting prohibited | | | | | | |

- When LVDOFF

| Detection voltage | | Option byte Setting Value | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $V_{LVD}$ | | LVIMDS1 | LVIMDS0 | VPOC2 | VPOC1 | VPOC0 | LVIS1 | LVIS0 |
| Rising edge | Falling edge | | | | | | | |
| – | – | × | 1 | 1 | × | × | × | × |
| Other than above | | Setting prohibited | | | | | | |

**Caution When the LVD is off, it is necessary to perform an external reset. For an external reset, input a low level of at least 10 μs or more to the $\overline{\text{RESET}}$ pin. To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin before power-on, keep the low level for at least 10 μs during the period in which the supply voltage is within the operating range, and then input a high level. After power is applied, do not input a high level to the $\overline{\text{RESET}}$ pin during a period in which the supply voltage is not within the operating range.**

**Remark** ×: don't care

## 26.4 Operation of Voltage Detector

### 26.4.1 When used as reset mode

- When starting operation

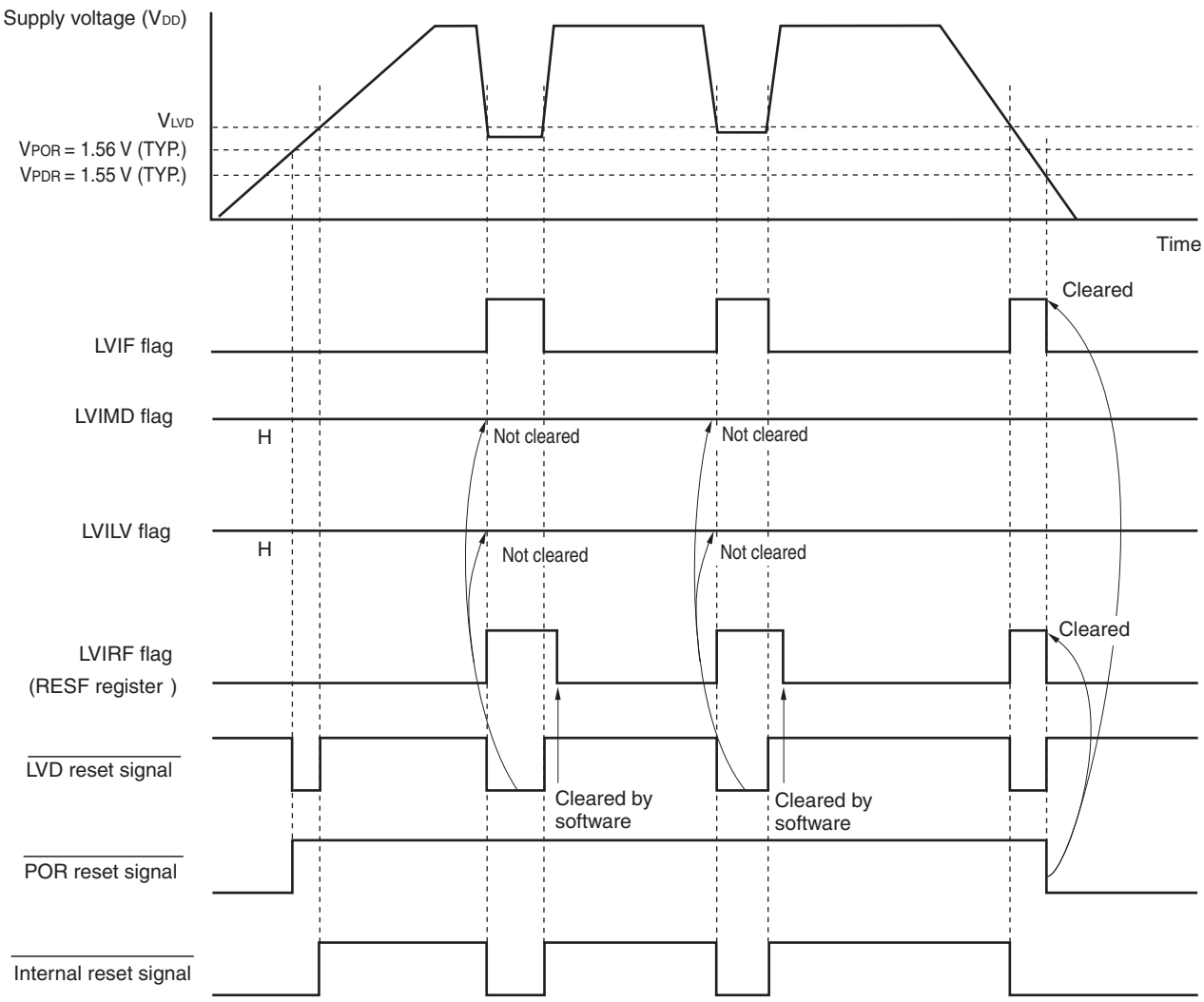
  Start in the following initial setting state.

  Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the detection voltage ($V_{LVD}$) by using the option byte 000C1H/020C1H.

  - Set bit 7 (LVISEN) of the voltage detection register (LVIM) to 0 (disable rewriting of voltage detection level register (LVIS))
  - When the option byte LVIMDS1 and LVIMDS0 are set to 1, the initial value of the LVIS register is set to 81H.
    Bit 7 (LVIMD) is 1 (reset mode).
    Bit 0 (LVILV) is 1 (low-voltage detection level: $V_{LVD}$).

  Figure 26-4 shows the timing of the internal reset signal generated by the voltage detector.

**Figure 26-4. Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)**




**Remark** VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

### 26.4.2 When used as interrupt mode

- When starting operation

  Specify the operation mode (the interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)) and the detection voltage ($V_{LVD}$) by using the option byte 000C1H/020C1H.

  Do not input a high level to the $\overline{\text{RESET}}$ pin when the supply voltage is not within the operating voltage range.

  Start in the following initial setting state.

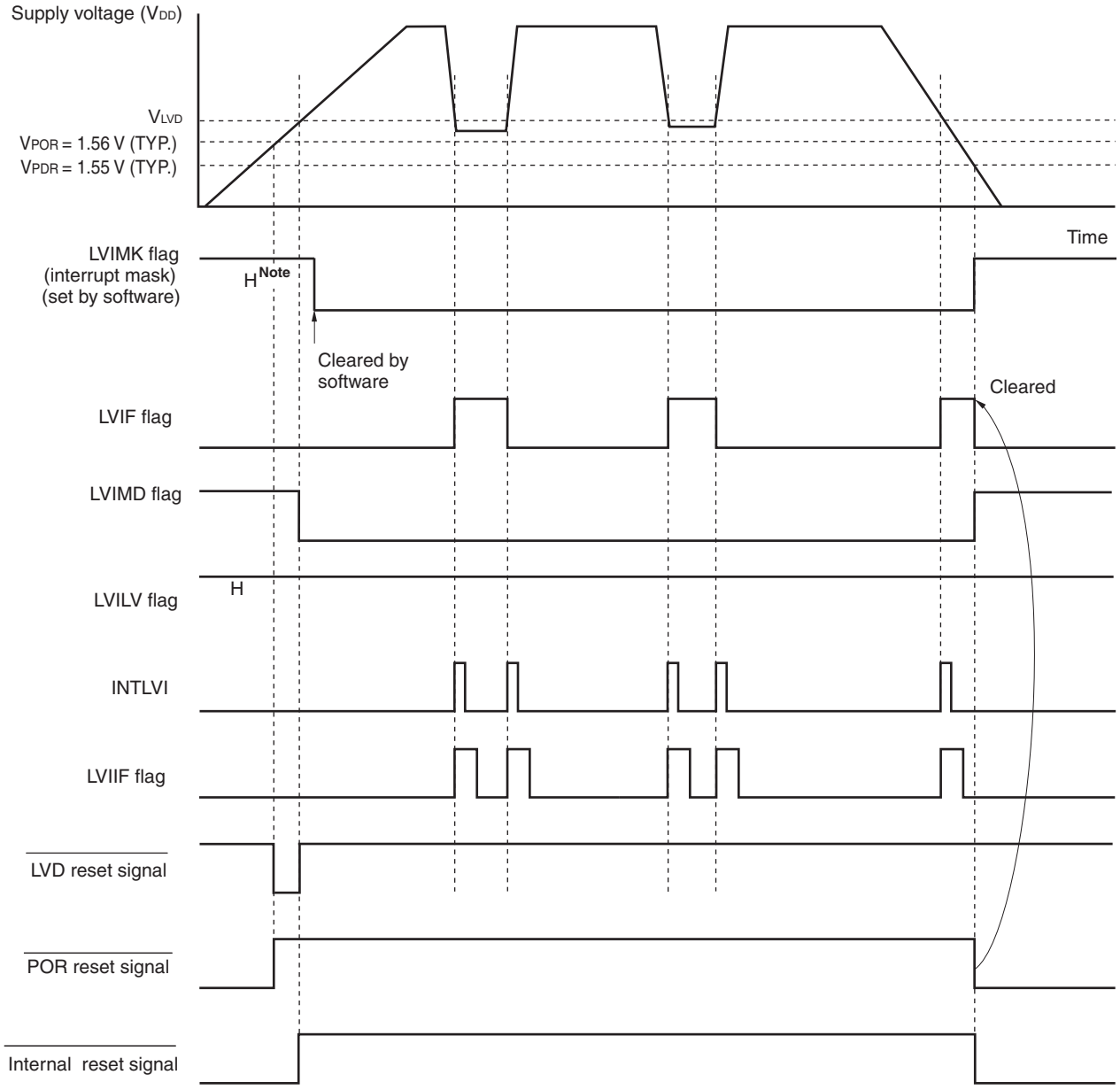
  - Set bit 7 (LVISEN) of the voltage detection register (LVIM) to 0 (disable rewriting of voltage detection level register (LVIS))
  - When the option byte LVIMDS1 is clear to 0 and LVIMDS0 is set to 1, the initial value of the LVIS register is set to 01H.

    Bit 7 (LVIMD) is 0 (interrupt mode).

    Bit 0 (LVILV) is 1 (low-voltage detection level: $V_{LVDL}$).

  Figure 26-5 shows the timing of the internal interrupt signal generated by the voltage detector.

**Figure 26-5. Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)**




**Note** The LVIMK flag is set to 1 by reset signal generation.

**Remark** $V_{POR}$: POR power supply rise detection voltage
$V_{PDR}$: POR power supply fall detection voltage

### 26.4.3 When used as interrupt and reset mode

- When starting operation

  Specify the operation mode (the interrupt and reset (LVIMDS1, LVIMDS0 = 1, 0)) and the detection voltage ($V_{LVDH}$, $V_{LVDL}$) by using the option byte 000C1H/020C1H.

  Start in the following initial setting state.

  - Set bit 7 (LVISEN) of the voltage detection register (LVIM) to 0 (disable rewriting of voltage detection level register (LVIS))
  - When the option byte LVIMDS1 is set to 1 and LVIMDS0 is clear to 0, the initial value of the LVIS register is set to 00H.

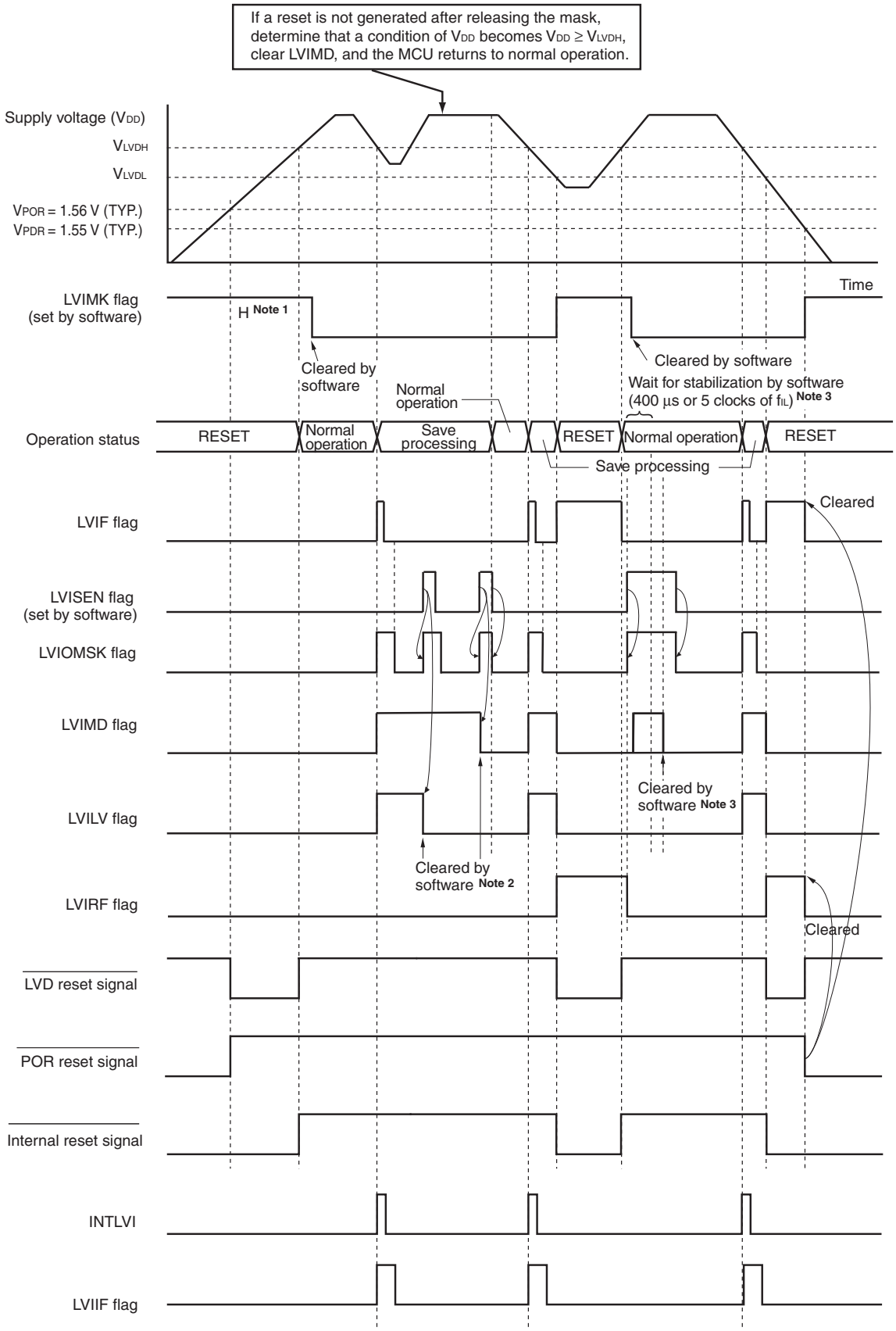
    Bit 7 (LVIMD) is 0 (interrupt mode).

    Bit 0 (LVILV) is 0 (high-voltage detection level: $V_{LVDH}$).

  Figures 26-6 shows the Timing of Voltage Detector Reset Signal and Interrupt Signal Generation.

  Perform the processing according to **Figure 26-7 Processing Procedure After an Interrupt Is Generated** and **Figure 26-8 Initial Setting of Interrupt and Reset Mode**.

**Figure 26-6. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)**




(**Notes** and **Remark** are listed on the next page.)

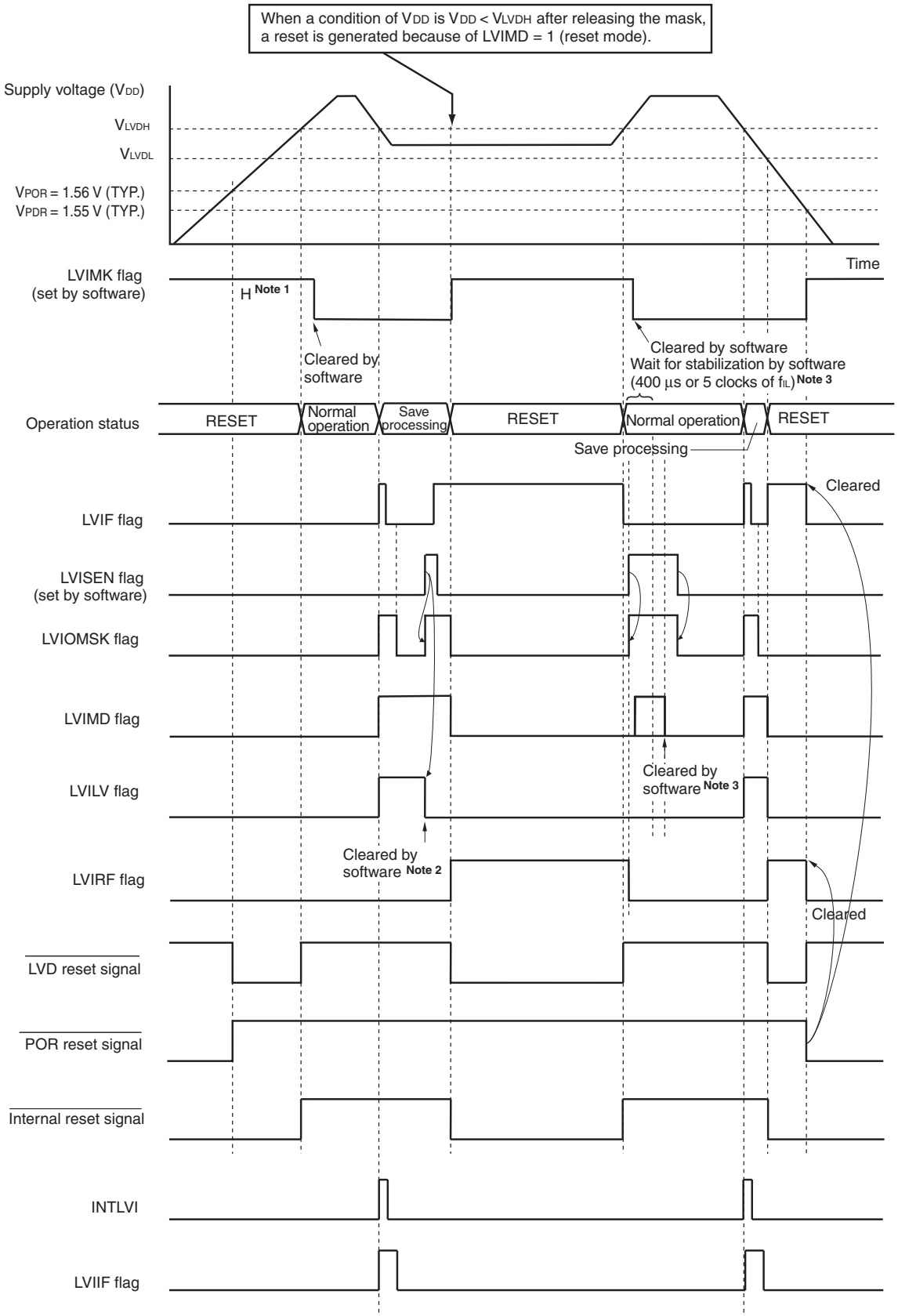
**Notes 1.** The LVIMK flag is set to 1 by reset signal generation.

**2.** After an interrupt is generated, perform the processing according to **Figure 26-7 Processing Procedure After an Interrupt Is Generated** in interrupt and reset mode.

**3.** After a reset is released, perform the processing according to **Figure 26-8 Initial Setting of Interrupt and Reset Mode** in interrupt and reset mode.

**Remark** $V_{POR}$: POR power supply rise detection voltage
$V_{PDR}$: POR power supply fall detection voltage

**Figure 26-6. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)**




(**Notes** and **Remark** are listed on the next page.)

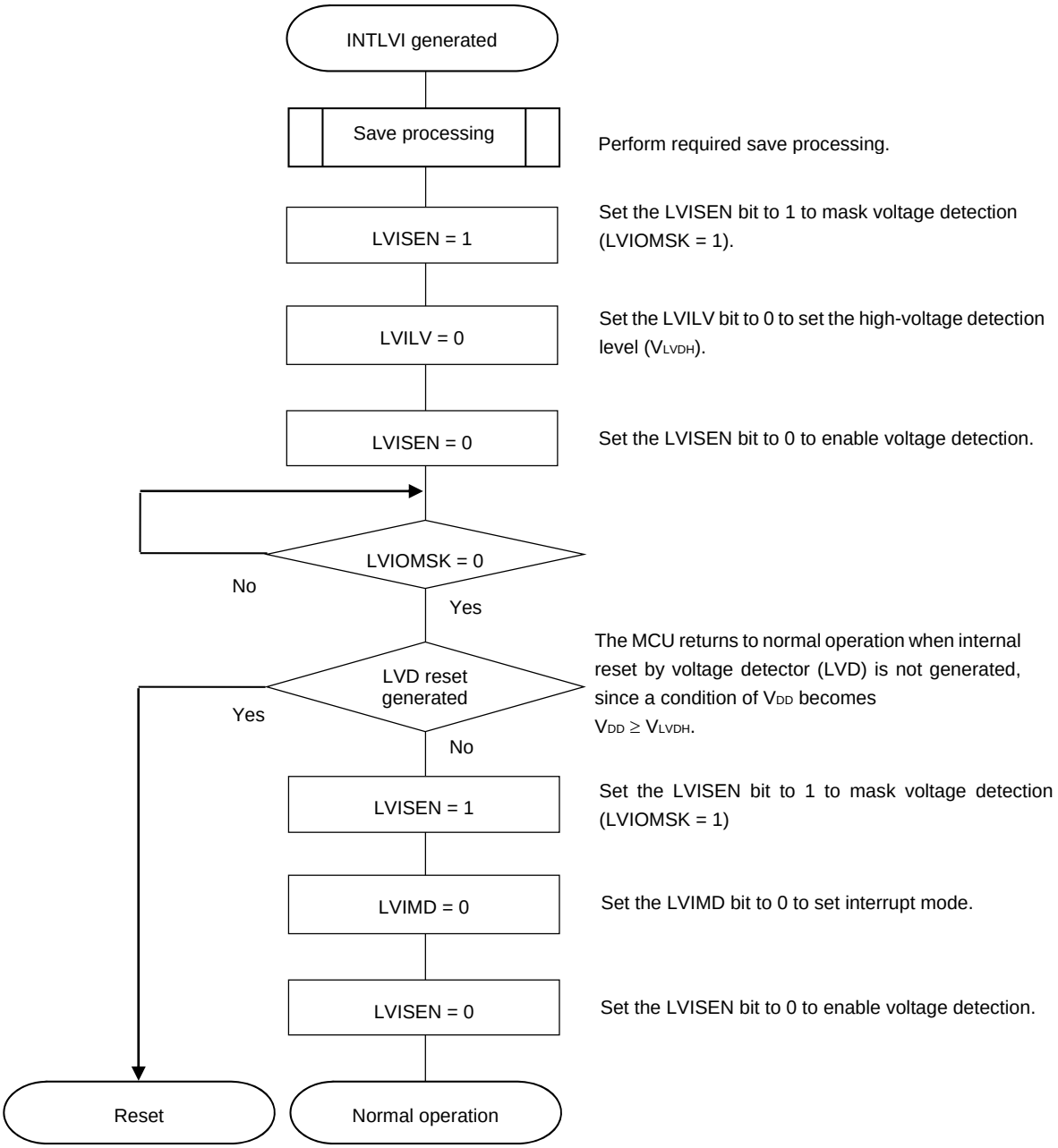
**Notes 1.** The LVIMK flag is set to 1 by reset signal generation.

**2.** After an interrupt is generated, perform the processing according to **Figure 26-7 Processing Procedure After an Interrupt Is Generated** in interrupt and reset mode.

**3.** After a reset is released, perform the processing according to **Figure 26-8 Initial Setting of Interrupt and Reset Mode** in interrupt and reset mode.

**Remark** $V_{POR}$: POR power supply rise detection voltage
$V_{PDR}$: POR power supply fall detection voltage

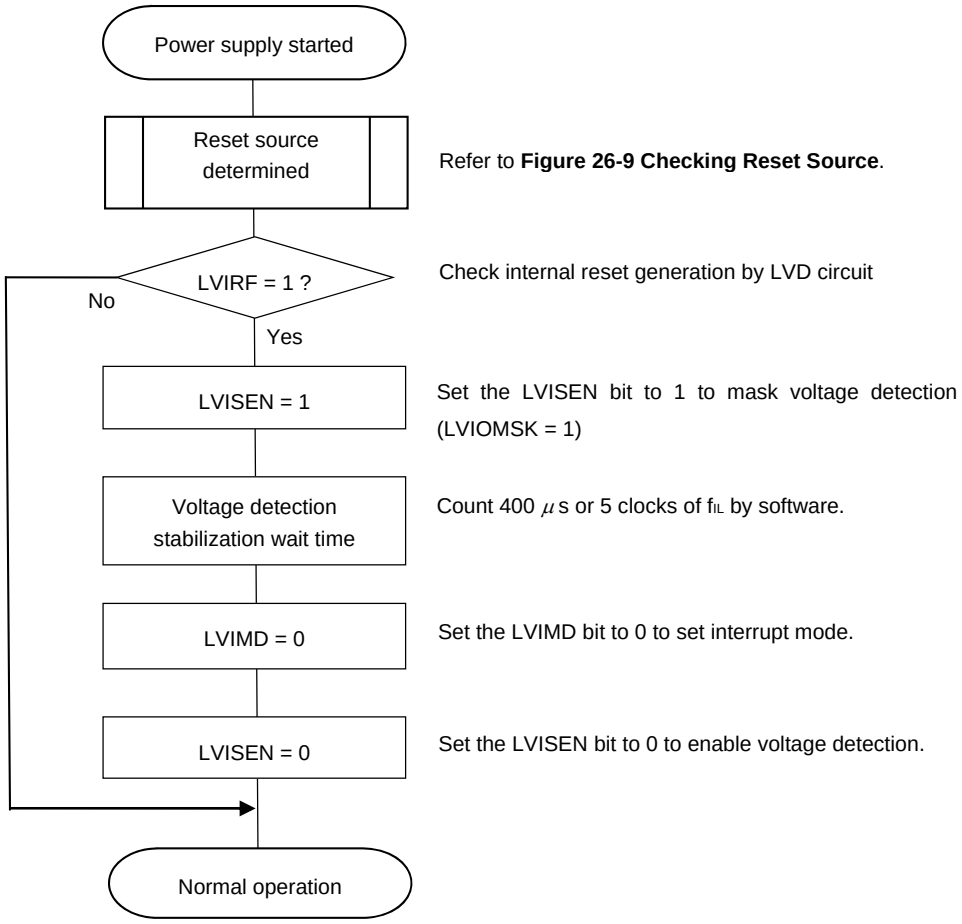
**Figure 26-7. Processing Procedure After an Interrupt Is Generated**



When setting an interrupt and reset mode (LVIMDS1, LVIMDS0 = 1, 0), voltage detection stabilization wait time for 400 $\mu$s or 5 clocks of $f_{IL}$ is necessary after LVD reset is released (LVIRF = 1). After waiting until voltage detection stabilizes, (0) clear the LVIMD bit for initialization. While voltage detection stabilization wait time is being counted and when the LVIMD bit is rewritten, set LVISEN to 1 to mask a reset or interrupt generation by LVD.

Figure 26-8 shows the procedure for initial setting of interrupt and reset mode.

**Figure 26-8. Initial Setting of Interrupt and Reset Mode**



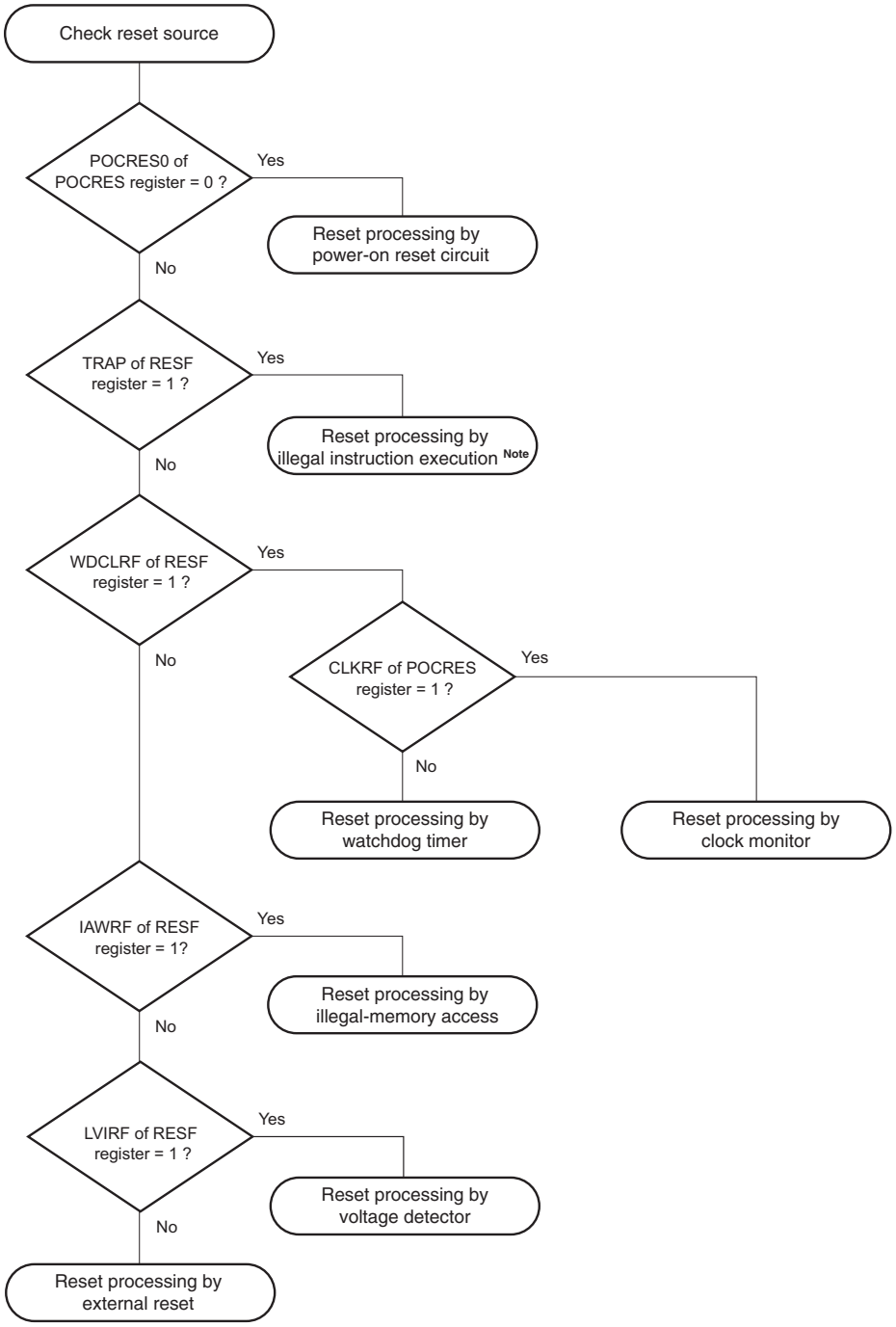

**Remark** $f_{IL}$: Low-speed on-chip oscillator clock frequency

## 26.5 Cautions for Voltage Detector

### 26.5.1 Checking reset source

When a reset occurs, check the reset source by using the following method.

**Figure 26-9. Checking Reset Source**



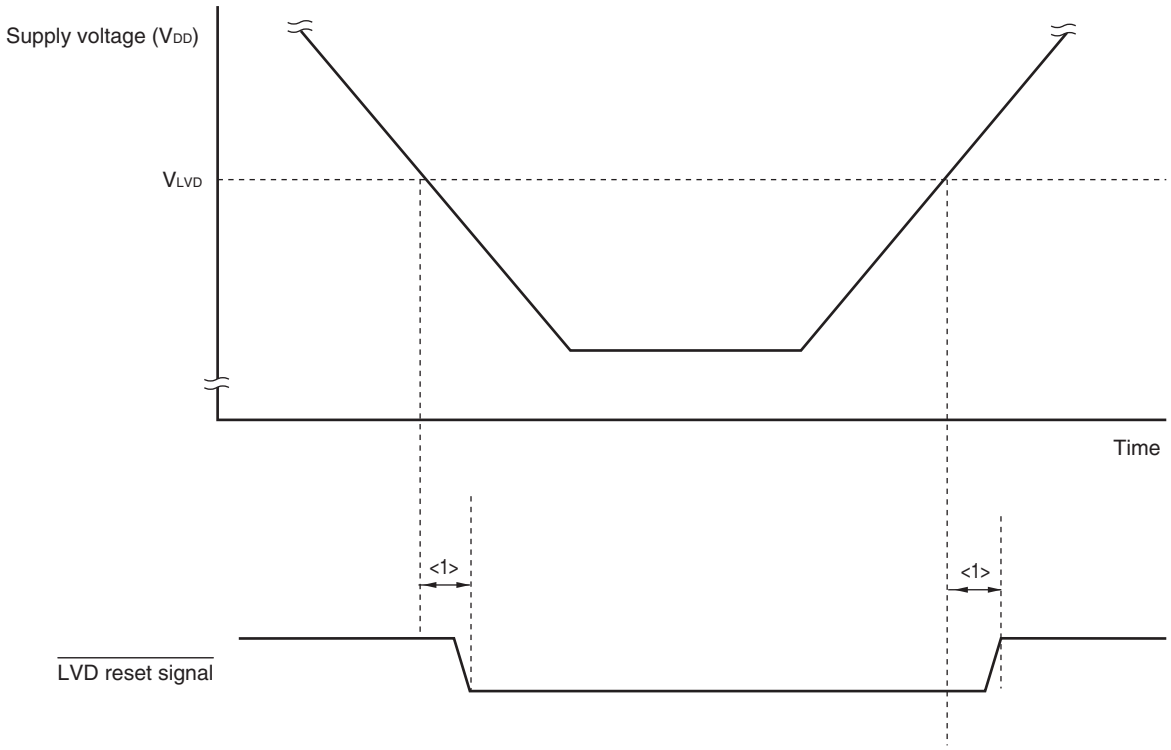

**Note** When instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

### 26.5.2 Delay from the time LVD reset source is generated until the time LVD reset has been generated or released

There is some delay from the time supply voltage ($V_{DD}$) < LVD detection voltage ($V_{LVD}$) until the time LVD reset has been generated.

In the same way, there is also some delay from the time LVD detection voltage ($V_{LVD}$) ≤ supply voltage ($V_{DD}$) until the time LVD reset has been released (see **Figure 26-10**).

**Figure 26-10. Delay from The Time LVD Reset Source Is Generated until The Time LVD Reset Has Been Generated or Released**




<1>: Detection delay (300 μs (MAX.))

# CHAPTER 27 SAFETY FUNCTIONS

## 27.1 Overview of Safety Functions

The following safety functions are provided in the RL78/F13 and RL78/F14 to comply with the IEC60730 and IEC61508 safety standards.

These functions enable the microcontroller to self-diagnose abnormalities and stop operating if an abnormality is detected.

**(1) Flash memory CRC operation function (high-speed CRC, general-purpose CRC)**

This detects data errors in the flash memory by performing CRC operations.

Two CRC functions are provided in the RL78/F13 and RL78/F14 that can be used according to the application or purpose of use.

- High-speed CRC: The CPU can be stopped and a high-speed check executed on its entire code flash memory area during the initialization routine.
- General CRC: This can be used for checking various data in addition to the code flash memory area while the CPU is running.

**(2) RAM-ECC function**

2-bit error detection and 1- bit error correction are available.

**(3) CPU stack pointer monitor function**

This detects underflows and overflows of the stack pointer.

**(4) Clock monitoring function**

The system clock ($f_{MAIN}$) and main/PLL select clock ($f_{MP}$) are monitored to detect oscillation stopping.

**(5) RAM guard function**

This prevents RAM data from being rewritten when the CPU freezes.

**(6) SFR guard function**

This prevents SFRs from being rewritten when the CPU freezes.

**(7) Invalid memory access detection function**

This detects illegal accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

**(8) Frequency detection function**

This uses TAU to detect the oscillation frequency.

**(9) A/D test function**

This is used to perform a self-check of A/D conversion by performing A/D conversion on the internal reference voltage.

**(10) Digital output signal level detection function for I/O ports**

When the I/O ports are output mode in which PMm bit of the port mode register (PMm) is 0, the output level of the pin can be read.

**Remarks 1.** m = 0, 1, 3 to 10, 12 to 15, n = 0 to 7

**2.** For details on usage, see Application Note (under creation).

## 27.2 Registers Used by Safety Functions

The safety functions use the following registers:

| Register | Each Function of Safety Function |
|---|---|
| • Flash memory CRC control register (CRC0CTL)<br>• Flash memory CRC operation result register (PGCRCL) | Flash memory CRC operation function (high-speed CRC) |
| • CRC input register (CRCIN)<br>• CRC operation mode control register (CRCMD)<br>• CRC data register (CRCD) | CRC operation function (general-purpose CRC) |
| • Error address store register (ERADR)<br>• 1-bit error detection interrupt enable register (ECCIER)<br>• Bit error detection register (ECCER)<br>• ECC test protect register (ECCTPR)<br>• ECC test mode register (ECCTMDR)<br>• Write data inversion register (ECCDWRVR) | RAM-ECC function |
| • SPM control register (SPMCTRL)<br>• SP overflow address setting register (SPOFR)<br>• SP underflow setting register (SPUFR) | Stack pointer monitor function |
| • Invalid memory access detection control register (IAWCTL) | RAM guard function |
| | SFR guard function |
| | Invalid memory access detection function |
| • Timer input select register 0 (TIS0)<br>• System clock control register (CKC) | Frequency detection function |
| • A/D test register (ADTES)<br>• Analog input channel specification register (ADS) | A/D test function |
| • Port mode select register (PMS) | Digital output signal level detection function for I/O ports |

The content of each register is described in **27.3 Operation of Safety Functions**.

## 27.3 Operation of Safety Functions

### 27.3.1 Flash memory CRC operation function (high-speed CRC)

The IEC60730 standard mandates the checking of data in the flash memory, and recommends using CRC to do it. The high-speed CRC provided in the RL78/F13 and RL78/F14 can be used to check the entire code flash memory area during the initialization routine. The high-speed CRC can be executed only when the program is allocated on the RAM and in the HALT mode of the main system clock.

High-speed CRC operations are performed by stopping the CPU and reading 32 bits of data from the flash memory in one clock cycle. The feature of this operation is the short time it takes until the end of the check (for example, 64 Kbytes of flash memory are checked in 512 μs when the operating clock is at 32 MHz).

The CRC generator polynomial used complies with "$X^{16} + X^{12} + X^5 + 1$" of CRC-16-CCITT.

The high-speed CRC operates in MSB first order from bit 31 to bit 0.

**Caution The result of the CRC operation will differ from that in on-chip debugging because of allocation of the monitor program.**

**Remark** The operation result is different between the high-speed CRC and the general CRC, because the general CRC operates in LSB first order.

**<Control register>**

**(1) Flash memory CRC control register (CRC0CTL)**

This register is used to control the operation of the high-speed CRC ALU, as well as to specify the operation range.

The CRC0CTL register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 27-1. Format of Flash Memory CRC Control Register (CRC0CTL)**

Address: F02F0H After reset: 00H R/W

| Symbol | <7> | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CRC0CTL | CRC0EN | 0 | FEA5 | FEA4 | FEA3 | FEA2 | FEA1 | FEA0 |

| CRC0EN | Control of high-speed CRC ALU operation |
|---|---|
| 0 | Stop the operation. |
| 1 | Start the operation according to HALT instruction execution. |

| FEA5 | FEA4 | FEA3 | FEA2 | FEA1 | FEA0 | High-speed CRC operation range |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 00000H to 3FFBH (16 Kbytes - 4 bytes) |
| 0 | 0 | 0 | 0 | 0 | 1 | 00000H to 7FFBH (32 Kbytes - 4 bytes) |
| 0 | 0 | 0 | 0 | 1 | 0 | 00000H to BFFBH (48 Kbytes - 4 bytes) |
| 0 | 0 | 0 | 0 | 1 | 1 | 00000H to FFFBH (64 Kbytes - 4 bytes) |
| 0 | 0 | 0 | 1 | 0 | 0 | 00000H to 13FFBH (80 Kbytes - 4 bytes) |
| 0 | 0 | 0 | 1 | 0 | 1 | 00000H to 17FFBH (96 Kbytes - 4 bytes) |
| 0 | 0 | 0 | 1 | 1 | 0 | 00000H to 1BFFBH (112 Kbytes - 4 bytes) |
| 0 | 0 | 0 | 1 | 1 | 1 | 00000H to 1FFFBH (128 Kbytes - 4 bytes) |
| 0 | 0 | 1 | 0 | 0 | 0 | 00000H to 23FFBH (144 Kbytes - 4 bytes) |
| 0 | 0 | 1 | 0 | 0 | 1 | 00000H to 27FFBH (160 Kbytes - 4 bytes) |
| 0 | 0 | 1 | 0 | 1 | 0 | 00000H to 2BFFBH (176 Kbytes - 4 bytes) |
| 0 | 0 | 1 | 0 | 1 | 1 | 00000H to 2FFFBH (192 Kbytes - 4 bytes) |
| 0 | 0 | 1 | 1 | 0 | 0 | 00000H to 33FFBH (208 Kbytes - 4 bytes) |
| 0 | 0 | 1 | 1 | 0 | 1 | 00000H to 37FFBH (224 Kbytes - 4 bytes) |
| 0 | 0 | 1 | 1 | 1 | 0 | 00000H to 3BFFBH (240 Kbytes - 4 bytes) |
| 0 | 0 | 1 | 1 | 1 | 1 | 00000H to 3FFFBH (256 Kbytes - 4 bytes) |
| Other than the above | | | | | | Setting prohibited |

**Remark** Input the expected CRC operation result value to be used for comparison in the lowest 4 bytes of the flash memory. Note that the operation range will thereby be reduced by 4 bytes.
The lowest 4 bytes of 16 Kbytes are used to store the expected value, so operation is not performed on these bytes.

**(2) Flash memory CRC operation result register (PGCRCL)**

This register is used to store the high-speed CRC operation results.

The PGCRCL register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

**Figure 27-2. Format of Flash Memory CRC Operation Result Register (PGCRCL)**

Address: F02F2H After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| PGCRCL | PGCRC15 | PGCRC14 | PGCRC13 | PGCRC12 | PGCRC11 | PGCRC10 | PGCRC9 | PGCRC8 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PGCRC7 | PGCRC6 | PGCRC5 | PGCRC4 | PGCRC3 | PGCRC2 | PGCRC1 | PGCRC0 |

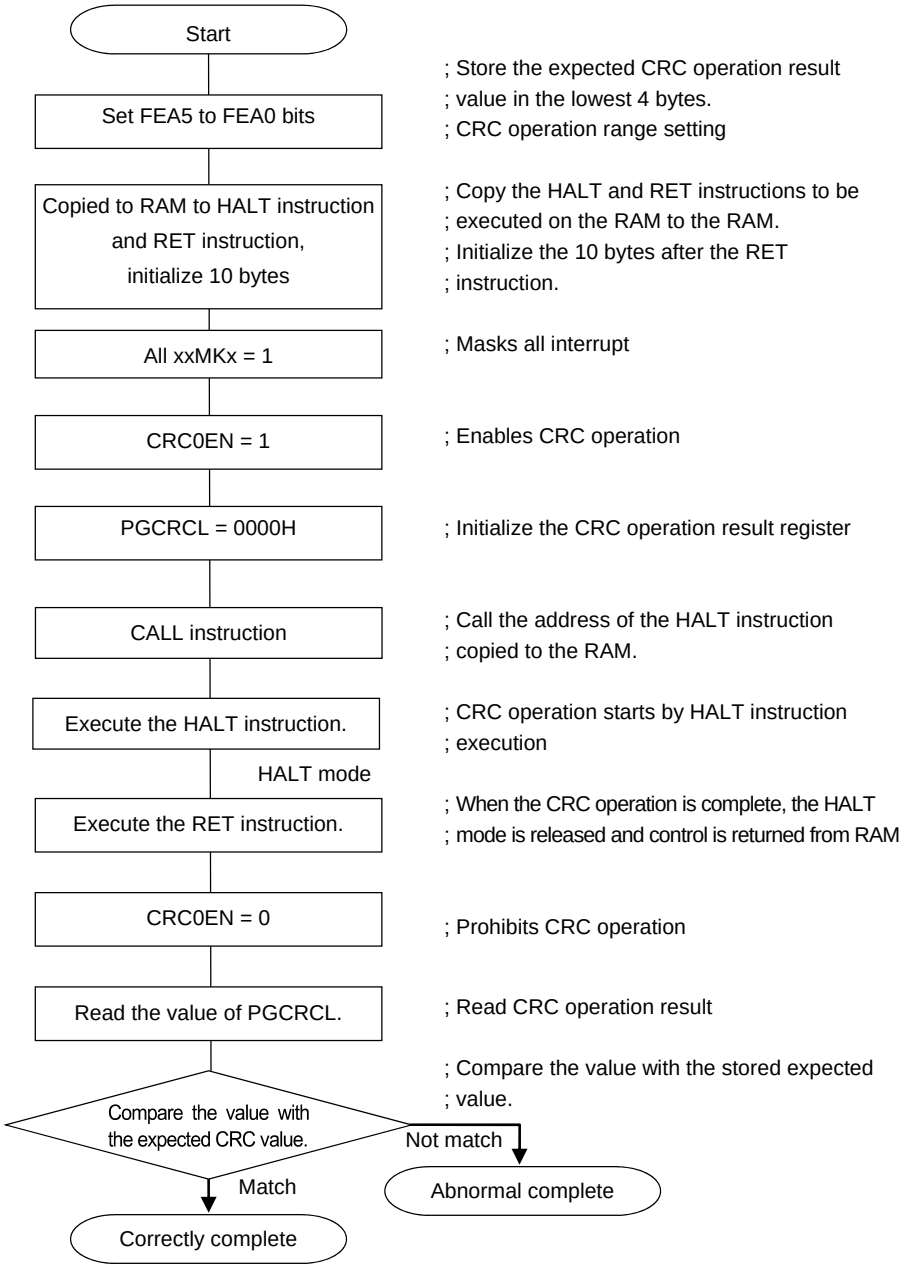
| | |
|---|---|
| PGCRC15 to PGCRC0 | High-speed CRC operation results |
| 0000H to FFFFH | Store the high-speed CRC operation results. |

**Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.**

Figure 27-3 shows the flowchart of flash memory CRC operation function (high-speed CRC).

**<Operation flow>**

**Figure 27-3. Flowchart of Flash Memory CRC Operation Function (High-speed CRC)**




**Cautions 1. The CRC operation is executed only on the code flash.**

**2. Store the expected CRC operation value in the area below the operation range in the code flash.**

**3. Boot swapping is not performed while the CRC operation is being executed.**

**4. The CRC operation is enabled by executing the HALT instruction in the RAM area.**
**Be sure to execute the HALT instruction in RAM area.**

The expected CRC value can be calculated by using tools such as the CubeSuite+ development environment (see the CubeSuite+ user's manual for details).

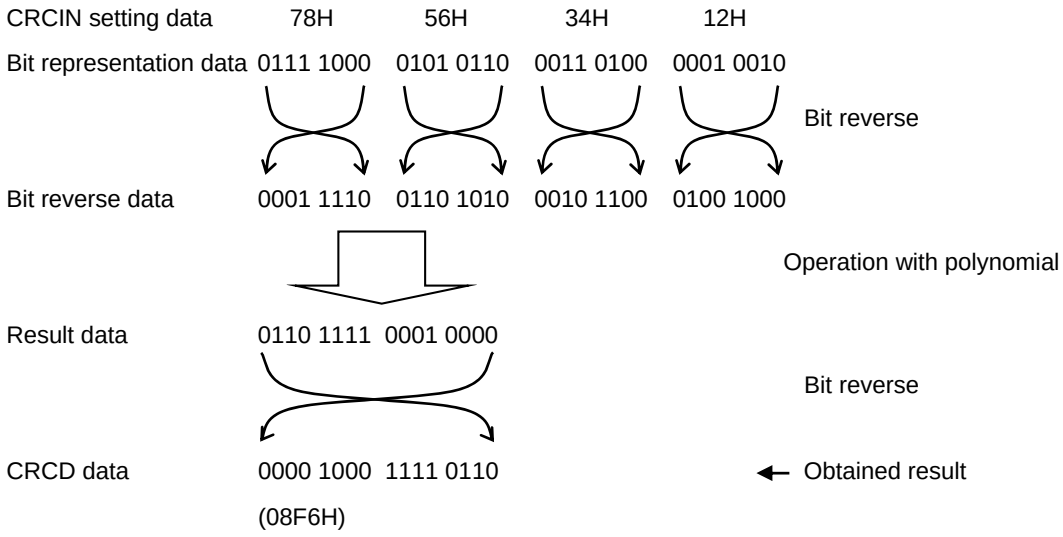
### 27.3.2 CRC operation function (general-purpose CRC)

In order to guarantee safety during operation, the IEC61508 standard mandates the checking of data even while the CPU is operating.

In the RL78/F13 and RL78/F14, a general CRC operation can be executed as a peripheral function while the CPU is operating. The general CRC can be used for checking various data in addition to the code flash memory area. The data to be checked can be specified by using software (a user-created program). In HALT mode, CRC operations can only proceed during DTC transfer.

The general CRC operation can be executed in the main system clock operation mode as well as the subsystem clock operation mode.

The CRC generator polynomial used is "$X^{16} + X^{12} + X^5 + 1$" of CRC-16-CCITT. The data to be input is inverted in bit order and then calculated to allow for LSB-first communication. For example, if the data 12345678H is sent from the LSB, values are written to the CRCIN register in the order of 78H, 56H, 34H, and 12H, enabling a value of 08F6H to be obtained from the CRCD register. This is the result obtained by executing a CRC operation on the bit rows shown below, which consist of the data 12345678H inverted in bit order.




**Caution Setting a software break in a target area of CRC operation alters the result of the CRC operation because the debugger changes the row where the software break is to be set into a break instruction during program execution.**

**<Control register>**

**(1) CRC input register (CRCIN)**

CRCIN register is an 8-bit register that is used to set the CRC operation data of general-purpose CRC.

The possible setting range is 00H to FFH.

The CRCIN register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 27-4. Format of CRC Input Register (CRCIN)**

Address: FFFACH After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CRCIN | | | | | | | | |

| CRCIN7 to CRCIN0 | Setting the CRC operation data of general-purpose CRC |
|---|---|
| 00H to FFH | Data input when supporting CRC-CCITT |
| 00H to 0FH | Data input when conforming to SENT Note |

**Note** For CRCIN register write when conforming to SENT, write valid data to the lower 4 bits (bits 3 to 0) and write 0 to the other bits (if any value other than 0 is written to, the written value is read because the bits other than the lower 4 bits are not processed).

**(2) CRC operation mode control register (CRCMD)**

CRCMD register is used to select the general-purpose CRC operation mode.

The CRCMD register can be set by an 8-bit memory manipulation instruction.

**Figure 27-5. Format of CRC Operation Mode Control Register (CRCMD)**

Address: F02F9H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| CRCMD | - | - | - | - | - | - | - | POLYSEL |

| POLYSEL | CRC code generation circuit select bit |
|---|---|
| 0 | CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) |
| 1 | Conform to SENT ($X^4 + X^3 + X^2 + 1$) |

**Cautions 1. To generate CRC code conforming to SENT, set the POLYSEL bit in the CRCMD register.**

**2. Bits 7 to 1 are always read as 0. The write value should always be 0.**

**(3) CRC data register (CRCD)**

This register is used to store the general-purpose CRC operation result.

The possible setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock ($f_{CLK}$) has elapsed from the time CRCIN register is written, the CRC operation result is stored to the CRCD register.

The CRCD register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

**Figure 27-6. Format of CRC Data Register (CRCD)**

Address: F02FAH After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CRCD | | | | | | | | | | | | | | | | |

| CRCD15 to CRCD0 | Store the general-purpose CRC operation result Note 1, 2 |
|---|---|
| 0000H to FFFFH | CRC operation result when supporting CRC-CCITT |
| 0000H to 000FH | CRC operation result when conforming to SENT Note3 |

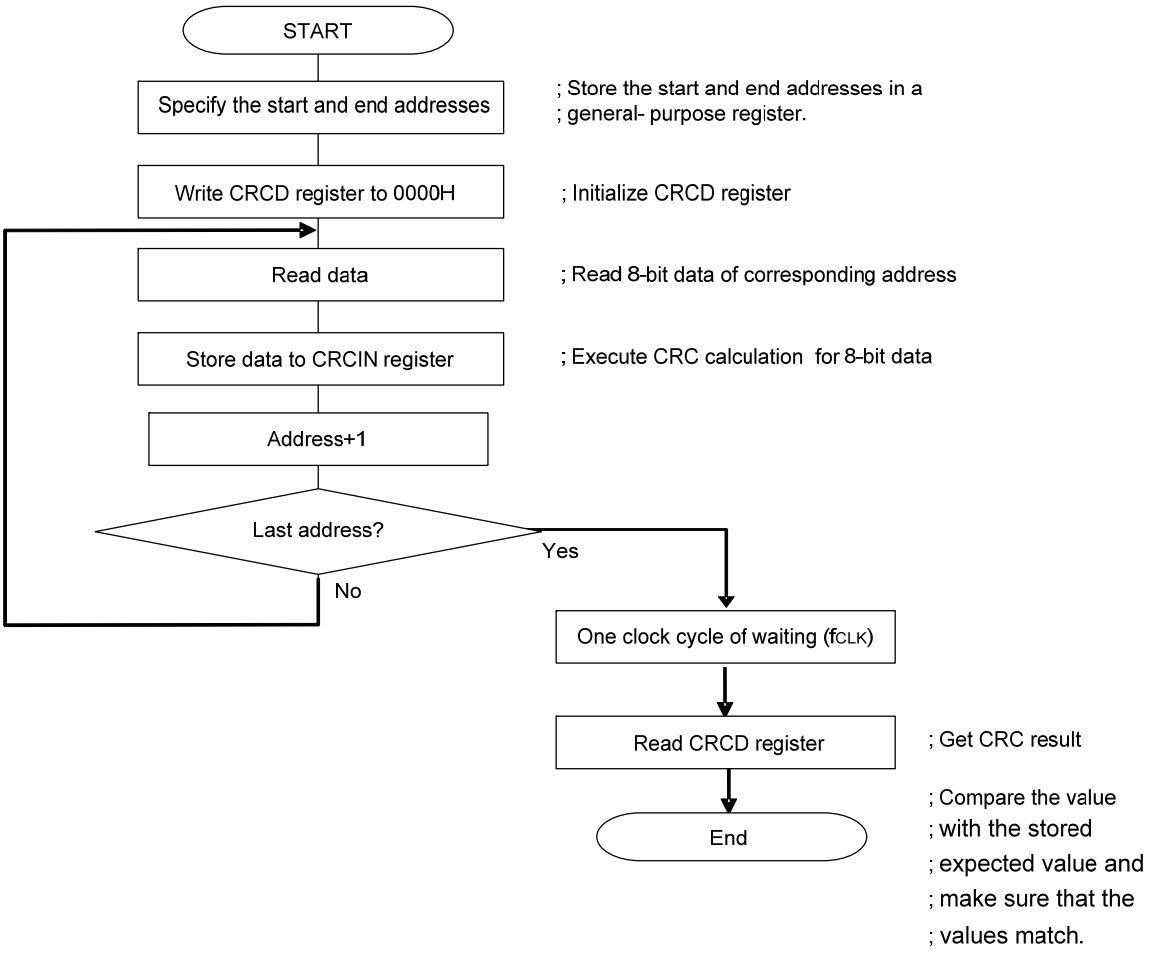
**Notes 1.** Read the value written to CRCD register before writing to CRCIN register.

**2.** If writing and storing operation result to CRCD register conflict, the writing is ignored.

**3.** For CRCD register write when conforming to SENT, write valid data to the lower 4 bits (bits 3 to 0) and write 0 to the other bits.

**<Operation flow>**

**Figure 27-7. CRC Operation Function (General-Purpose CRC)**



### 27.3.3 RAM-ECC function

The RL78/F13 and RL78/F14 have the RAM-ECC function. This function is used to detect erroneous data (bit errors), generate interrupt requests, and retain the addresses of bit errors. If only one bit is in error, the data are corrected.

**Caution The RAM-ECC function is disabled during on-chip debugging. Therefore, do not use the ECC test mode to check the on-chip debugging operation. Even if the ECC test mode is used, bit errors are not detected, error addresses are not stored, or an interrupt is not generated. In addition, even if the bit error is 1 bit, the data is not corrected.**

**<Control register>**

| Register Name | Description | Access Size |
|---|---|---|
| ERADR | Error address store register | 16 bits |
| ECCIER | 1-bit error detection interrupt enable register | 8 bits |
| ECCER | Bit error detection register | 8 bits |
| ECCTPR | ECC test protect register | 8 bits |
| ECCTMDR | ECC test mode register | 8 bits |
| ECCDWRVR | Write data inversion register | 16 bits |

**(1) Error address store register (ERADR)**

**Figure 27-8. Format of Error Address Store Register (ERADR)**

Address: F0200H After reset: 00H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ERADR | ERAD | | | | | | | | | | | | | | | |

| ERAD | Bit error address |
|---|---|
| 0000H to FFFFH | Address when a bit error interrupt request is generated |

**Cautions 1. Access the ERADR register in word units.**

**2. The register value is updated each time a bit error interrupt request is generated.**

**(2) 1-bit error detection interrupt enable register (ECCIER)**

**Figure 27-9. Format of 1-bit Error Detection Interrupt Enable Register (ECCIER)**

Address: F0202H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ECCIER | - | - | - | - | - | - | - | IEN |

| IEN | 1-bit error detection interrupt enable bit |
|---|---|
| 0 | Interrupt disabled |
| 1 | Interrupt enabled |

**Cautions 1. Bits 1 to 7 of the ECCIER register are always read as 0. The write value should always be 0.**

<R> **2. INTRAM interrupt request occurs regardless of the value of ECCIER on two bits error.**

### (3) Bit error detection register (ECCER)

**Figure 27-10. Format of Bit Error Detection Register (ECCER)**

Address: F0203H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ECCER | - | - | - | - | - | - | - | DBERR |

| DBERR | Bit error detection flag |
|---|---|
| 0 | A 1-bit error detected. |
| 1 | A 2-bit error detected. |

**Cautions 1. The DBERR bit is cleared to 0 by writing 0.**

**2. If setting to 1 due to bit error detection and clearing to 0 by the CPU occur simultaneously, setting to 1 due to bit error detection has a priority.**

**3. If a bit error detection interrupt request (INTRAM) is not generated, the DBERR value is invalid.**

### (4) ECC test protect register (ECCTPR)

This register is used to prevent accidentally changing the setting of the ECCTMDR register to trigger entry to the ECC test mode.

Writing a value other than 07H prevents changes to the value of the ECCTMDR register

**Figure 27-11. Format of ECC Test Protect Register (ECCTPR)**

Address: F0204H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ECCTPR | - | - | - | - | - | TPR2 | TPR1 | TPR0 |

| TPR2 to TPR0 | ECC test protect bits |
|---|---|
| Other than 00000111 | Access to the ECCTMDR register is disabled. |
| 00000111 | Access to the ECCTMDR register is enabled. |

**(5) ECC test mode register (ECCTMDR)**

**Figure 27-12. Format of ECC Test Mode Register (ECCTMDR)**

Address: F0205H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ECCTMDR | - | - | - | - | - | TMD2 | TMD1 | TMD0 |

| TMD2 to TMD0 | ECC test mode bits |
|---|---|
| 000 | Normal operating mode |
| 001 | ECC test mode |
| Other than above | Setting prohibited |

**Cautions 1. Set the ECCTPR register to "07H" before accessing the ECCTMDR register.**

**2. Bits 3 to 7 of the ECCTMDR register are always read as 0. The write value should always be 0.**

**(6) Write data inversion register (ECCDWRVR)**

This register is for use in confirming that the ECC is operating correctly by inverting both the parity bit of the write data and the ECC code.

**Figure 27-13. Format of Write Data Inversion Register (ECCWRDR)**

Address: F0206H After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| ECCWRVR | - | - | - | PRTYRV | ECCRV3 | ECCRV2 | ECCRV1 | ECCRV0 |

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ECCWRVR | DWRV7 | DWRV6 | DWRV5 | DWRV4 | DWRV3 | DWRV2 | DWRV1 | DWRV0 |

| PRTYRV | Parity inversion bit |
|---|---|
| 0 | Parity bit not inverted. |
| 1 | Parity bit inverted. |

| ECCRV3 | ECC code inversion bit 3 |
|---|---|
| 0 | Bit 3 of ECC code not inverted. |
| 1 | Bit 3 of ECC code inverted. |

| ECCRV2 | ECC code inversion bit 2 |
|---|---|
| 0 | Bit 2 of ECC code not inverted. |
| 1 | Bit 2 of ECC code inverted. |

| ECCRV1 | ECC code inversion bit 1 |
|---|---|
| 0 | Bit 1 of ECC code not inverted. |
| 1 | Bit 1 of ECC code inverted. |

| ECCRV0 | ECC code inversion bit 0 |
|---|---|
| 0 | Bit 0 of ECC code not inverted. |
| 1 | Bit 0 of ECC code inverted. |

| DWRV7 | Write data inversion bit 7 |
|---|---|
| 0 | Bit 7 of write data not inverted. |
| 1 | Bit 7 of write data inverted. |

| DWRV6 | Write data inversion bit 6 |
|---|---|
| 0 | Bit 6 of write data not inverted. |
| 1 | Bit 6 of write data inverted. |

| DWRV5 | Write data inversion bit 5 |
|---|---|
| 0 | Bit 5 of write data not inverted. |
| 1 | Bit 5 of write data inverted. |

| DWRV4 | Write data inversion bit 4 |
|---|---|
| 0 | Bit 4 of write data not inverted. |
| 1 | Bit 4 of write data inverted. |

| DWRV3 | Write data inversion bit 3 |
|---|---|
| 0 | Bit 3 of write data not inverted. |
| 1 | Bit 3 of write data inverted. |

| DWRV2 | Write data inversion bit 2 |
|---|---|
| 0 | Bit 2 of write data not inverted. |
| 1 | Bit 2 of write data inverted. |

| DWRV1 | Write data inversion bit 1 |
|---|---|
| 0 | Bit 1 of write data not inverted. |
| 1 | Bit 1 of write data inverted. |

| DWRV0 | Write data inversion bit 0 |
|---|---|
| 0 | Bit 0 of write data not inverted. |
| 1 | Bit 0 of write data inverted. |

**Cautions 1. Access the ECCDWRVR register in word units.**

**2. Bits 13 to 15 of the ECCDWRVR register are always read as 0. The write value should always be 0.**

**3. All data written to the RAM, including data written to the stack, is inverted. Therefore, all peripheral functions that might rewrite the RAM must be stopped before a write data inversion bit is set. Do not set a write data inversion bit during OCD.**

<Bit error detection interrupt>

When a bit error is detected, an interrupt request signal (INTRAM) is generated, and the address of the bit error is held in the error address store register (ERADR). If the bit error is 2 bits, the bit error detection flag (DBERR) in the bit error detection register (ECCER) is set to 1.

The 1-bit error detection interrupt enable register (ECCIER) can be used to specify whether to output or not an interrupt request signal when the bit error is 1 bit.

Since the CPU of the RL78 pre-reads the instruction code, RAM fetch area + 10 bytes should be initialized to perform RAM fetch.

Even when a bit error is detected by reading instruction code, an interrupt request is not generated. Thus, the address that causes the bit error cannot be known.

<ECC test function>

The following two modes can be selected by the ECC test mode register (ECCTMDR).

- Normal operating mode
- Test mode (bit error correction function test)

The ECC test mode register should be accessed after the protection by the ECC test protect register (ECCTPR) is cancelled.

Inverting the bit may significantly affect operation of the stack. The bit must thus only be inverted at times such as power-on so that it has no effect on the application.

For data read from the RAM, the existence of a bit error is detected in the 8-bit read data, 4-bit ECC code, and 1-bit parity bit.

If a bit error exists, an interrupt request is output and the address of the bit error is stored in the register. If the bit error is 1 bit, the data is corrected.

(a) Normal operating mode

For data write, a 4-bit ECC code is generated using 8-bit write data, and a 1-bit parity bit is generated using the write data and the ECC code. The generated data is written to the RAM as 13-bit data.

For data read, the existence of a bit error is detected in the 8-bit read data, 4-bit ECC code, and 1-bit parity bit. If the bit error is 1 bit, the data is corrected and then read.

(b) Test mode (bit error correction function test)

For data write, an ECC code is generated using write data, and a parity bit is generated using the write data and the ECC code. A given bit value of the 13-bit write data is inverted by the write data inversion register (ECCDWRVR), and the data is written to the RAM.

For data read, the existence of a bit error is detected in the read data, ECC code, and parity bit. If the bit error is 1 bit, the data is corrected and then read.

### 27.3.4 CPU stack pointer monitor function

The CPU stack pointer monitor is used to detect overflows and underflows of the stack pointer and to generate interrupts in response.

<R> **Caution The CPU stack pointer monitor function is disabled during on-chip debugging.**

**<Configuration>**

This function has the following functions.

- SP overflow/underflow detection function
- SP overflow/underflow interrupt output function

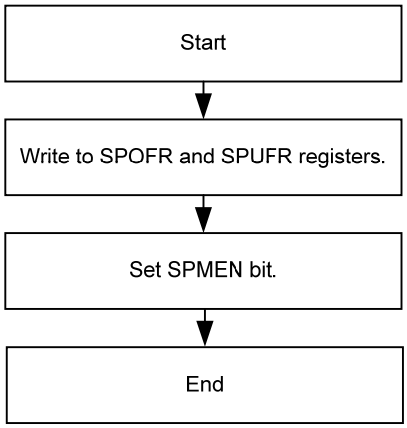
When the SPM enable bit (SPMEN) is 1, an interrupt signal (INTSPM) is generated if the monitored stack pointer value is greater than the specified SFR value (SPOFR) or smaller than the specified SFR value (SPUFR).

When the SPM enable bit (SPMEN) is 1, writing to the SPOFR and SPUFR registers is invalid.

**<Register setting method>**

Figure 27-14 shows the register setting method of this function.

**Figure 27-14. Register Setting Flow**




[Standard usage]

1. Write the initial value to the SPOFR and SPUFR registers.
2. Set the SPMEN bit in the SPMCTRL register.

**<Control register>**

| Register Name | Description | Access Size |
|---|---|---|
| SPMCTRL | SPM control register | 8 bits |
| SPOFR | SP overflow address setting register | 16 bits |
| SPUFR | SP underflow address setting register | 16 bits |

**Remarks 1.** If the overflow or underflow state is retained, another overflow or underflow will not be detected. After an overflow or underflow was detected, reset the stack pointer to a value within the range of monitoring.

**2.** If an overflow or underflow interrupt request is received, the value obtained by subtracting 4 from the current value of the stack pointer is always used for saving the interrupt.

**(1) SPM control register (SPMCTRL)**

**Figure 27-15. Format of SPM Control Register (SPMCTRL)**

Address: F00D8H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| SPMCTRL | SPMEN | - | - | - | - | - | - | - |

| SPMEN | Stack pointer monitor SFR write enable/disable |
|---|---|
| 0 | Stack pointer monitoring disabled. |
| 1 | Stack pointer monitoring enabled. |

**Caution Writing 1 to the SPMEN bit is only valid, and writing 0 after setting SPMEN to 1 is invalid.**

**(2) SP overflow address setting register (SPOFR)**

**Figure 27-16. Format of SP Overflow Address Setting Register (SPOFR)**

Address: F00DAH After reset: FFFEH R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPOFR | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |

| - | Stack pointer overflow address setting |
|---|---|
| 0 | Stack pointer overflow address |
| 1 | |

**Cautions 1. The lowest bit is fixed to 0.**

**2. If the values of bits 15 to 1 in stack pointer are greater than the specified values of bits 15 to 1 in SPOFR, an interrupt signal (INTSPM) is generated.**
**Stack pointer > SPOFR: INTSPM interrupt signal is generated.**

**3. When SPMEN = 1, writing to SPOFR is invalid.**

**(3) SP underflow address setting register (SPUFR)**

**Figure 27-17. Format of SP Underflow Address Setting Register (SPUFR)**

Address: F00DCH After reset: 0000H R/W

| Symbol | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPUFR | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |

| | |
|---|---|
| - | Stack pointer underflow address setting |
| 0 | Stack pointer underflow address |
| 1 | |

**Cautions 1. The lowest bit is fixed to 0.**

**2. If the values of bits 15 to 1 in stack pointer are smaller than the specified values of bits 15 to 1 in SPUFR, an interrupt signal (INTSPM) is generated.**
**Stack pointer < SPUFR: INTSPM interrupt signal is generated.**

**3. When SPMEN = 1, writing to SPUFR is invalid.**

### 27.3.5 Clock monitor

The clock monitor samples the main system clock ($f_{MAIN}$) and main system/PLL select clock ($f_{MP}$) by using the low-speed on-chip oscillator. When oscillation of the main system clock stops, a reset request signal (RESCLM) is generated. When the main system/PLL select clock ($f_{MP}$) stops, the clock through mode is forcibly selected and SELPLLS is cleared (but SELPLL is not).

At the same time, an interrupt request signal (INTCLM) is generated.

#### (1) Configuration

Figure 27-18 shows a block diagram of the clock monitor.

**Figure 27-18. Block Diagram of Clock Monitor**

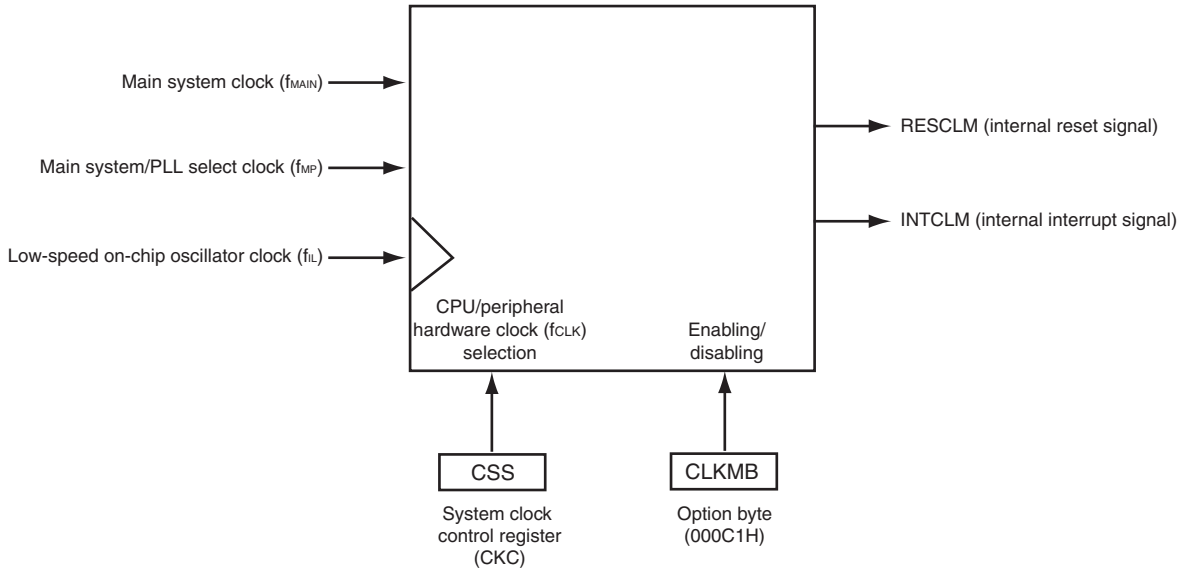



**Table 27-1. Operating State of Clock Monitor**

| Operating State of Clock Monitor | | State of Clock Monitor |
|---|---|---|
| $f_{CLK} = f_{SUB}$ or $f_{IL}$ | | Stopped |
| $f_{CLK} = f_{MP}/2^N$ | STOP mode | Stopped |
| | SNOOZE mode | Stopped |
| | Oscillation stabilization time after setting of the MCM0 bit | Stopped |
| | CLKMB = 1 | Stopped |
| | CLKMB = 0 | Operating |

**(2) Starting and stopping of operation**

Bit 4 (CLKMB) of the option byte (000C1H) should be set to 0 to enable operation of the clock monitor.

After the oscillation of the low-speed on-chip oscillator is set, the clock monitor starts operation.

The clock monitor automatically stops operating under the following conditions.

- In STOP mode
- In SNOOZE mode
- During counting of the oscillation stabilization time after STOP mode was released
- When the CPU/peripheral hardware clock frequency ($f_{CLK}$) is equal to the subsystem clock ($f_{SUB}$) or low-speed on-chip oscillator clock ($f_{IL}$)
- When the sampling clock is stopped (low-speed on-chip oscillator is stopped)
- When bit 4 (CLKMB) of the option byte (000C1H) is 1

**(3) Cautions for use**

When entering the STOP mode by stopping the PLL clock during the operation of the clock monitor, set bit 0 (PLLON) in the PLL control register (PLLCTL) before executing the STOP instruction.

### 27.3.6 RAM guard function

This RAM guard function is used to protect data in the specified memory space.

If the RAM guard function is specified, writing to the specified RAM space is disabled, but reading from the space can be carried out as usual. The area used as the stack must not be a target of the RAM guard function.

**<Control register>**

**(1) Invalid memory access detection control register (IAWCTL)**

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GRAM1 and GRAM0 bits are used in RAM guard function.

The IAWCTL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 27-19. Format of Invalid Memory Access Detection Control Register (IAWCTL)**

Address: F0078H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| IAWCTL | IAWEN | 0 | GRAM1 | GRAM0 | 0 | GPORT | GINT | GCSC |

| GRAM1 | GRAM0 | RAM guard space Note |
|---|---|---|
| 0 | 0 | Disabled. RAM can be written to. |
| 0 | 1 | The 128 bytes starting at the lower RAM address |
| 1 | 0 | The 256 bytes starting at the lower RAM address |
| 1 | 1 | The 512 bytes starting at the lower RAM address |

**Note** Do not set the RAM guard space to an area exceeding the size of the RAM of the product.

### 27.3.7 SFR guard function

This SFR guard function is used to protect data in the control registers used by the port function, interrupt function, clock control function, and voltage detector.

If the SFR guard function is specified, writing to the specified SFRs is disabled, but reading from the SFRs can be carried out as usual.

**<Control register>**

**(1) Invalid memory access detection control register (IAWCTL)**

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GPORT, GINT and GCSC bits are used in SFR guard function.

The IAWCTL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 27-20. Format of Invalid Memory Access Detection Control Register (IAWCTL)**

Address: F0078H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| IAWCTL | IAWEN | 0 | GRAM1 | GRAM0 | 0 | GPORT | GINT | GCSC |

| GPORT | Control registers of port function guard |
|---|---|
| 0 | Disabled. Control registers of port function can be read or written to. |
| 1 | Enabled. Writing to control registers of port function is disabled. Reading is enabled.<br>[Guarded SFR] PMxx, PUxx, PIMxx, POMxx, PMCxx, PITHLxx, ADPC, PIOR Note |

| GINT | Registers of interrupt function guard |
|---|---|
| 0 | Disabled. Registers of interrupt function can be read or written to. |
| 1 | Enabled. Writing to registers of interrupt function is disabled. Reading is enabled.<br>[Guarded SFR] IFxx, MKxx, PRxx, EGPx, EGNx |

| GCSC | Control registers of clock control function and voltage detector guard |
|---|---|
| 0 | Disabled. Control registers of clock control function and voltage detector can be read or written to. |
| 1 | Enabled. Writing to control registers of clock control function and voltage detector is disabled. Reading is enabled.<br>[Guarded SFR] CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS, CANCKSEL, LINCKSEL, CKSEL, PLLCTL, MDIV, RTCCL, POCRES, STPSTC |

<R> **Note** Pxx (Port register) is not guarded.

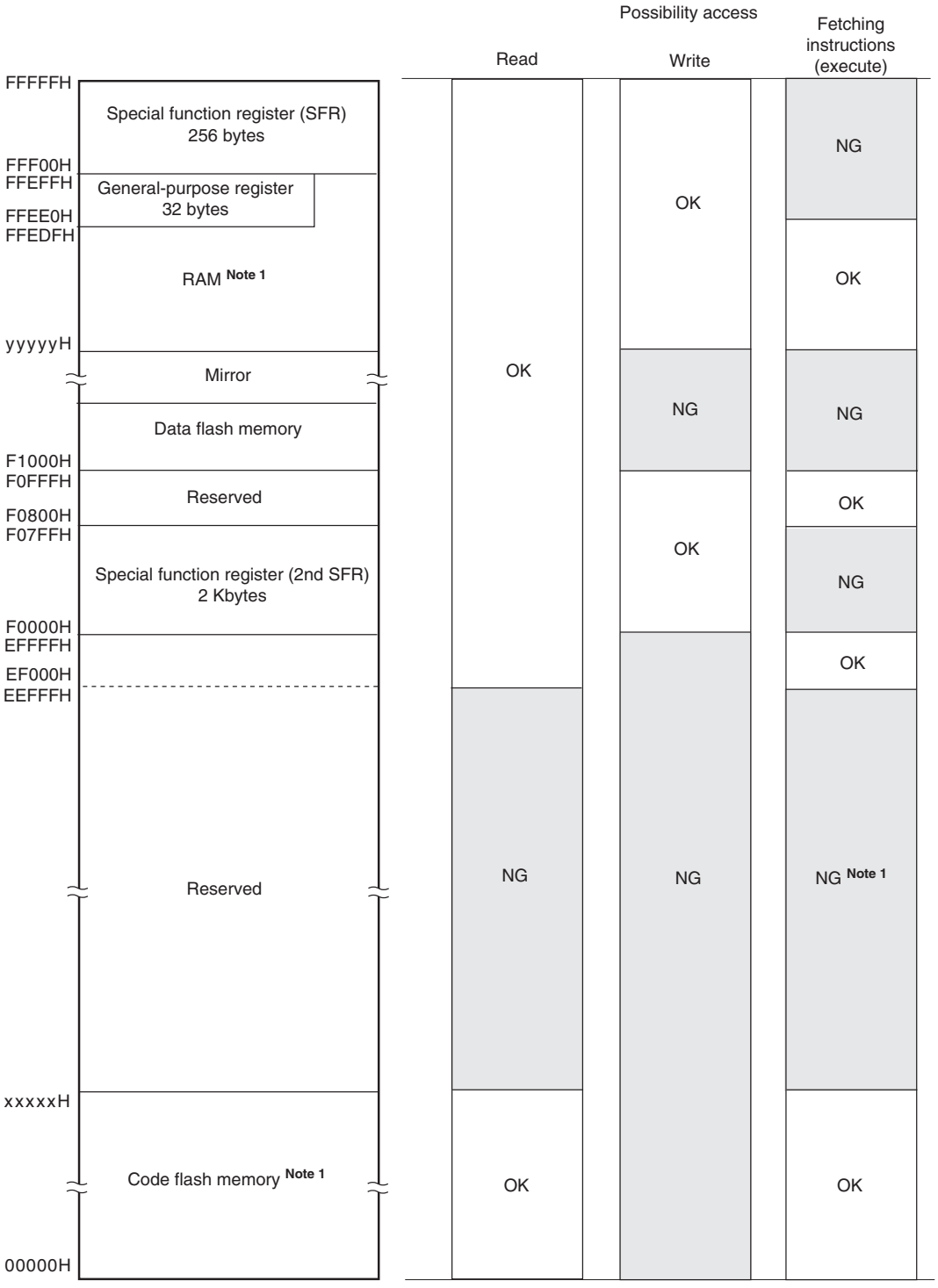
### 27.3.8 Invalid memory access detection function

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly.

The illegal memory access detection function triggers a reset if a memory space specified as access-prohibited is accessed.

The illegal memory access detection function applies to the areas indicated by NG in Figure 27-21.

**Figure 27-21. Invalid access detection area**




(**Notes** are listed on the next page.)

**Notes 1.** Code flash memory and RAM address of each product are as follows.

- RL78/F13 (LIN incorporated)

| ROM size | Code flash memory (00000H to xxxxxH) |
|---|---|
| 16 Kbytes | 16384 × 8 bits (00000H to 03FFFH) Note 2 |
| 32 Kbytes | 32768 × 8 bits (00000H to 07FFFH) Note 2 |
| 48 Kbytes | 49152 × 8 bits (00000H to 0BFFFH) Note 2 |
| 64 Kbytes | 65536 × 8 bits (00000H to 0FFFFH) |
| 96 Kbytes | 98304 × 8 bits (00000H to 17FFFH) Note 3 |
| 128 Kbytes | 131072 × 8 bits (00000H to 1FFFFH) |

| RAM size | RAM(yyyyyH to FFEFFH) |
|---|---|
| 1 Kbyte | 1024 × 8 bits (FFB00H to FFEFFH) |
| 2 Kbytes | 2048 × 8 bits (FF700H to FFEFFH) |
| 3 Kbytes | 3072 × 8 bits (FF300H to FFEFFH) |
| 4 Kbytes | 4096 × 8 bits (FEF00H to FFEFFH) |
| 6 Kbytes | 6144 × 8 bits (FE700H to FFEFFH) |
| 8 Kbytes | 8192 × 8 bits (FDF00H to FFEFFH) |

- RL78/F13 (CAN and LIN incorporated)

| ROM size | Code flash memory (00000H to xxxxxH) |
|---|---|
| 32 Kbytes | 32768 × 8 bits (00000H to 07FFFH) Note 2 |
| 48 Kbytes | 49152 × 8 bits (00000H to 0BFFFH) Note 2 |
| 64 Kbytes | 65536 × 8 bits (00000H to 0FFFFH) |
| 96 Kbytes | 98304 × 8 bits (00000H to 17FFFH) Note 3 |
| 128 Kbytes | 131072 × 8 bits (00000H to 1FFFFH) |

| RAM size | RAM (yyyyyH-FFEFFH) |
|---|---|
| 2 Kbytes | 2048 × 8 bits (FF700H to FFEFFH) |
| 3 Kbytes | 3072 × 8 bits (FF300H to FFEFFH) |
| 4 Kbytes | 4096 × 8 bits (FEF00H to FFEFFH) |
| 6 Kbytes | 6144 × 8 bits (FE700H to FFEFFH) |
| 8 Kbytes | 8192 × 8 bits (FDF00H to FFEFFH) |

- RL78/F14

| ROM size | Code flash memory (00000H to xxxxxH) |
|---|---|
| 48 Kbytes | 49152 × 8 bits (00000H to 0BFFFH) Note 2 |
| 64 Kbytes | 65536 × 8 bits (00000H to 0FFFFH) |
| 96 Kbytes | 98304 × 8 bits (00000H to 17FFFH) Note 3 |
| 128 Kbytes | 131072 × 8 bits (00000H to 1FFFFH) |
| 192 Kbytes | 196608 × 8 bits (00000H to 2FFFFH) |
| 256 Kbytes | 262144 × 8 bits (00000H to 3FFFFH) |

| RAM size | RAM (yyyyyH to FFEFFH) |
|---|---|
| 4 Kbytes | 4096 × 8 bits (FEF00H to FFEFFH) |
| 6 Kbytes | 6144 × 8 bits (FE700H to FFEFFH) |
| 8 Kbytes | 8192 × 8 bits (FDF00H to FFEFFH) |
| 10 Kbytes | 10240 × 8 bits (FD700H to FFEFFH) |
| 16 Kbytes | 16384 × 8 bits (FBF00H to FFEFFH) |
| 20 Kbytes | 20480 × 8 bits (FAF00H to FFEFFH) |

**2.** Fetching of an instruction (for execution) by illegal access to a location in the area from xxxxxH to 0FFFFH leads to the generation of a reset due to the execution of an illegal instruction rather than being handled as illegal memory access.

**3.** Fetching of an instruction (for execution) by illegal access to a location in the area from xxxxxH to 1FFFFH leads to the generation of a reset due to the execution of an illegal instruction rather than being handled as illegal memory access.

**<Control register>**

- **Invalid memory access detection control register (IAWCTL)**

  This register is used to control the detection of invalid memory access and RAM/SFR guard function.

  IAWEN bit is used in invalid memory access detection function.

  The IAWCTL register can be set by an 8-bit memory manipulation instruction.

  Reset signal generation clears this register to 00H.

**Figure 27-22. Format of Invalid Memory Access Detection Control Register (IAWCTL)**

Address: F0078H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| IAWCTL | IAWEN | 0 | GRAM1 | GRAM0 | 0 | GPORT | GINT | GCSC |

| IAWEN Note | Control of invalid memory access detection |
|---|---|
| 0 | Disable the detection of invalid memory access. |
| 1 | Enable the detection of invalid memory access. |

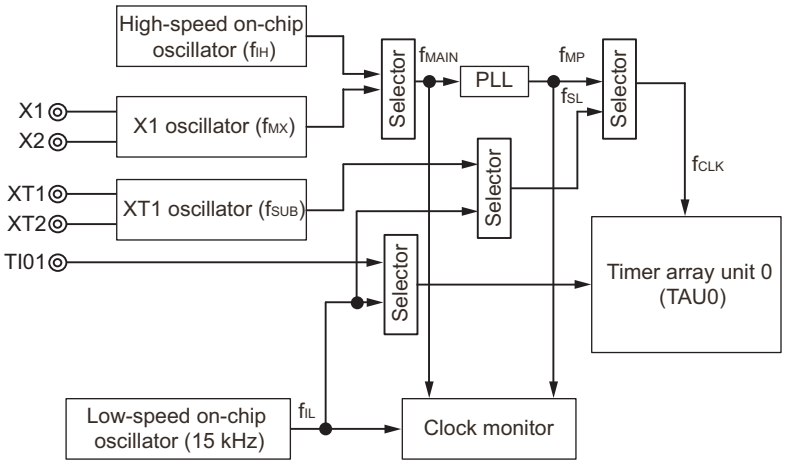
**Remark** By specifying WDTON = 1 for the option byte, the invalid memory access function is always enabled regardless of the setting for the IAWEN bit. (For details, see **CHAPTER 29 OPTION BYTE**.)

**Note** Only writing 1 to the IAWEN bit is valid, and writing 0 after setting the IAWEN bit to 1 is invalid.

### 27.3.9 Frequency detection function

The frequency detection function can detect whether the clock is operating on an abnormal frequency by comparing the high-speed on-chip oscillator clock, external X1 oscillation clock, or PLL clock with the low-speed on-chip oscillator clock (15 kHz).

**Figure 27-23. Configuration of Frequency Detection Function**




**<Operational overview>**

Whether the clock frequency is correct or not can be judged by measuring the pulse width under the following conditions:

- The high-speed on-chip oscillator clock ($f_{IH}$), external X1 oscillation clock ($f_{MX}$), or PLL clock ($f_{PLL}$) is selected as the CPU/peripheral hardware clock ($f_{CLK}$).
- The low-speed on-chip oscillator clock ($f_{IL}$: 15 kHz) is selected as the timer input for channel 1 of timer array unit 0 (TAU0).

If pulse width measurement results in an abnormal value, it can be concluded that the clock frequency is abnormal.
For how to execute pulse width measurement, see **6.7.4 Operation as input pulse interval measurement**.

**<Control register>**

- **Timer input select register 0 (TIS0)**

  This register is used to select the timer input of channel 1.

  By selecting the low-speed on-chip oscillator clock for the timer input, its pulse width can be measured to determine whether the proportional relationship between the low-speed on-chip oscillator clock and the timer operation clock is correct.

  The TIS0 register can be set by an 8-bit memory manipulation instruction.

  Reset signal generation clears this register to 00H.

**Figure 27-24. Format of Timer Input Select Register 0 (TIS0)**

Address: F0074H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| TIS0 | TIS07 Note 1 | TIS06 Note 1 | 0 | TIS04 Note 2 | 0 | TIS02 | TIS01 | TIS00 |

| TIS07 Note 1 | Selection of timer input used with channel 3 of timer array unit 0 |
|---|---|
| 0 | Input signal of timer input pin (TI03) |
| 1 | Event input signal from ELC Note 3 |

| TIS06 Note 1 | Selection of timer input used with channel 2 of timer array unit 0 |
|---|---|
| 0 | Input signal of timer input pin (TI02) |
| 1 | Event input signal from ELC Note 3 |

| TIS04 Note 2 | Selection of timer input used with channel 0 of timer array unit 0 |
|---|---|
| 0 | Input signal of timer input pin (TI00) |
| 1 | Event input signal from ELC Note 3 |

| TIS02 | TIS01 | TIS00 | Selection of timer input used with channel 1 of timer array unit 0 |
|---|---|---|---|
| 0 | 0 | 0 | Input signal of timer input pin (TI01) |
| 0 | 0 | 1 | Event input signal from ELC Note 3 |
| 0 | 1 | 0 | Input signal of timer input pin (TI01) |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | Low-speed on-chip oscillator clock ($f_{IL}$) |
| 1 | 0 | 1 | Sub/low-speed on-chip oscillator select clock ($f_{SL}$) |
| Other than above | | | Setting prohibited |

**Notes 1.** Provided only in products of group E. When writing data to the timer input select register 0 (TIS0) of other products, always write 0.

**2.** Provided only in products of groups D and E. When writing data to the timer input select register 0 (TIS0) of other products, always write 0.

**3.** Provided only in products of groups D and E. Do not set a value for any other products.

(**Cautions** are provided on the next page.)

**Cautions 1. When selecting an event input signal from the ELC using timer input select register 0 (TIS0), select $f_{CLK}$ using timer clock select register 0 (TPS0).**

**2. Do not change the select bit of the timer input while inputting data to the TImn pin (m = 0, 1; n = 0 to 7).**

**3. Each of the high-level and low-level widths of the timer input to be selected should be ($1/f_{MCK}$ + 10 ns) or more. So, the TIS02 bit cannot be set to 1 when $f_{SL}$ is selected as $f_{CLK}$ (the CSS bit in the CKC register is set to 1).**

### 27.3.10 A/D test function

The A/D test function is used to check whether the A/D converter is operating normally by executing A/D conversions of an internal voltage of 0 V, the $AV_{REF}$ voltage, and the internal reference voltage (1.45 V).
The below procedure can be used to confirm selection of the analog multiplexer and that wiring is not disconnected.

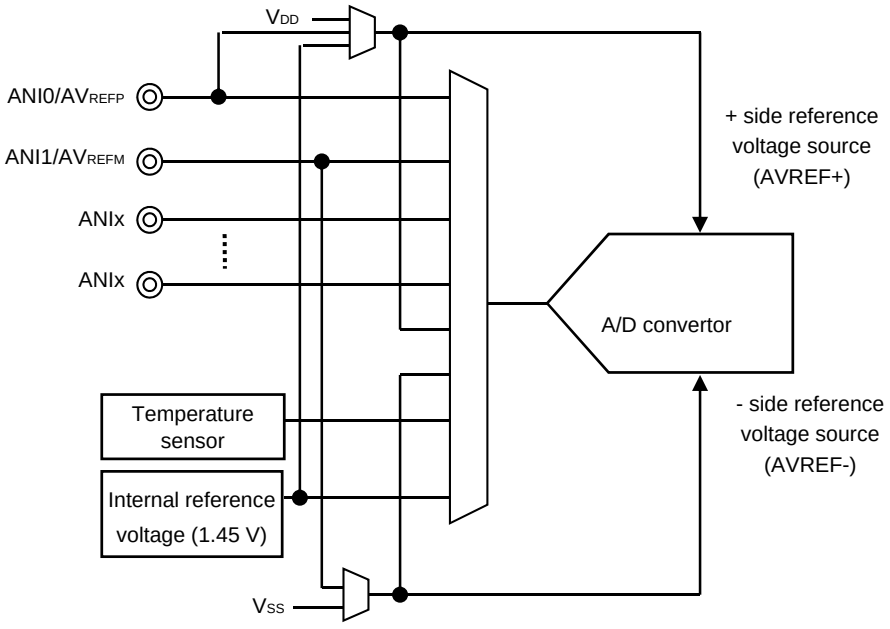
<1> Perform A/D conversion of the voltage on the ANIx pin (result of conversion 1).
<2> Select $AV_{REFM}$ with the ADTES register, perform A/D conversion, and set the potential difference between the two terminals of the sampling capacitor of the A/D converter to 0 V.
<3> Perform A/D conversion of the voltage on the ANIx pin (result of conversion 2).
<4> Select $AV_{REFP}$ with the ADTES register, perform A/D conversion, and set the potential difference between the two terminals of the sampling capacitor of the A/D converter to $AV_{REF}$.
<5> Perform A/D conversion of the voltage on the ANIx pin (result of conversion 3).
<6> Confirm that result of conversion 1 = result of conversion 2 = result of conversion 3.

The above procedure can be used to confirm selection of the analog multiplexer and that wiring is not disconnected.

**Remarks 1.** When the analog input voltage varies during conversion in steps <1> to <5>, use a different method to check the analog multiplexer.

**2.** The result of conversion will include error. Therefore, take the error into consideration when comparing the results of conversion.

**Figure 27-25. Configuration of A/D Test Function**



**<Control register>**

**(1) A/D test register (ADTES)**

This register is used to select the A/D converter's positive reference voltage $AV_{REFP}$, the A/D converter's negative reference voltage $AV_{REFM}$, or the analog input channel (ANIxx) as the target of A/D conversion.

When using the A/D test function, specify the following settings:

- Select $AV_{REFM}$ as the target of A/D conversion when converting the internal 0 V.
- Select $AV_{REFP}$ as the target of A/D conversion when converting $AV_{REF}$.

The ADTES register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 27-26. Format of A/D Test Register (ADTES)**

Address: F0013H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADTES | 0 | 0 | 0 | 0 | 0 | 0 | ADTES1 | ADTES0 |

| ADTES1 | ADTES0 | A/D conversion target |
|---|---|---|
| 0 | 0 | ANIxx/temperature sensor output/ internal reference voltage output (1.45 V) (This is specified using the analog input channel specification register (ADS).) |
| 1 | 0 | $AV_{REFM}$ |
| 1 | 1 | $AV_{REFP}$ |
| Other than the above | | Setting prohibited |

**(2) Analog input channel specification register (ADS)**

This register specifies the input channel of the analog voltage to be A/D converted.

The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 27-27. Format of Analog Input Channel Specification Register (ADS) (1/2)**

Address: FFF31H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADS | ADISS | 0 | 0 | ADS4 | ADS3 | ADS2 | ADS1 | ADS0 |

○ Select mode (ADMD = 0)

| ADISS | ADS4 | ADS3 | ADS2 | ADS1 | ADS0 | Analog input channel | Input source |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | ANI0 | P33/AV$_{REFP}$/ANI0 |
| 0 | 0 | 0 | 0 | 0 | 1 | ANI1 | P34/AV$_{REFM}$/ANI1 |
| 0 | 0 | 0 | 0 | 1 | 0 | ANI2 | P80/ANI2/ANO0 |
| 0 | 0 | 0 | 0 | 1 | 1 | ANI3 | P81/ANI3/IVCMP00 |
| 0 | 0 | 0 | 1 | 0 | 0 | ANI4 | P82/ANI4/IVCMP01 |
| 0 | 0 | 0 | 1 | 0 | 1 | ANI5 | P83/ANI5/IVCMP02 |
| 0 | 0 | 0 | 1 | 1 | 0 | ANI6 | P84/ANI6/IVCMP03 |
| 0 | 0 | 0 | 1 | 1 | 1 | ANI7 | P85/ANI7/IVREF0 |
| 0 | 0 | 1 | 0 | 0 | 0 | ANI8 | P86/ANI8 |
| 0 | 0 | 1 | 0 | 0 | 1 | ANI9 | P87/ANI9 |
| 0 | 0 | 1 | 0 | 1 | 0 | ANI10 | P90/ANI10 |
| 0 | 0 | 1 | 0 | 1 | 1 | ANI11 | P91/ANI11 |
| 0 | 0 | 1 | 1 | 0 | 0 | ANI12 | P92/ANI12 |
| 0 | 0 | 1 | 1 | 0 | 1 | ANI13 | P93/ANI13 |
| 0 | 0 | 1 | 1 | 1 | 0 | ANI14 | P94/ANI14 |
| 0 | 0 | 1 | 1 | 1 | 1 | ANI15 | P95/ANI15 |
| 0 | 1 | 0 | 0 | 0 | 0 | ANI16 | P96/ANI16 Note1 |
| 0 | 1 | 0 | 0 | 0 | 1 | ANI17 | P97/ANI17 Note2 |
| 0 | 1 | 0 | 0 | 1 | 0 | ANI18 | P100/ANI18 |
| 0 | 1 | 0 | 0 | 1 | 1 | ANI19 | P101/ANI19 |
| 0 | 1 | 0 | 1 | 0 | 0 | ANI20 | P102/ANI20 |
| 0 | 1 | 0 | 1 | 0 | 1 | ANI21 | P103/ANI21 |
| 0 | 1 | 0 | 1 | 1 | 0 | ANI22 | P104/ANI22 |
| 0 | 1 | 0 | 1 | 1 | 1 | ANI23 | P105/ANI23 |
| 0 | 1 | 1 | 0 | 0 | 0 | ANI24 | P125/ANI24 |
| 0 | 1 | 1 | 0 | 0 | 1 | ANI25 | P120/ANI25 |
| 0 | 1 | 1 | 0 | 1 | 0 | ANI26 | P96/ANI26 Note3, P70/ANI26 Note4 |
| 0 | 1 | 1 | 0 | 1 | 1 | ANI27 | P97/ANI27 Note5, P71/ANI27 Note6 |
| 0 | 1 | 1 | 1 | 0 | 0 | ANI28 | P72/ANI28 |
| 0 | 1 | 1 | 1 | 0 | 1 | ANI29 | P73/ANI29 |
| 0 | 1 | 1 | 1 | 1 | 0 | ANI30 | P74/ANI30 |
| 1 | 1 | 1 | 1 | 1 | 1 | Setting prohibited | |
| 1 | 0 | 0 | 0 | 0 | 0 | – | Temperature sensor output |
| 1 | 0 | 0 | 0 | 0 | 1 | – | Internal reference voltage output (1.45 V) |

(**Notes** are listed on the next page.)

**Notes 1.** RL78/F14 products with 64 or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory, or RL78/F14 products with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory.

**2.** RL78/F14 products with 80 pins and 128 Kbytes to 256 Kbytes of code flash memory, or RL78/F14 products with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory.

**3.** RL78/F14 products with 64 or 80 pins and 64 Kbytes to 96 Kbytes of code flash memory, RL78/F13 (CAN and LIN incorporated) products with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory,
or RL78/F13 (CAN and LIN incorporated) products with 64 pins and 32 Kbytes to 128 Kbytes of code flash memory,
or RL78/F13 (LIN incorporated) products with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory,
or RL78/F13 (LIN incorporated) products with 64 pins and 96 Kbytes to 128 Kbytes of code flash memory.

**4.** RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory, or RL78/F14 products with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory.

**5.** RL78/F14 products with 80 pins and 64 Kbytes to 96 Kbytes of code flash memory, RL78/F13 (CAN and LIN incorporated) or RL78/F13 (LIN incorporated) products with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory.

**6.** RL78/F14 products with 48 or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory, or RL78/F14 products with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory

**Figure 27-27. Format of Analog Input Channel Specification Register (ADS) (2/2)**

Address: FFF31H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADS | ADISS | 0 | 0 | ADS4 | ADS3 | ADS2 | ADS1 | ADS0 |

○ Scan mode (ADMD = 1)

| ADISS | ADS4 | ADS3 | ADS2 | ADS1 | ADS0 | Analog input channel | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Scan 0 | Scan 1 | Scan 2 | Scan 3 |
| 0 | 0 | 0 | 0 | 0 | 0 | ANI0 | ANI1 | ANI2 | ANI3 |
| 0 | 0 | 0 | 0 | 0 | 1 | ANI1 | ANI2 | ANI3 | ANI4 |
| 0 | 0 | 0 | 0 | 1 | 0 | ANI2 | ANI3 | ANI4 | ANI5 |
| 0 | 0 | 0 | 0 | 1 | 1 | ANI3 | ANI4 | ANI5 | ANI6 |
| 0 | 0 | 0 | 1 | 0 | 0 | ANI4 | ANI5 | ANI6 | ANI7 |
| 0 | 0 | 0 | 1 | 0 | 1 | ANI5 | ANI6 | ANI7 | ANI8 |
| 0 | 0 | 0 | 1 | 1 | 0 | ANI6 | ANI7 | ANI8 | ANI9 |
| 0 | 0 | 0 | 1 | 1 | 1 | ANI7 | ANI8 | ANI9 | ANI10 |
| 0 | 0 | 1 | 0 | 0 | 0 | ANI8 | ANI9 | ANI10 | ANI11 |
| 0 | 0 | 1 | 0 | 0 | 1 | ANI9 | ANI10 | ANI11 | ANI12 |
| 0 | 0 | 1 | 0 | 1 | 0 | ANI10 | ANI11 | ANI12 | ANI13 |
| 0 | 0 | 1 | 0 | 1 | 1 | ANI11 | ANI12 | ANI13 | ANI14 |
| 0 | 0 | 1 | 1 | 0 | 0 | ANI12 | ANI13 | ANI14 | ANI15 |
| 0 | 1 | 0 | 0 | 0 | 0 | ANI16 | ANI17 | ANI18 | ANI19 |
| 0 | 1 | 0 | 0 | 0 | 1 | ANI17 | ANI18 | ANI19 | ANI20 |
| 0 | 1 | 0 | 0 | 1 | 0 | ANI18 | ANI19 | ANI20 | ANI21 |
| 0 | 1 | 0 | 0 | 1 | 1 | ANI19 | ANI20 | ANI21 | ANI22 |
| 0 | 1 | 0 | 1 | 0 | 0 | ANI20 | ANI21 | ANI22 | ANI23 |
| Other than the above | | | | | | Setting prohibited | | | |

**Cautions 1. Be sure to clear bits 5 and 6 to 0.**

**2. In port mode registers 3, 7 to 10, and 12 (PM3, PM7 to PM10, PM12), set the ADPC and PMCxx registers to the input mode for pins to be set as analog inputs.**

**3. In the ADS register, do not select a pin which is set as a digital I/O pin in the A/D port configuration register (ADPC).**

**4. In the ADS register, do not select a pin which is set as a digital I/O pin in the port mode control register 7, 9, 12 (PMC7, PMC9, PMC12)**

**5. Only rewrite the value of the ADISS bit while A/D voltage comparator operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).**

**6. If using $AV_{REFP}$ as the + side reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.**

**7. If using $AV_{REFM}$ as the - side reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.**

**8. If ADISS is set to 1, the internal reference voltage (1.45 V) cannot be used for the + side reference voltage source.**

**9. When the CPU enters the STOP mode or HALT mode while being driven by the subsystem/low-speed on-chip oscillator select clock, do not set the ADISS bit to 1. Setting ADISS to 1 increases the value given in 34.3.2 Supply Current Characteristics or 35.3.2 Supply Current Characteristics.**

**10. Ignore the result of conversion if the corresponding ANI pin is not present.**

### 27.3.11 Digital output signal level detection function for I/O ports

By using the digital output signal level detection function for I/O ports, the digital output level of the pin can be read when the port is set to output mode (the PMmn bit in the port mode register (PMm) is 0).

This function allows the output level of the pin to be read even when the PMnm (I/O mode) bit is set to output mode. As a result, the CPU can determine the current output level is a high or low level. For details on the registers to control this function, see **CHAPTER 4 PORT FUNCTIONS**.

**<Control register>**

**(1) Port mode select register (PMS)**

This register is used to select the output level from output latch level or pin output level when the port is output mode in which PMm bit of port mode register (PMm) is 0.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Figure 27-28. Format of Port Mode Select Register (PMS)**

Address: F0077H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| PMS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PMS0 |

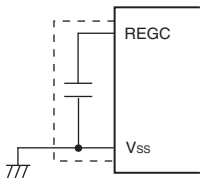
| PMS0 | Method for selecting output level to be read when port is output mode (PMmn = 0) |
|---|---|
| 0 | Pmn register value is read. |
| 1 | Output level of the pin is read. |

**Remark** m = 0, 1, 3 to 10, 12 to 15
n = 0 to 7

# CHAPTER 28 REGULATOR

## 28.1 Regulator Overview

The RL78/F13 and RL78/F14 contain a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to $V_{SS}$ via a capacitor (0.47 to 1 $\mu$F). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.




**Caution Keep the wiring length as short as possible for the broken-line part in the above figure.**

For the regulator output voltage, see **Table 28-1**.

**Table 28-1. Regulator Output Voltage Conditions**

| Mode | Output Voltage | Condition |
|---|---|---|
| High-speed main mode | 1.86 V | In STOP mode |
| | | When the high-speed system clock ($f_{MX}$), the high-speed on-chip oscillator clock ($f_{IH}$), and PLL clock ($f_{PLL}$) are stopped during CPU operation with the subsystem/low-speed on-chip oscillator clock select clock ($f_{SL}$) |
| | | When the high-speed system clock ($f_{MX}$), the high-speed on-chip oscillator clock ($f_{IH}$), and PLL clock ($f_{PLL}$) are stopped during the HALT mode when the CPU operation with the subsystem/low-speed on-chip oscillator clock select clock ($f_{SL}$) has been set |
| | 2.1 V | Other than above (include during OCD mode) **Note** |

**Note** When it shifts to the subsystem/low-speed on-chip oscillator clock select clock operation or STOP mode during the on-chip debugging, the regulator output voltage is kept at 2.1 V (not decline to 1.86 V).

# CHAPTER 29 OPTION BYTE

## 29.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/F13 and RL78/F14 form an option byte area.

Option bytes consist of user option byte (000C0H to 000C2H) and on-chip debug option byte (000C3H).

Upon power application or resetting and starting, an option byte is automatically referenced and a specified function is set. When using the product, be sure to set the following functions by using the option bytes.

To use the boot swap operation during self programming, 000C0H to 000C3H are replaced by 020C0H to 020C3H. Therefore, set the same values as 000C0H to 000C3H to 020C0H to 020C3H.

### 29.1.1 User option byte (000C0H to 000C2H/020C0H to 020C2H)

**(1) 000C0H/020C0H**

- Operation of watchdog timer
  - Operation is stopped or enabled in the HALT, STOP, or SNOOZE mode.
- Setting of interval time of watchdog timer
- Operation of watchdog timer
  - Operation is stopped or enabled.
- Setting of window open period of watchdog timer
- Setting of interval interrupt of watchdog timer
  - Used or not used

**Caution Set the same value as 000C0H to 020C0H when the boot swap operation is used because 000C0H is replaced by 020C0H.**

**(2) 000C1H/020C1H**

- Setting of LVD operation mode
  - Interrupt & reset mode
  - Reset mode
  - Interrupt mode
- Setting of LVD detection level ($V_{LVDH}$, $V_{LVDL}$, $V_{LVD}$)
- Operation of clock monitor
  - Operation is stopped or enabled.

**Caution Set the same value as 000C1H to 020C1H when the boot swap operation is used because 000C1H is replaced by 020C1H.**

**(3) 000C2H/020C2H**

- Setting of RESOUTB output function
- Setting of the frequency of the high-speed on-chip oscillator
  - Select from 1 MHz, 4 MHz, 8 MHz, 12 MHz, 16 MHz, 24 MHz, 32 MHz, 48 MHz, and 64 MHz.

**Caution Set the same value as 000C2H to 020C2H when the boot swap operation is used because 000C2H is replaced by 020C2H.**

### 29.1.2 On-chip debug option byte (000C3H/ 020C3H)

- Control of on-chip debug operation
  - On-chip debug operation is disabled or enabled.
- Control of hot plug-in
  - Hot plug-in operation is disabled or enabled.
- Handling of data of flash memory in case of failure in on-chip debug security ID authentication
  - Data of flash memory is erased or not erased in case of failure in on-chip debug security ID authentication.

**Caution Set the same value as 000C3H to 020C3H when the boot swap operation is used because 000C3H is replaced by 020C3H.**

## 29.2 Format of User Option Byte

The format of user option byte is shown below.

**Figure 29-1. Format of User Option Byte (000C0H/020C0H)**

Address: 000C0H/020C0H[Note 1] After reset: — (user setting value [Note 2])

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WDTINT | WINDOW1 | WINDOW0 | WDTON | WDCS2 | WDCS1 | WDCS0 | WDSTBYON |

| WDTINT | Use of interval interrupt of watchdog timer |
|---|---|
| 0 | Interval interrupt is not used. |
| 1 | Interval interrupt is generated when 75% + 1/2 $f_{WDT}$ of the overflow time is reached. |

| WINDOW1 | WINDOW0 | Watchdog timer window open period[Note 3] |
|---|---|---|
| 0 | 0 | Setting prohibited |
| 0 | 1 | 50% |
| 1 | 0 | 75% |
| 1 | 1 | 100% |

| WDTON | Operation control of watchdog timer counter |
|---|---|
| 0 | Counter operation disabled (counting stopped after reset) |
| 1 | Counter operation enabled (counting started after reset) |

| WDCS2 | WDCS1 | WDCS0 | Watchdog timer overflow time ($f_{WDT}$ = 17.25 kHz (MAX.)) |
|---|---|---|---|
| 0 | 0 | 0 | $2^6/f_{WDT}$ (3.71 ms) |
| 0 | 0 | 1 | $2^7/f_{WDT}$ (7.42 ms) |
| 0 | 1 | 0 | $2^8/f_{WDT}$ (14.84 ms) |
| 0 | 1 | 1 | $2^9/f_{WDT}$ (29.68 ms) |
| 1 | 0 | 0 | $2^{11}/f_{WDT}$ (118.72 ms) |
| 1 | 0 | 1 | $2^{13}/f_{WDT}$ (474.90 ms) |
| 1 | 1 | 0 | $2^{14}/f_{WDT}$ (949.80 ms) |
| 1 | 1 | 1 | $2^{16}/f_{WDT}$ (3799.19 ms) |

| WDSTBYON | Operation control of watchdog timer counter (HALT/STOP/SNOOZE mode) |
|---|---|
| 0 | Counter operation stopped in HALT/STOP/SNOOZE mode[Note 3] |
| 1 | Counter operation enabled in HALT/STOP/SNOOZE mode |

**Notes 1.** Set the same value as 000C0H to 020C0H when the boot swap operation is used because 000C0H is replaced by 020C0H.

**2.** The setting at shipment of the user option byte is FFH.

**3.** The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

**Caution The watchdog timer continues its operation during EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window open period taking this delay into consideration.**

**Remarks 1.** $f_{WDT}$: Low-speed on-chip oscillator clock frequency

**2.** By specifying WDTON = 1, the invalid memory access detection function is always enabled regardless of the setting for the IAWEN bit. (For details, see **27.3.8 Invalid memory access detection function**.)

**Figure 29-2. Format of User Option Byte (000C1H/020C1H) (1/2)**

Address: 000C1H/020C1H[Note 1] After reset: — (user setting value [Note 2])

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPOC2 | VPOC1 | VPOC0 | CLKMB | LVIS1 | LVIS0 | LVIMDS1 | LVIMDS0 |

- LVD setting (interrupt & reset mode)

| Detection voltage | | | Option byte Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $V_{LVDH}$ | | $V_{LVDL}$ | VPOC2 | VPOC1 | VPOC0 | CLKMB | LVIS1 | LVIS0 | LVIMDS1 | LVIMDS0 |
| Rising edge | Falling edge | Falling edge | | | | | | | | |
| 4.42 V | 4.32 V | 2.75 V | 0 | 0 | 1 | ×[Note 3] | 0 | 0 | 1 | 0 |
| 4.62 V | 4.52 V | 2.75 V | 0 | 1 | 0 | ×[Note 3] | 0 | 0 | | |
| 3.32 V | 3.15 V | 2.75 V | 0 | 1 | 1 | ×[Note 3] | 0 | 1 | | |
| 4.74 V | 4.64 V | | | | | ×[Note 3] | 0 | 0 | | |
| Other than above | | | Setting prohibited | | | | | | | |

- LVD setting (reset mode)

| Detection voltage | | Option byte Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $V_{LVD}$ | | VPOC2 | VPOC1 | VPOC0 | CLKMB | LVIS1 | LVIS0 | LVIMDS1 | LVIMDS0 |
| Rising edge | Falling edge | | | | | | | | |
| 2.81 V | 2.75 V | 0 | 1 | 1 | ×[Note 3] | 1 | 1 | 1 | 1 |
| 3.02 V | 2.96 V | 0 | 0 | 0 | ×[Note 3] | 0 | 1 | | |
| 3.22 V | 3.15 V | 0 | 1 | 1 | ×[Note 3] | 0 | 1 | | |
| 4.42 V | 4.32 V | 0 | 0 | 1 | ×[Note 3] | 0 | 0 | | |
| 4.62 V | 4.52 V | 0 | 1 | 0 | ×[Note 3] | 0 | 0 | | |
| 4.74 V | 4.64 V | 0 | 1 | 1 | ×[Note 3] | 0 | 0 | | |
| Other than above | | Setting prohibited | | | | | | | |

**Notes 1.** Set the same value as 000C1H to 020C1H when the boot swap operation is used because 000C1H is replaced by 020C1H.

**2.** The setting at shipment of the user option byte is FFH.

**3.** Write the setting value of the clock monitor bit (CLKMB).

**Remarks 1.** ×: Don't care

**2.** For details of the LVD, see **26.1 Functions of Voltage Detector**.

**Figure 29-2. Format of User Option Byte (000C1H/020C1H) (2/2)**

Address: 000C1H/020C1H[Note 1] After reset: — (user setting value [Note 2])

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VPOC2 | VPOC1 | VPOC0 | CLKMB | LVIS1 | LVIS0 | LVIMDS1 | LVIMDS0 |

- LVD setting (interrupt mode)

| Detection voltage | | Option byte Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $V_{LVD}$ | | VPOC2 | VPOC1 | VPOC0 | CLKMB | LVIS1 | LVIS0 | LVIMDS1 | LVIMDS0 |
| Rising edge | Falling edge | | | | | | | | |
| 2.81 V | 2.75 V | 0 | 1 | 1 | × [Note 3] | 1 | 1 | 0 | 1 |
| 3.02 V | 2.96 V | 0 | 0 | 0 | × [Note 3] | 0 | 1 | | |
| 3.22 V | 3.15 V | 0 | 1 | 1 | × [Note 3] | 0 | 1 | | |
| 4.42 V | 4.32 V | 0 | 0 | 1 | × [Note 3] | 0 | 0 | | |
| 4.62 V | 4.52 V | 0 | 1 | 0 | × [Note 3] | 0 | 0 | | |
| 4.74 V | 4.64 V | 0 | 1 | 1 | × [Note 3] | 0 | 0 | | |
| Other than above | | Setting prohibited | | | | | | | |

- LVD setting (LVD off)

| Detection voltage | | Option byte Setting Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $V_{LVD}$ | | VPOC2 | VPOC1 | VPOC0 | CLKMB | LVIS1 | LVIS0 | Mode setting | |
| Rising edge | Falling edge | | | | | | | LVIMDS1 | LVIMDS0 |
| – | – | 1 | × | × | × [Note 3] | × | × | × | 1 |
| Other than above | | Setting prohibited | | | | | | | |

- Setting of clock monitor operation

| CLKMB | Control of clock monitor operation |
|---|---|
| 0 | Operation is enabled. |
| 1 | Operation is stopped. |

**Notes 1.** Set the same value as 000C1H to 020C1H when the boot swap operation is used because 000C1H is replaced by 020C1H.

**2.** The setting at shipment of the user option byte is FFH.

**3.** Write the setting value of the clock monitor bit (CLKMB).

**Remarks 1.** ×: don't care

**2.** For details of the LVD, see **26.1 Functions of Voltage Detector**.

**Figure 29-3. Format of Option Byte (000C2H/020C2H)**

Address: 000C2H/020C2H[Note 1] After reset: — (user setting value [Note 2])

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | RESOUTB | FRQSEL4 | FRQSEL3 | FRQSEL2 | FRQSEL1 | FRQSEL0 |

| RESOUTB | RESOUTB output function |
|---|---|
| 0 | Selects P130 as the RESOUT pin<br>• The low level is output during a reset.<br>• The high level is automatically output upon release from the reset state.<br>• The output latch value has no effect on the output. |
| 1 | Selects P130 as a general port pin (output only)<br>• The low level is output during a reset.<br>• The output latch value is output upon release from the reset state. |

| FRQSEL4 | FRQSEL3 | FRQSEL2 | FRQSEL1 | FRQSEL0 | Frequency of the high-speed on-chip oscillator clock |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 64 MHz |
| 1 | 0 | 0 | 0 | 0 | 48 MHz |
| 0 | 1 | 0 | 0 | 0 | 32 MHz |
| 0 | 0 | 0 | 0 | 0 | 24 MHz |
| 0 | 1 | 0 | 0 | 1 | 16 MHz |
| 0 | 0 | 0 | 0 | 1 | 12 MHz |
| 0 | 1 | 0 | 1 | 0 | 8 MHz |
| 0 | 1 | 0 | 1 | 1 | 4 MHz |
| 0 | 1 | 1 | 0 | 1 | 1 MHz |
| Other than above | | | | | Setting prohibited |

**Notes 1.** Set the same value as 000C2H to 020C2H when the boot swap operation is used because 000C2H is replaced by 020C2H.

**2.** The setting at shipment of the user option byte is FFH.

## 29.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

**Figure 29-4. Format of On-chip Debug Option Byte (000C3H/020C3H)**

Address: 000C3H/020C3H[Note 1]

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| OCDENSET | 0 | 0 | 0 | 0 | 1 | HPIEN [Note 2] | OCDERSD |

| OCDENSET | HPIEN [Note 2] | OCDERSD | Control of on-chip debug operation |
|---|---|---|---|
| 0 | 0 | 0 | Disables on-chip debug operation. |
| 1 | 0 | 0 | Enables on-chip debugging and disables hot plug-in operation.<br>Erases data of flash memory in case of failures in authenticating on-chip debug security ID. |
| 1 | 0 | 1 | Enables on-chip debugging and disables hot plug-in operation.<br>Does not erase data of flash memory in case of failures in authenticating on-chip debug security ID. |
| 1 | 1 | 1 | Enables on-chip debugging and hot plug-in operation.<br>Does not erase data of flash memory in case of failures in authenticating on-chip debug security ID. |
| Other than the above | | | Setting prohibited |

**Notes 1.** Set the same value as 000C3H to 020C3H when the boot swap operation is used because 000C3H is replaced by 020C3H.

**2.** When the HPIEN bit is set to 1, the low-speed on-chip oscillator operates and cannot be stopped by the user program. The low-speed on-chip oscillator can be stopped by register setting only in standby mode. Such operation is performed because the low-speed on-chip oscillator detects hot plug-in.

**Caution Bits 7, 1, and 0 (OCDENSET, HPIEN, and OCDERSD) can only be specified a value. Be sure to set 00001B to bits 6 to 2.**

**Remark** The value on bits 3 and 2 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting. However, be sure to set the default values (0, 1) to bits 3 and 2 at setting.

## 29.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the assembler linker option, in addition to describing to the source. When doing so, the contents set by using the linker option take precedence, even if descriptions exist in the source, as mentioned below.

A software description example of the option byte setting is shown below.

```
OPT   CSEG        OPT_BYTE
      DB          36H     ; Does not use interval interrupt of watchdog timer,
                          ; Enables watchdog timer operation,
                          ; Window open period of watchdog timer is 50%,
                          ; Overflow time of watchdog timer is 2^9/fWDT,
                          ; Stops watchdog timer operation during HALT/STOP/SNOOZE mode
      DB          22H     ; Select 2.75 V for VLVDL
                          ; Select rising edge 4.42 V, falling edge 4.32 V for VLVDH
                          ; Select the interrupt & reset mode as the LVD operation mode
                          ; Operation of clock monitor
      DB          EDH     ; Setting of the RESOUTB output function
                          ; Select 1 MHz as the frequency of the high-speed on-chip oscillator clock
      DB          85H     ; Enables on-chip debug operation, disables hot plug-in operation,
                            does not erase flash memory data when security ID authorization fails
```

When the boot swap function is used during self programming, 000C0H to 000C3H is switched to 020C0H to 020C3H. Describe to 020C0H to 020C3H, therefore, the same values as 000C0H to 000C3H as follows.

```
OPT2  CSEG   AT   020C0H
      DB          36H     ; Does not use interval interrupt of watchdog timer,
                          ; Enables watchdog timer operation,
                          ; Window open period of watchdog timer is 50%,
                          ; Overflow time of watchdog timer is 2^9/fWDT,
                          ; Stops watchdog timer operation during HALT/STOP/SNOOZE mode
      DB          22H     ; Select 2.75 V for VLVDL
                          ; Select rising edge 4.42 V, falling edge 4.32 V for VLVDH
                          ; Select the interrupt & reset mode as the LVD operation mode
                          ; Operation of clock monitor
      DB          EDH     ; Setting of the RESOUTB output function
                          ; Select 1 MHz as the frequency of the high-speed on-chip oscillator clock
      DB          85H     ; Enables on-chip debug operation, disables hot plug-in operation,
                            does not erase flash memory data when security ID authorization fails
```

**Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify the option byte to 020C0H to 020C3H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.**

# CHAPTER 30 FLASH MEMORY

The RL78/F13 and RL78/F14 incorporate the flash memory to which a program can be written, erased, and overwritten. The flash memory includes the “code flash memory”, in which programs can be executed, and the “data flash memory”, an area for storing data.

The methods for programming the flash memory are shown below.

The code flash memory can be rewritten to through serial programming using a flash memory programmer or external device (UART communication) or through self-programming.

- Serial programming using flash memory programmer
  Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer. For details, see **30.4 Serial Programming Method**.
- Serial programming using external device (UART communication)
  Data can be written to the flash memory on-board through UART communication with an external device (a microcontroller or ASIC). For details, see **30.2 Serial Programming Using External Device (that Incorporates UART)**.
- Self-programming
  The user application can execute self-programming of the code flash memory by using the flash self-programming library. For details, see **30.6 Self-Programming**.

The data flash memory can be rewritten to by using the data flash library during user program execution (background operation). For details about accessing or writing to the data flash memory, see **30.8 Data Flash**.

## 30.1 Serial Programming Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the RL78/F13 and RL78/F14.

- PG-FP5, FL-PR5
- E1 on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

**(1) On-board programming**

The contents of the flash memory can be rewritten after the RL78/F13 or RL78/F14 has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

**(2) Off-board programming**

Data can be written to the flash memory with a dedicated program adapter (FA series) before the RL78/F13 or RL78/F14 is mounted on the target system.

**Remark** FL-PR5 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

**Table 30-1. Wiring Between the RL78/F13 or RL78/F14 and Dedicated Flash Memory Programmer**

| Pin Configuration of Dedicated Flash Memory Programmer | | | | Pin Name | Pin No. | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | 20-pin | 30-pin | 32-pin | 48-pin |
| Signal Name | | I/O | Pin Function | | SSOP | SSOP | VQFN (5x5) | LQFP (7x7), VQFN (7x7) |
| PG-FP5, FL-PR5 | E1 on-chip debugging emulator | | | | | | | |
| – | TOOL0 | I/O | Transmit/receive signal | TOOL0/ P40 | 3 | 8 | 3 | 3 |
| SI/RxD | – | I/O | Transmit/receive signal | | | | | |
| – | $\overline{\text{RESET}}$ | Output | Reset signal | $\overline{\text{RESET}}$ | 4 | 9 | 4 | 4 |
| /RESET | – | Output | | | | | | |
| $V_{DD}$ | | I/O | $V_{DD}$ voltage generation/ power monitoring | $V_{DD}$ | 10 | 15 | 10 | 12 |
| GND | | – | Ground | $V_{SS}$ | 9 | 14 | 9 | 11 |
| | | | | $EV_{SS}$ | – | – | – | – |
| | | | | REGC **Note** | 8 | 13 | 8 | 10 |
| $EMV_{DD}$ | | – | Driving power for TOOL0 pin | $V_{DD}$ | 10 | 15 | 10 | 12 |
| | | | | $EV_{DD}$ | – | – | – | – |

| Pin Configuration of Dedicated Flash Memory Programmer | | | | Pin Name | Pin No. | | |
|---|---|---|---|---|---|---|---|
| | | | | | 64-pin | 80-pin | 100-pin |
| Signal Name | | I/O | Pin Function | | LQFP (10x10) | LQFP (12x12) | LQFP (14x14) |
| PG-FP5, FL-PR5 | E1 on-chip debugging emulator | | | | | | |
| – | TOOL0 | I/O | Transmit/receive signal | TOOL0/ P40 | 5 | 9 | 12 |
| SI/RxD | – | I/O | Transmit/receive signal | | | | |
| – | $\overline{\text{RESET}}$ | Output | Reset signal | $\overline{\text{RESET}}$ | 6 | 10 | 13 |
| /RESET | – | Output | | | | | |
| $V_{DD}$ | | I/O | $V_{DD}$ voltage generation/ power monitoring | $V_{DD}$ | 15 | 19 | 22 |
| GND | | – | Ground | $V_{SS}$ | 13 | 17 | 20 |
| | | | | $EV_{SS}$ | 14 | 18 | 21, 43 |
| | | | | REGC **Note** | 12 | 16 | 19 |
| $EMV_{DD}$ | | – | Driving power for TOOL0 pin | $V_{DD}$ | – | – | – |
| | | | | $EV_{DD}$ | 16 | 20 | 23, 53 |

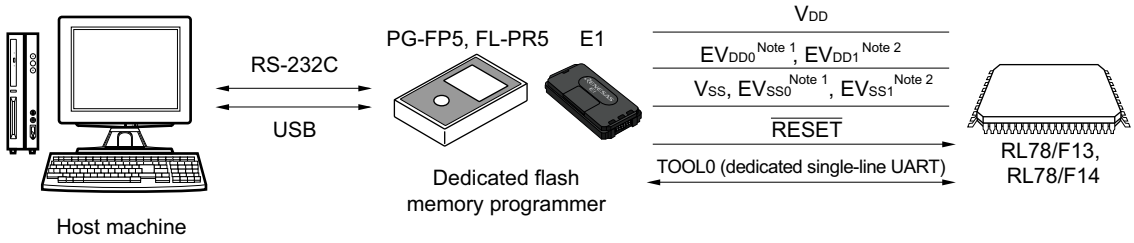
**Note** Connect REGC pin to ground via a capacitor (0.47 to 1 $\mu$F).

**Remark** Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

### 30.1.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78/F13 or RL78/F14 is illustrated below.

**Figure 30-1. Environment for Writing Program to Flash Memory**




**Notes 1.** 64, 80, 100-pin products only.
**2.** 100-pin products only.

A host machine that controls the dedicated flash memory programmer is necessary.

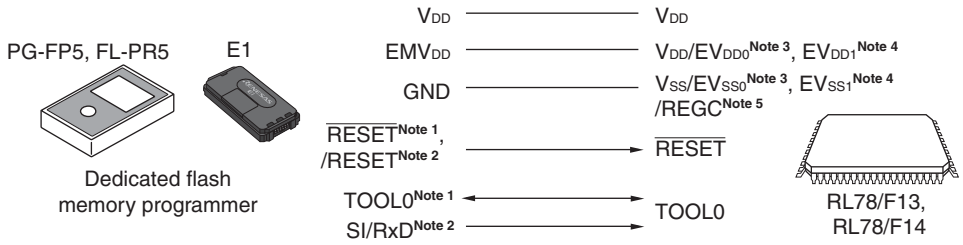
To interface between the dedicated flash memory programmer and the RL78/F13 or RL78/F14, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

### 30.1.2 Communication Mode

Communication between the dedicated flash memory programmer and the RL78/F13 or RL78/F14 is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78/F13 or RL78/F14.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

**Figure 30-2. Communication with Dedicated Flash Memory Programmer**




**Notes 1.** When using E1 on-chip debugging emulator.
**2.** When using PG-FP5 or FL-PR5.
**3.** 64, 80, 100-pin products only.
**4.** 100-pin products only.
**5.** Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The dedicated flash memory programmer generates the following signals for the RL78/F13 and RL78/F14. See the manual of PG-FP5, FL-PR5, or E1 on-chip debugging emulator for details.

**Table 30-2. Pin Connection**

| Dedicated Flash Memory Programmer | | | | RL78/F13 and RL78/F14 |
|---|---|---|---|---|
| Signal Name | | I/O | Pin Function | Pin Name |
| PG-FP5, FL-PR5 | E1 on-chip debugging emulator | | | |
| $V_{DD}$ | | I/O | $V_{DD}$ voltage generation/power monitoring | $V_{DD}$ |
| GND | | – | Ground | $V_{SS}$, $EV_{SS0}$ Note 1, $EV_{SS0}$ Note 2, REGC Note 3 |
| $EMV_{DD}$ | | – | Driving power for TOOL0 pin | $V_{DD}$, $EV_{DD0}$ Note 1, $EV_{DD1}$ Note 2 |
| /RESET | – | Output | Reset signal | $\overline{\text{RESET}}$ |
| – | $\overline{\text{RESET}}$ | Output | | |
| – | TOOL0 | I/O | Transmit/receive signal | TOOL0 |
| SI/RxD | – | I/O | Transmit/receive signal | |

**Notes 1.** 64, 80, 100-pin products only.

**2.** 100-pin products only.

**3.** Connect REGC pin to ground via a capacitor (0.47 to 1 $\mu$F).

**Caution The connection destination pins differ depending on the product. For details, see Table 30-1.**

## 30.2 Serial Programming Using External Device (that Incorporates UART)

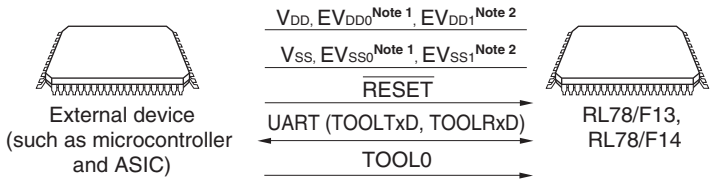
On-board data writing to the internal flash memory is possible by using the RL78/F13 or RL78/F14 and an external device (a microcontroller or ASIC) connected to a UART.

On the development of flash memory programmer by user, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

### 30.2.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78/F13 or RL78/F14 is illustrated below.

**Figure 30-3. Environment for Writing Program to Flash Memory**




**Notes 1.** 64, 80, 100-pin products only.

**2.** 100-pin products only.

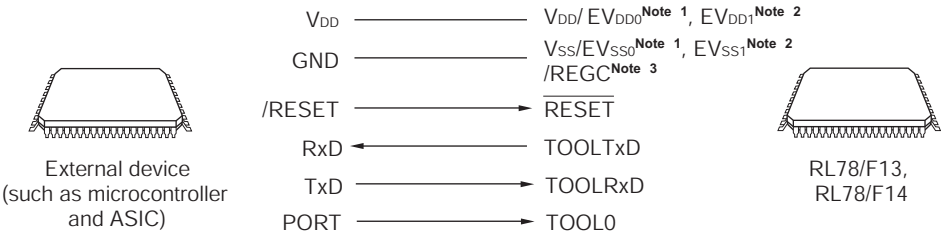
Processing to write data to or delete data from the RL78/F13 or RL78/F14 by using an external device is performed on-board. Off-board writing is not possible.

### 30.2.2 Communication Mode

Communication between the external device and the RL78/F13 or RL78/F14 is established by serial communication using the TOOLTxD and TOOLRxD pins via the dedicated UART of the RL78/F13 or RL78/F14.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

**Figure 30-4. Communication with External Device**




**Notes 1.** 64, 80, 100-pin products only.
**2.** 100-pin products only.
**3.** Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The external device generates the following signals for the RL78/F13 and RL78/F14.

**Table 30-3. Pin Connection**

| External Device | | | RL78/F13 and RL78/F14 |
|---|---|---|---|
| Signal Name | I/O | Pin Function | Pin Name |
| $V_{DD}$ | I/O | $V_{DD}$ voltage generation/power monitoring | $V_{DD}$, $EV_{DD0}$ Note 1, $EV_{DD1}$ Note 2 |
| GND | – | Ground | $V_{SS}$, $EV_{SS0}$ Note 1, $EV_{SS1}$ Note 2, REGC Note 3 |
| RESETOUT | Output | Reset signal output | $\overline{\text{RESET}}$ |
| RxD | Input | Receive signal | TOOLTxD |
| TxD | Output | Transmit signal | TOOLRxD |
| PORT | Output | Mode signal | TOOL0 |

**Notes 1.** 64, 80, 100-pin products only.
**2.** 100-pin products only.
**3.** Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

## 30.3 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

**Remark** For flash programming mode, see **30.4.2 Flash memory programming mode**.

### 30.3.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1 kΩ pull-up resistor.

When this pin is used as the port pin, use that by the following method.

When used as an input pin: Input of low-level is prohibited for 1 ms period after the pin reset is released. However, when this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

When used as an output pin: When this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

**Remarks 1.** $t_{HD}$: How long to keep the TOOL0 pin at the low level from when the external and internal resets end for setting of the flash memory programming mode.

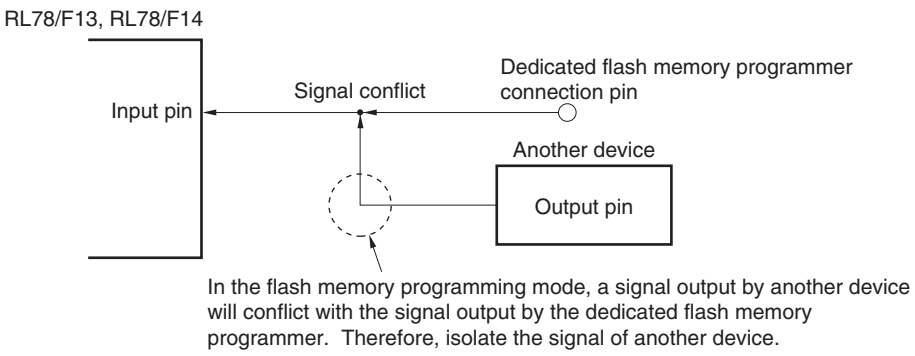
**2.** The SAU and IICA pins are not used for communication between the RL78/F13 or RL78/F14 and the dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

### 30.3.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.

The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer and external device.

**Figure 30-5. Signal Conflict ($\overline{\text{RESET}}$ Pin)**



### 30.3.3 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to either to $V_{DD}$, $EV_{DD0}$ Note1, or $EV_{DD1}$ Note2, or $V_{SS}$, $EV_{SS0}$ Note 1, or $EV_{SS1}$ Note 2, via a resistor.

**Notes 1.** 64, 80, 100-pin products only.

**2.** 100-pin products only.

### 30.3.4 REGC pin

Connect the REGC pin to GND via a capacitor (0.47 to 1 $\mu$F) in the same manner as during normal operation. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

### 30.3.5 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

**Remark** In the flash memory programming mode, the high-speed on-chip oscillator clock ($f_{IH}$) is used.

### 30.3.6 Power supply

To use the supply voltage output of the flash memory programmer, connect the $V_{DD}$ pin to $V_{DD}$ of the flash memory programmer, and the $V_{SS}$ pin to GND of the flash memory programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

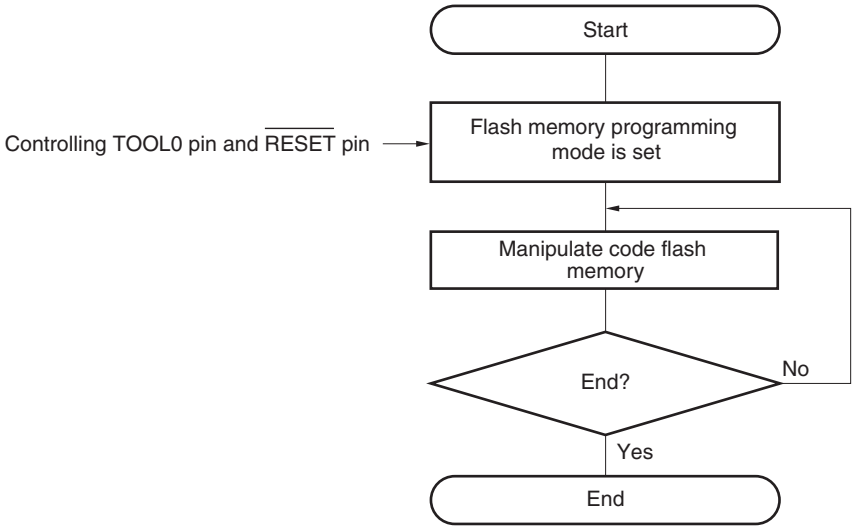
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the $V_{DD}$ and $V_{SS}$ pins to $V_{DD}$ and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

## 30.4 Serial Programming Method

### 30.4.1 Serial programming procedure

The following figure illustrates the procedure to manipulate the flash memory.

**Figure 30-6. Code Flash Memory Manipulation Procedure**



### 30.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory through serial programming, specify the flash memory programming mode. To switch to the mode, set as follows.

<Serial programming using the dedicated flash memory programmer>

Connect the RL78/F13 and RL78/F14 to the dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

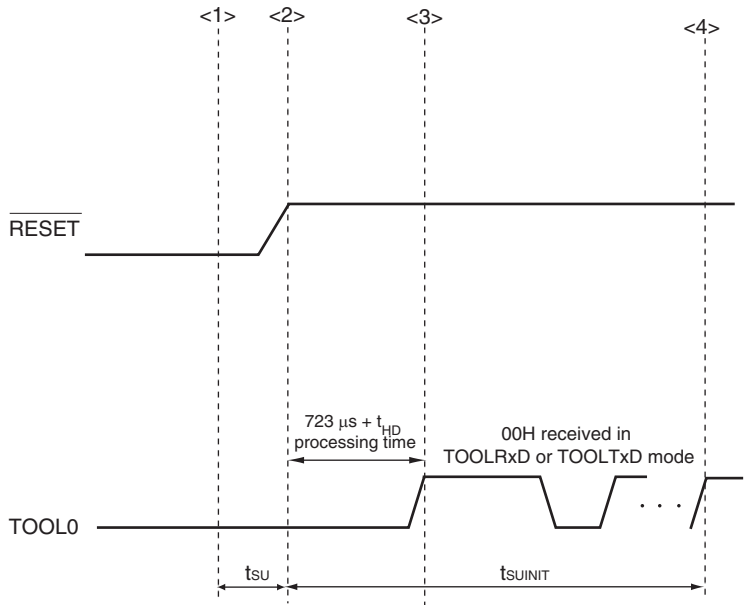
<Serial programming using an external device (UART communication)>

Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 30-4**). Then, perform steps <1> to <4> in Figure 30-7 to enter the flash memory programming mode. For details, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

**Table 30-4. Relationship Between TOOL0 Pin and Operation Mode After Reset Release**

| TOOL0 | Operation Mode |
|---|---|
| $EV_{DD}$ | Normal operation mode |
| 0 V | Flash memory programming mode |

**Figure 30-7. Setting of Flash Memory Programming Mode**




<1> The low level is input to the TOOL0 pin.

<2> The external reset ends (POR and LVD reset must end before the external reset ends.).

<3> The TOOL0 pin is set to the high level.

<4> Baud rate setting by UART reception is completed.

**Remark** $t_{SUINIT}$: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the resets end.

$t_{SU}$: How long from when the TOOL0 pin is placed at the low level until an external reset ends

$t_{HD}$: How long to keep the TOOL0 pin at the low level from when the external and internal resets end (the flash firmware processing time is excluded)

The voltage range in which to write, erase, or verify data in flash memory programming mode is shown in Table 30-5.

**Table 30-5. Voltages at Which Data Can Be Written, Erased, or Verified**

| Voltages at which data can be written, erased, or verified | Operating frequency |
|---|---|
| 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | 1 MHz to 32 MHz |

**Remark** For details about communication commands, see **30.4.4 Communication commands**.

### 30.4.3 Selecting communication mode

Communications modes of the RL78/F13 and RL78/F14 are as follows.

**Table 30-6. Communication Modes**

| Communication Mode | Standard Setting[Note 1] | | | | Pins Used |
|---|---|---|---|---|---|
| | Port | Speed [Note 2] | Frequency | Multiply Rate | |
| 1-line mode (when flash memory programmer or an external device is used) | UART | 115200 bps, 250000 bps, 500000 bps, 1 Mbps | – | – | TOOL0 |
| UART0 (when external device is used) | UART | 115200 bps, 250000 bps, 500000 bps, 1 Mbps | – | – | TOOLTxD, TOOLRxD |

**Notes 1.** Selection items for Standard settings on GUI of the flash memory programmer.

**2.** Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

### 30.4.4 Communication commands

The RL78/F13 and RL78/F14 executes serial programming through the commands listed in Table 30-7.

The signals sent from the dedicated flash memory programmer or external device to the RL78/F13 or RL78/F14 are called commands, and programming functions corresponding to the commands are executed. For details, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

**Table 30-7. Flash Memory Control Commands**

| Classification | Command Name | Function |
|---|---|---|
| Verify | Verify | Compares the contents of a specified area of the flash memory with data transmitted from the programmer. |
| Erase | Block Erase | Erases a specified area in the flash memory. |
| Blank check | Block Blank Check | Checks if a specified block in the flash memory has been correctly erased. |
| Write | Programming | Writes data to a specified area in the flash memory.Note |
| Getting information | Silicon Signature | Gets information from the RL78/F13 or RL78/F14 (such as the part number, flash memory configuration, and programming firmware version). |
| | Checksum | Gets the checksum data for a specified area. |
| Security | Security Set | Sets security information. |
| | Security Get | Gets security information. |
| | Security Release | Release setting of prohibition of writing. |
| Others | Reset | Used to detect synchronization status of communication. |
| | Baud Rate Set | Sets baud rate when UART communication mode is selected. |

**Note** Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

Product information (such as product name and firmware version) can be obtained by executing the “Silicon Signature” command.

Table 30-8 lists and describes signature data. Table 30-9 shows examples of signature data.

**Table 30-8. Signature Data List**

| Field name | Description | Number of transmit data |
|---|---|---|
| Device code | The serial number assigned to the device | 3 bytes |
| Device name | Device name (ASCII code) | 10 bytes |
| Code flash memory area last address | Last address of code flash memory area<br>(Sent from lower address.<br>Example. 00000H to 0FFFFH (64 KB) → FFH, 1FH, 00H) | 3 bytes |
| Data flash memory area last address | Last address of data flash memory area<br>(Sent from lower address.<br>Example. F1000H to F1FFFH (4 KB) → FFH, 1FH, 0FH) | 3 bytes |
| Firmware version | Version information of firmware for programming<br>(Sent from upper address.<br>Example. From Ver. 1.23 → 01H, 02H, 03H) | 3 bytes |

**Table 30-9. Example of Signature Data**

| Field name | Description | Number of transmit data | Data (hexadecimal) |
|---|---|---|---|
| Device code | RL78 protocol A | 3 bytes | 10 00 06 |
| Device name | R5F10BLE | 10 bytes | 52 = "R"<br>35 = "5"<br>46 = "F"<br>31 = "1"<br>30 = "0"<br>42 = "B"<br>4C = "L"<br>45 = "E"<br>20 = " "<br>20 = “ ” |
| Code flash memory area last address | Code flash memory area<br>00000H to 0FFFFH (64 KB) | 3 bytes | FF FF 00 |
| Data flash memory area last address | Data flash memory area<br>F1000H to F1FFFH (4 KB) | 3 bytes | FF 1F 0F |
| Firmware version | Ver.1.23 | 3 bytes | 01 02 03 |

## 30.5 Processing Time for Each Command when PG-FP5 Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP5 is used as a dedicated flash memory programmer.

**Table 30-10. Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)**

| PG-FP5 Command | Code Flash | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 16 KB | 32 KB | 48 KB | 64 KB | 96 KB | 128 KB | 192 KB | 256 KB |
| Erase | 1 s | 1 s | 1 s | 1.5 s | 1.5 s | 2 s | 2 s | 2.5 s |
| Write | 1.5 s | 1.5 s | 2 s | 2.5 s | 3 s | 3.5 s | 5 s | 6 s |
| Verify | 1.5 s | 1.5 s | 2 s | 2 s | 3 s | 3.5 s | 4.5 s | 5.5 s |
| Write after erase | 1.5 s | 2 s | 2.5 s | 3 s | 4 s | 4.5 s | 6.5 s | 8 s |

**Remark** The command processing times (reference value) shown in the table are typical values under the following conditions.
Port: TOOL0 (single-line UART)
Speed: 1,000,000 bps

## 30.6 Self-Programming

The RL78/F13 and RL78/F14 support self-programming functions that can be used to rewrite the code flash memory via a user program. Because this function allows a user application to rewrite the code flash memory by using the flash self-programming library, it can be used to upgrade the program in the field.

**Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem/low-speed on-chip oscillator select clock.**

**2. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the flash self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the flash self-programming library.**

**3. The high-speed on-chip oscillator should be kept operating during self-programming. If this oscillator is stopped, start the high-speed on-chip oscillator clock by setting HIOSTOP to 0. Then, after 30 μs has elapsed, execute the flash self-programming library.**

**Remarks 1.** For details of the self-programming function, refer to **RL78 Microcontroller Flash Self Programming Library Type01 User's Manual (R01AN0350).**

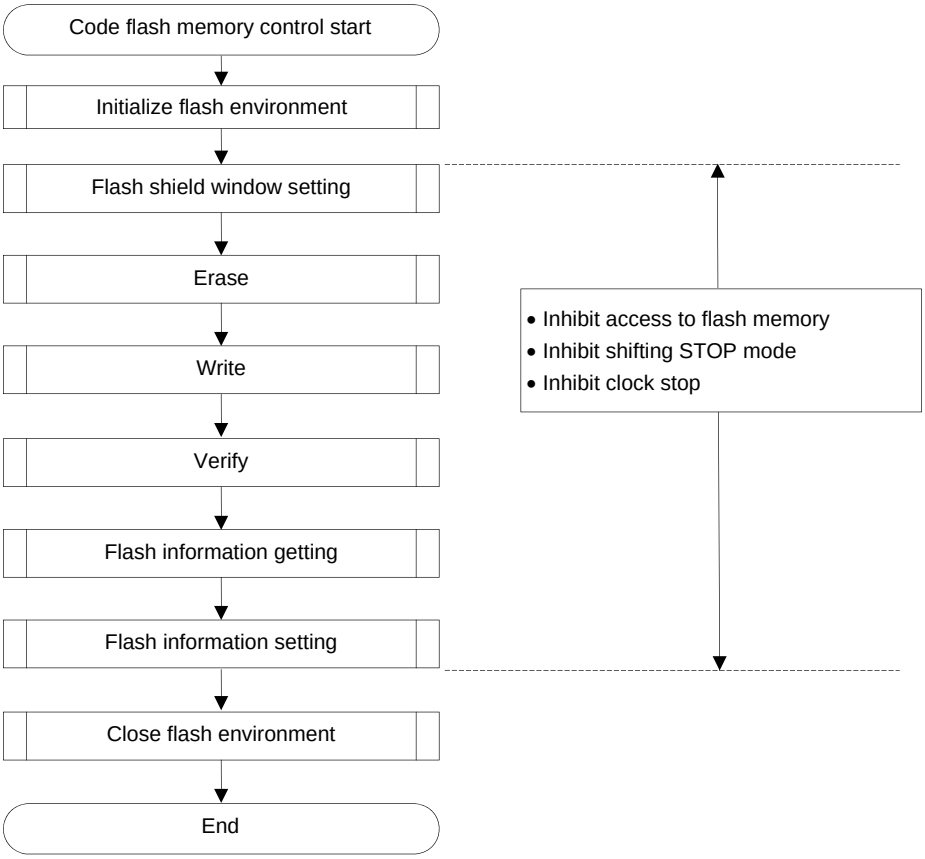
**2.** For details of the time required to execute self programming, see the notes on use that accompany the flash self-programming library tool.

Note that the self-programming function has two modes for flash memory programming mode: wide voltage mode and full-speed mode. Because RL78/F13 and RL78/F14 do not have wide voltage mode, select full-speed mode.

### 30.6.1 Self-programming procedure

The following figure illustrates a flow of rewriting the code flash memory by using a flash self-programming library.

**Figure 30-8. Flow of Self Programming (Rewriting Flash Memory)**



### 30.6.2 Boot swap function

If rewriting the boot area failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.

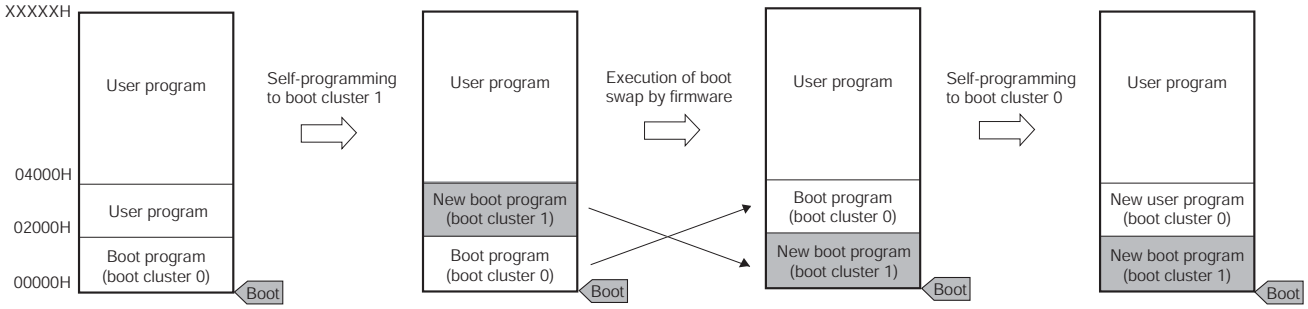
The boot swap function is used to avoid this problem.

Before erasing boot cluster 0[Note], which is a boot program area, by self-programming, write a new boot program to boot cluster 1 in advance. When the program has been correctly written to boot cluster 1, swap this boot cluster 1 and boot cluster 0 by using the set information function of the firmware of the RL78/F13 or RL78/F14, so that boot cluster 1 is used as a boot area. After that, erase or write the original area, boot cluster 0.

As a result, even if a power failure occurs while the area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 to be swapped when the program is reset and started next.

**Note** A boot cluster is an 8 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

**Figure 30-9. Boot Swap Function**

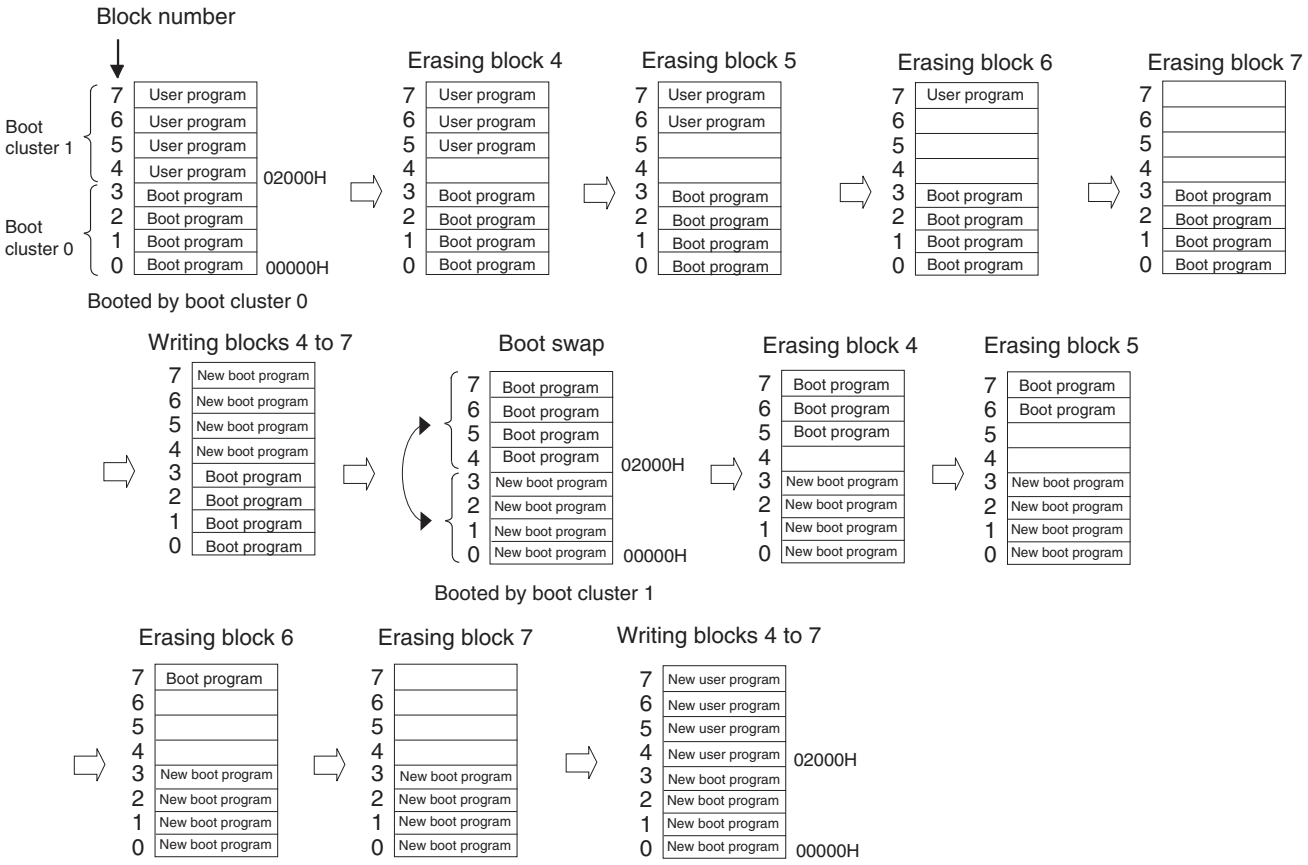



In an example of above figure, it is as follows.

Boot cluster 0: Boot program area before boot swap

Boot cluster 1: Boot program area after boot swap

**Figure 30-10. Example of Executing Boot Swapping**



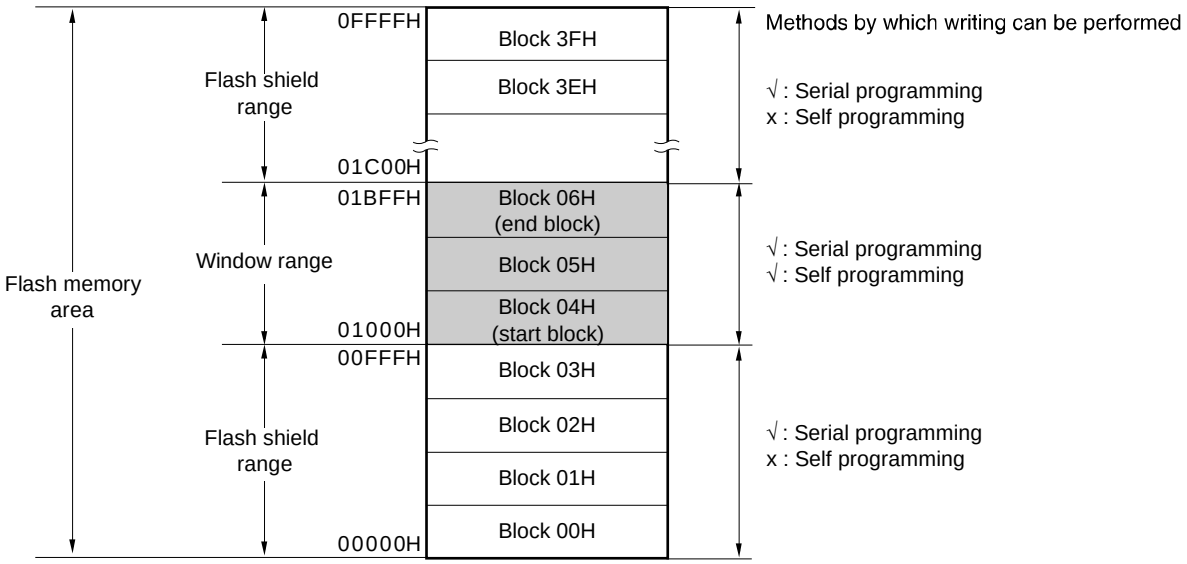
### 30.6.3 Flash shield window function

The flash shield window function is provided as one of the security functions for self programming. It disables writing to and erasing areas outside the range specified as a window only during self programming.

The window range can be set by specifying the start and end blocks. The window range can be set or changed during both serial programming and self-programming.

Writing to and erasing areas outside the window range are disabled during self programming. During serial programming, however, areas outside the range specified as a window can be written and erased.

**Figure 30-11. Flash Shield Window Setting Example**
**(Target Devices: R5F10BLE, Start Block: 04H, End Block: 06H)**




**Cautions 1. If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.**

**2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).**

**Table 30-11. Relationship between Flash Shield Window Function Setting/Change Methods and Commands**

| Programming conditions | Window Range Setting/Change Methods | Execution Commands | |
|---|---|---|---|
| | | Block erase | Write |
| Self-programming | Specify the starting and ending blocks by the flash self programming library. | Block erasing is enabled only within the window range. | Writing is enabled only within the range of window range. |
| Serial programming | Specify the starting and ending blocks on GUI of dedicated flash memory programmer, etc. | Block erasing is enabled also outside the window range. | Writing is enabled also outside the window range. |

**Remark** See **30.7 Security Settings** to prohibit writing or erasing during serial programming.

## 30.7 Security Settings

The RL78/F13 and RL78/F14 support security functions that prohibit rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the Security Set command.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during serial programming. However, blocks can be erased by means of self programming.

- Disabling write
  Execution of the write command for entire blocks in the code flash memory is prohibited during serial programming. However, blocks can be written by means of self programming.
  After the setting of prohibition of writing is specified, releasing the setting by the Security Release command is enabled by a reset.

- Disabling rewriting boot cluster 0
  Execution of the block erase command and write command on boot cluster 0 (00000H to 1FFFH) in the code flash memory is prohibited by this setting.

The block erase, write commands, and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by serial programming and self programming. Each security setting can be used in combination.

Table 30-12 shows the relationship between the erase and write commands when the RL78/F13 or RL78/F14 security function is enabled.

**Caution The security functions of the dedicated flash programmer does not support self-programming.**

**Remark** To prohibit writing and erasing during self-programming, use the flash shield window function (see **30.6.3 Flash shield window function** for detail).

**Table 30-12. Relationship Between Enabling Security Function and Command**

**(1) During serial programming**

| Valid Security | Executed Command | |
|---|---|---|
| | Block Erase | Write |
| Prohibition of block erase | Blocks cannot be erased. | Can be performed. Note |
| Prohibition of writing | Blocks can be erased. | Cannot be performed. |
| Prohibition of rewriting boot cluster 0 | Boot cluster 0 cannot be erased. | Boot cluster 0 cannot be written. |

**Note** Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

**(2) During self programming**

| Valid Security | Executed Command | |
|---|---|---|
| | Block Erase | Write |
| Prohibition of block erase | Blocks can be erased. | Can be performed. |
| Prohibition of writing | | |
| Prohibition of rewriting boot cluster 0 | Boot cluster 0 cannot be erased. | Boot cluster 0 cannot be written. |

**Remark** To prohibit writing and erasing during self-programming, use the flash shield window function (see **30.6.3 Flash shield window function** for detail).

**Table 30-13. Setting Security in Each Programming Mode**

**(1) Serial programming**

| Security | Security Setting | How to Disable Security Setting |
|---|---|---|
| Prohibition of block erase | Set via GUI of dedicated flash memory programmer, etc. | Cannot be disabled after set. |
| Prohibition of writing | | Set via GUI of dedicated flash memory programmer, etc. |
| Prohibition of rewriting boot cluster 0 | | Cannot be disabled after set. |

**Caution Releasing the setting of prohibition of writing is enabled only when the security is not set as the block erase prohibition and the boot cluster 0 rewrite prohibition with code flash memory area and data flash memory area being blanks.**

**(2) Self programming**

| Security | Security Setting | How to Disable Security Setting |
|---|---|---|
| Prohibition of block erase | Set by using flash self programming library. | Cannot be disabled after set. |
| Prohibition of writing | | Cannot be disabled during self-programming. (Set via GUI of dedicated flash memory programmer, etc. during serial programming.) |
| Prohibition of rewriting boot cluster 0 | | Cannot be disabled after set. |

## 30.8 Data Flash

### 30.8.1 Data flash overview

An overview of the data flash memory is provided below.

- The data flash memory can be rewritten by a user program using the data flash library. For details, refer to the **RL78 Family Microcontroller Data Flash Library User's Manual**.
- The data flash memory can also be rewritten by serial programming using a dedicated flash memory programmer or an external device.
- Blocks in the data flash memory can be erased in 1-KB units.
- The data flash memory can be accessed only in 8-bit units.
- The data flash memory can be directly read by CPU instructions.
- Instructions can be executed from the code flash memory while rewriting the data flash memory (back ground operation (BGO) is supported).
- Because the data flash memory is an area exclusively used for data, executing instructions from the data flash memory is prohibited.
- Accessing the data flash memory is prohibited while rewriting the code flash memory (during self-programming)
- Manipulating the DFLCTL register is prohibited while rewriting the data flash memory.
- Transition to the STOP status is prohibited while rewriting the data flash memory.
- The data flash memory can be programmed by using a Renesas library while other programs are running.

**Cautions 1. The data flash memory is stopped after a reset is released. To use the data flash memory, the data flash control register (DFLCTL) must be set up.**

**2. The high-speed on-chip oscillator must be running while rewriting the data flash memory. If this oscillator is stopped, start the high-speed on-chip oscillator clock by setting HIOSTOP to 0. Then, after 30 μs has elapsed, execute the data flash library.**

**Remark** For details about rewriting the code flash memory by using a user program, see **30.6 Self-Programming**.

### 30.8.2 Register controlling data flash memory

#### 30.8.2.1 Data flash control register (DFLCTL)

This register is used to enable or disable accessing to the data flash.

The DFLCTL register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

**Figure 30-12. Format of Data Flash Control Register (DFLCTL)**

Address: F0090H After reset: 00H R/W

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | <0> |
|---|---|---|---|---|---|---|---|---|
| DFLCTL | 0 | 0 | 0 | 0 | 0 | 0 | 0 | DFLEN |

| DFLEN | Data flash access control |
|---|---|
| 0 | Disables data flash access |
| 1 | Enables data flash access |

**Caution Manipulating the DFLCTL register is not possible while rewriting the data flash memory.**

### 30.8.3 Procedure for accessing data flash memory

The data flash memory is stopped after a reset is released. To access the data flash memory, initial settings must be specified as described below.

After the initial settings are specified, the data flash memory can be read by CPU instructions and can be read or rewritten by using a data flash library.

<1> Set 1 to bit 0 (DFLEN) of the data flash control register (DFLCTL).
<2> Use a software timer to wait for the setup to finish.
Setup time: 5 μs
<3> After the wait, the data flash memory can be accessed.

**Cautions 1. Accessing the data flash memory is prohibited during the setup time.**
**2. Transition to the STOP mode is not possible during the setup time. To enter the STOP mode during the setup time, clear DFLEN to 0 and then execute the STOP instruction.**
**3. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The data flash library should be executed after 30 μs have elapsed.**

# CHAPTER 31 ON-CHIP DEBUG FUNCTION

## 31.1 Overview of On-chip Debug Function

The RL78/F13 and RL78/F14 have stronger on-chip debug functions than the conventional RL78 family microcontrollers.

The following three functions are stronger. For points requiring cautions in using these functions, refer to E1/E20 Emulator User's Manual (R20UT0398).

- Hot plug-in
- Real-time RAM monitor (RRM) and dynamic memory modification (DMM) by the DTC
- On-chip trace

### 31.1.1 Hot Plug-in

This function is for connecting the MCU with an emulator without stopping or resetting a user program which is in execution. This function uses RAM in some products.

### 31.1.2 Real-time RAM Monitor (RRM) and Dynamic Memory Modification (DMM) by DTC

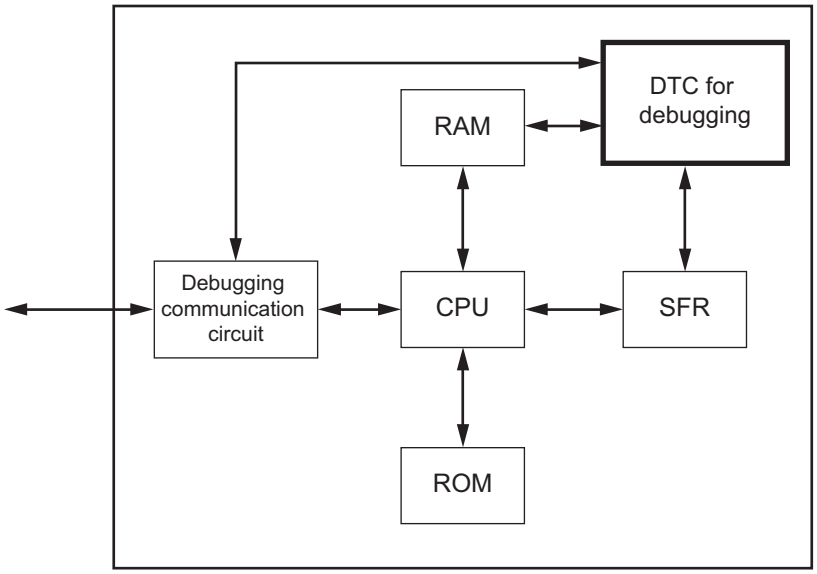
These functions are for accessing the MCU memory during the execution of the user program after the connection between the MCU and emulator.

The CPU handles all access to memory in the conventional RL78 family microcontrollers. The RL78/F13 and RL78/F14 allow access to memory without using the CPU because they have a DTC for debugging.

These functions use RAM in some products.

Figure 31-1 shows the configuration of RRM and DMM by the DTC.

**Figure 31-1. Configuration of RRM and DMM by DTC**



### 31.1.3 On-chip Trace

This function is for retaining the program counter values of branch sources when branches occur.

This function can retain the values of branches due to the execution of branch instructions, interrupts, and resets. This function uses RAM to retain traces in some products.

The RAM area used and the number of branches retained by the on-chip trace vary with the product.

Table 31-1 shows the RAM area used and the number of branches retained by on-chip trace.

**Table 31-1. RAM Area Used and Number of Branches Retained by On-chip Trace**

| Series Name | Products | RAM | RAM Area Used | Number of Branches |
|---|---|---|---|---|
| RL78/F13 (LIN incorporated) | R5F10A6A, R5F10AAA, R5F10ABA, R5F10AGA | 1 KB | – | 64 |
| | R5F10A6C, R5F10AAC, R5F10ABC, R5F10AGC, R5F10ALC | 2 KB | | |
| | R5F10A6D, R5F10AAD, R5F10ABD, R5F10AGD, R5F10ALD | 3 KB | • 0FF400H-0FF42FH (Hot plug-in/RRM and DDM by DTC)<br>• 0FF300H-0FF37FH (On-chip trace) | |
| | R5F10A6E, R5F10AAE, R5F10ABE, R5F10AGE, R5F10ALE | 4 KB | | |
| | R5F10AME | 4 KB | – | 128 |
| | R5F10AGF, R5F10ALF, R5F10AMF | 6 KB | | |
| | R5F10AGG, R5F10ALG, R5F10AMG | 8 KB | • 0FE500H-0FE52FH (Hot plug-in/RRM and DDM by DTC)<br>• 0FE300H-0FE4FFH (On-chip trace) | |
| RL78/F13 (CAN and LIN incorporated) | R5F10BAC, R5F10BBC, R5F10BGC, R5F10BLC | 2 KB | – | |
| | R5F10BAD, R5F10BBD, R5F10BGD, R5F10BLD | 3 KB | | |
| | R5F10BAE, R5F10BBE, R5F10BGE, R5F10BLE, R5F10BME | 4 KB | | |
| | R5F10BAF, R5F10BBF, R5F10BGF, R5F10BLF, R5F10BMF | 6 KB | | |
| | R5F10BAG, R5F10BBG, R5F10BGG, R5F10BLG, R5F10BMG | 8 KB | • 0FE500H-0FE52FH (Hot plug-in/RRM and DDM by DTC)<br>• 0FE300H-0FE4FFH (On-chip trace) | |
| RL78/F14 | R5F10PAD, R5F10PBD, R5F10PGD | 4 KB | – | |
| | R5F10PAE, R5F10PBE, R5F10PGE, R5F10PLE, R5F10PME | 6 KB | | |
| | R5F10PPE | 6 KB | | |
| | R5F10PGF, R5F10PLF, R5F10PMF | 8 KB | • 0FE500H-0FE52FH (Hot plug-in/RRM and DDM by DTC)<br>• 0FE300H-0FE4FFH (On-chip trace) | |
| | R5F10PPF | 8 KB | – | |
| | R5F10PGG, R5F10PLG, R5F10PMG, R5F10PPG | 10 KB | | |
| | R5F10PGH, R5F10PLH, R5F10PMH, R5F10PPH | 16 KB | | |
| | R5F10PGJ, R5F10PLJ, R5F10PMJ, R5F10PPJ | 20 KB | • 0FB500H-0FB52FH (Hot plug-in/RRM and DDM by DTC)<br>• 0FB300H-0FB4FFH (On-chip trace) | |

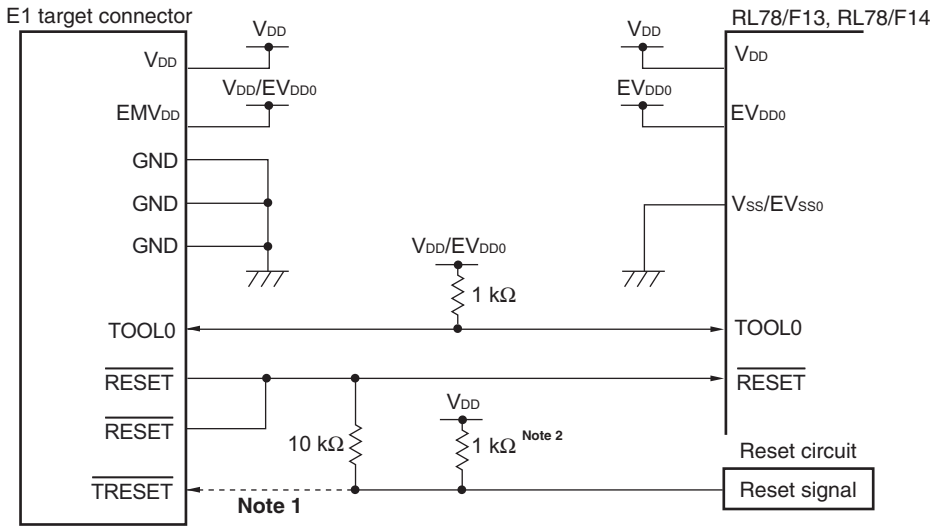
## 31.2 Connecting E1 On-chip Debugging Emulator to RL78/F13 or RL78/F14

The RL78/F13 and RL78/F14 use the $V_{DD}$, $EV_{DD0}$, $\overline{RESET}$, TOOL0, and $V_{SS}$ pins to communicate with the host machine via an E1 on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

The RL78/F13 and RL78/F14 are provided with the hot plug-in detection function.

**Caution The RL78/F13 and RL78/F14 have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.**

**Figure 31-2. Connection Example of E1 On-chip Debugging Emulator and RL78/F13 or RL78/F14**




**Notes 1.** Connecting the dotted line is not necessary during flash programming.

**2.** If the reset circuit on the target system does not have a buffer and generates a reset signal only with resistors and capacitors, this pull-up resistor is not necessary.

**Caution This circuit diagram is assumed that the reset signal outputs from an N-ch O.D. buffer (output resistor: 100 Ω or less)**

## 31.3 On-Chip Debug Security ID

The RL78/F13 and RL78/F14 have an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 29 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

When the boot swap function is used, also set a value that is the same as that of 020C3H and 020C4H to 020CDH in advance, because 000C3H, 000C4H to 000CDH and 020C3H, and 020C4H to 020CDH are switched.

**Table 31-2. On-Chip Debug Security ID**

| Address | On-Chip Debug Security ID |
|---|---|
| 000C4H to 000CDH | Any ID code of 10 bytes (except for All FFH) |
| 020C4H to 020CDH | |

## 31.4 Securing of User Resources

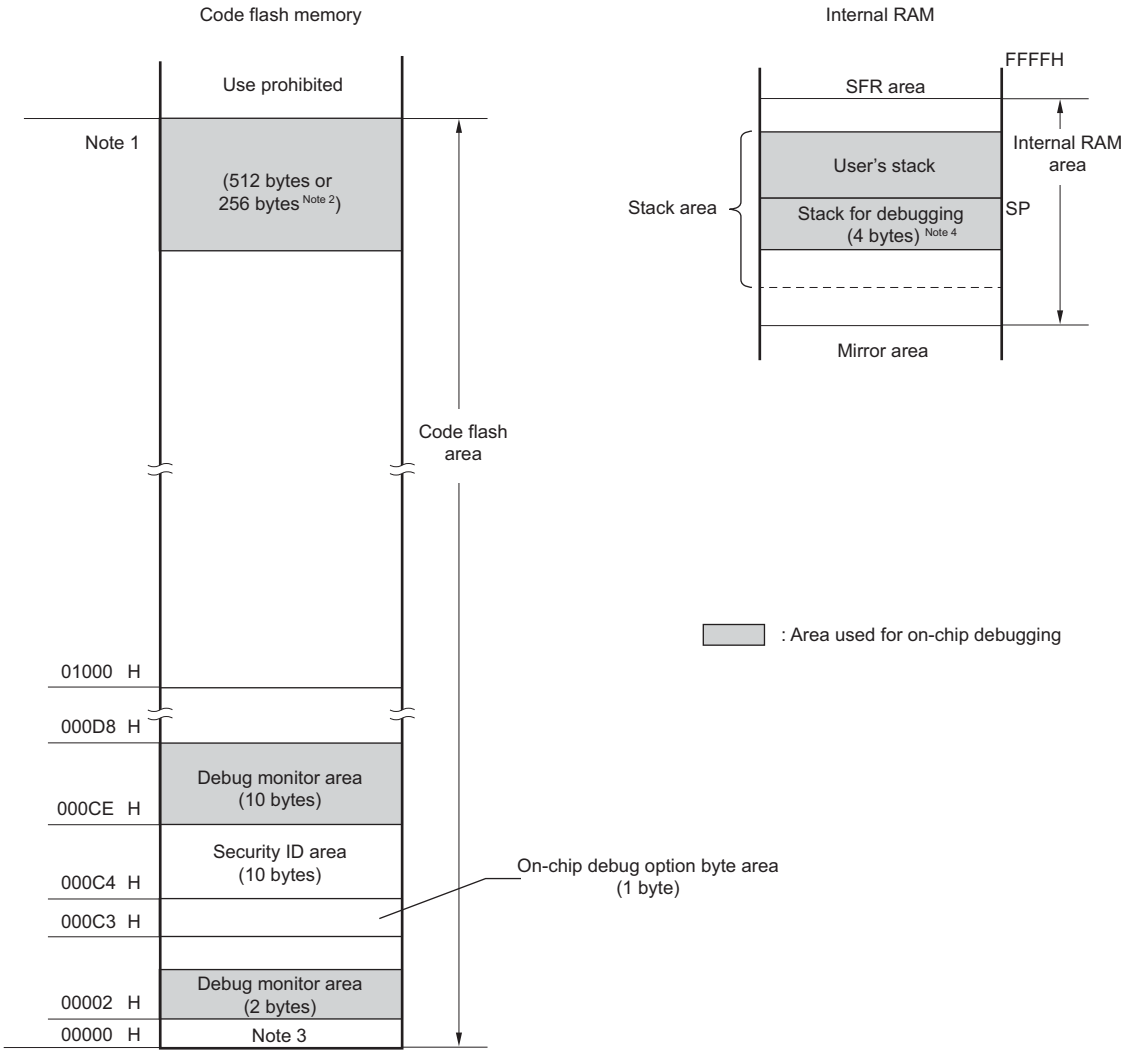
To perform communication between the RL78/F13 or RL78/F14 and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using linker options.

### 31.4.1 Securement of memory space

The shaded portions in Figure 31-2 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

**Figure 31-3. Memory Spaces Where Debug Monitor Programs Are Allocated**



**Notes 1.** Address differs depending on products as follows.

| Products (code flash memory capacity) | Address of **Note 1** |
|---|---|
| R5F10AxA (x = 6, A, B, G) | 03FFFH |
| R5F10AxC (x = 6, A, B, G, L) | 07FFFH |
| R5F10AxD (x = 6, A, B, G, L) | 0BFFFH |
| R5F10AxE (x = 6, A, B, G, L, M) | 0FFFFH |
| R5F10AxF (x = G, L, M) | 17FFFH |
| R5F10AxG (x = G, L, M) | 1FFFFH |
| R5F10BxC (x = A, B, G, L) | 07FFFH |
| R5F10BxD (x = A, B, G, L) | 0BFFFH |
| R5F10BxE (x = A, B, G, L, M) | 0FFFFH |
| R5F10BxF (x = A, B, G, L, M) | 17FFFH |
| R5F10BxG (x = A, B, G, L, M) | 1FFFFH |
| R5F10PxD (x = A, B, G) | 0BFFFH |
| R5F10PxE (x = A, B, G, L, M, P) | 0FFFFH |
| R5F10PxF (x = G, L, M, P) | 17FFFH |
| R5F10PxG (x = G, L, M, P) | 1FFFFH |
| R5F10PxH (x = G, L, M, P) | 2FFFFH |
| R5F10PxJ (x = G, L, M, P) | 3FFFFH |

**2.** When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

**3.** In debugging, reset vector is rewritten to address allocated to a monitor program.

**4.** Since this area is allocated just under the stack area, the address of this area varies depending on the stack increase and decrease. That is, 4 extra bytes are consumed for the stack area used.
When using self-programming, 12 extra bytes are consumed for the stack area used.

# CHAPTER 32 BCD CORRECTION CIRCUIT

## 32.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.

The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/ subtracting the BCD correction result register (BCDADJ).

## 32.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

### (1) BCD correction result register (BCDADJ)

The BCDADJ register stores correction values for obtaining the add/subtract result as BCD code through add/subtract instructions using the A register as the operand.

The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags.

The BCDADJ register is read by an 8-bit memory manipulation instruction.

Reset input sets this register to undefined.

**Figure 32-1. Format of BCD Correction Result Register (BCDADJ)**

Address: F00FEH After reset: undefined R

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| BCDADJ | | | | | | | | |

## 32.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

**(1) Addition: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code value**

<1> The BCD code value to which addition is performed is stored in the A register.

<2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by adding in binary the value of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

**Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.**

An example is shown below.

Examples 1: 99 + 89 = 188

| Instruction | A Register | CY Flag | AC Flag | BCDADJ Register |
|---|---|---|---|---|
| `MOV A, #99H ; <1>` | 99H | – | – | – |
| `ADD A, #89H ; <2>` | 22H | 1 | 1 | 66H |
| `ADD A, !BCDADJ ; <3>` | 88H | 1 | 0 | – |

Examples 2: 85 + 15 = 100

| Instruction | A Register | CY Flag | AC Flag | BCDADJ Register |
|---|---|---|---|---|
| `MOV A, #85H ; <1>` | 85H | – | – | – |
| `ADD A, #15H ; <2>` | 9AH | 0 | 0 | 66H |
| `ADD A, !BCDADJ ; <3>` | 00H | 1 | 1 | – |

Examples 3: 80 + 80 = 160

| Instruction | A Register | CY Flag | AC Flag | BCDADJ Register |
|---|---|---|---|---|
| `MOV A, #80H ; <1>` | 80H | – | – | – |
| `ADD A, #80H ; <2>` | 00H | 1 | 0 | 60H |
| `ADD A, !BCDADJ ; <3>` | 60H | 1 | 0 | – |

**(2) Subtraction: Calculating the result of subtracting a BCD code value from another BCD code value by using a BCD code value**

<1> The BCD code value from which subtraction is performed is stored in the A register.

<2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

**Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.**

An example is shown below.

Example: 91 − 52 = 39

| Instruction | A Register | CY Flag | AC Flag | BCDADJ Register |
|---|---|---|---|---|
| `MOV A, #91H ; <1>` | 91H | − | − | − |
| `SUB A, #52H ; <2>` | 3FH | 0 | 1 | 06H |
| `SUB A, !BCDADJ ; <3>` | 39H | 0 | 0 | − |

# CHAPTER 33 INSTRUCTION SET

This chapter lists the instructions in the RL78 microcontroller instruction set. For details of each operation and operation code, refer to the separate document **RL78 Family User's Manual Software**.

## 33.1 Conventions Used in Operation List

### 33.1.1 Operand identifiers and specification methods

Operands are described in the “Operand” column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Alphabetic letters in capitals and the symbols, #, !, !!, $, $!, [ ], and ES: are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: 16-bit absolute address specification
- !!: 20-bit absolute address specification
- $: 8-bit relative address specification
- $!: 16-bit relative address specification
- [ ]: Indirect address specification
- ES:: Extension address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, !!, $, $!, [ ], and ES: symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

**Table 33-1. Operand Identifiers and Specification Methods**

| Identifier | Description Method |
|---|---|
| r<br>rp<br>sfr<br>sfrp | X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)<br>AX (RP0), BC (RP1), DE (RP2), HL (RP3)<br>Special-function register symbol (SFR symbol) FFF00H to FFFFFH<br>Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only[Note]) FFF00H to FFFFFH |
| saddr<br>saddrp | FFE20H to FFF1FH Immediate data or labels<br>FFE20H to FF1FH Immediate data or labels (even addresses only[Note]) |
| addr20<br>addr16<br>addr5 | 00000H to FFFFFH Immediate data or labels<br>0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions[Note])<br>0080H to 00BFH Immediate data or labels (even addresses only) |
| word<br>byte<br>bit | 16-bit immediate data or label<br>8-bit immediate data or label<br>3-bit immediate data or label |
| RBn | RB0 to RB3 |

**Note** Bit 0 = 0 when an odd address is specified.

**Remark** The special function registers can be described to operand sfr as symbols. See **Table 3-5 SFR List** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Table 3-6 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

### 33.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

**Table 33-2. Symbols in “Operation” Column**

| Symbol | Function |
|---|---|
| A | A register; 8-bit accumulator |
| X | X register |
| B | B register |
| C | C register |
| D | D register |
| E | E register |
| H | H register |
| L | L register |
| ES | ES register |
| CS | CS register |
| AX | AX register pair; 16-bit accumulator |
| BC | BC register pair |
| DE | DE register pair |
| HL | HL register pair |
| PC | Program counter |
| SP | Stack pointer |
| PSW | Program status word |
| CY | Carry flag |
| AC | Auxiliary carry flag |
| Z | Zero flag |
| RBS | Register bank select flag |
| IE | Interrupt request enable flag |
| () | Memory contents indicated by address or register contents in parentheses |
| $X_H$, $X_L$<br>$X_S$, $X_H$, $X_L$ | 16-bit registers: $X_H$ = higher 8 bits, $X_L$ = lower 8 bits<br>20-bit registers: $X_S$ = (bits 19 to 16), $X_H$ = (bits 15 to 8), $X_L$ = (bits 7 to 0) |
| ∧ | Logical product (AND) |
| ∨ | Logical sum (OR) |
| ∨̶ | Exclusive logical sum (exclusive OR) |
| ¯ | Inverted data |
| addr5 | 16-bit immediate data (even addresses only in 0080H to 00BFH) |
| addr16 | 16-bit immediate data |
| addr20 | 20-bit immediate data |
| jdisp8 | Signed 8-bit data (displacement value) |
| jdisp16 | Signed 16-bit data (displacement value) |

### 33.1.3 Description of flag operation column

The change of the flag value when the instruction is executed is shown in the “Flag” column using the following symbols.

**Table 33-3. Symbols in “Flag” Column**

| Symbol | Change of Flag Value |
|---|---|
| (Blank) | Unchanged |
| 0 | Cleared to 0 |
| 1 | Set to 1 |
| × | Set/cleared according to the result |
| R | Previously saved value is restored |

### 33.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

A interrupt and DTC transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

**Table 33-4. Use Example of PREFIX Operation Code**

| Instruction | Opcode | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| MOV !addr16, #byte | CFH | !addr16 | | #byte | – |
| MOV ES:!addr16, #byte | 11H | CFH | !addr16 | | #byte |
| MOV A, [HL] | 8BH | – | – | – | – |
| MOV A, ES:[HL] | 11H | 8BH | – | – | – |

**Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.**

## 33.2 Operation List

**Table 33-5. Operation List (1/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z | AC | CY |
| 8-bit data transfer | MOV | r, #byte | 2 | 1 | – | r ← byte | | | |
| | | PSW, #byte | 3 | 3 | – | PSW ← byte | × | × | × |
| | | CS, #byte | 3 | 1 | – | CS ← byte | | | |
| | | ES, #byte | 2 | 1 | – | ES ← byte | | | |
| | | !addr16, #byte | 4 | 1 | – | (addr16) ← byte | | | |
| | | ES:!addr16, #byte | 5 | 2 | – | (ES, addr16) ← byte | | | |
| | | saddr, #byte | 3 | 1 | – | (saddr) ← byte | | | |
| | | sfr, #byte | 3 | 1 | – | sfr ← byte | | | |
| | | [DE+byte], #byte | 3 | 1 | – | (DE+byte) ← byte | | | |
| | | ES:[DE+byte],#byte | 4 | 2 | – | ((ES, DE)+byte) ← byte | | | |
| | | [HL+byte], #byte | 3 | 1 | – | (HL+byte) ← byte | | | |
| | | ES:[HL+byte],#byte | 4 | 2 | – | ((ES, HL)+byte) ← byte | | | |
| | | [SP+byte], #byte | 3 | 1 | – | (SP+byte) ← byte | | | |
| | | word[B], #byte | 4 | 1 | – | (B+word) ← byte | | | |
| | | ES:word[B], #byte | 5 | 2 | – | ((ES, B)+word) ← byte | | | |
| | | word[C], #byte | 4 | 1 | – | (C+word) ← byte | | | |
| | | ES:word[C], #byte | 5 | 2 | – | ((ES, C)+word) ← byte | | | |
| | | word[BC], #byte | 4 | 1 | – | (BC+word) ← byte | | | |
| | | ES:word[BC], #byte | 5 | 2 | – | ((ES, BC)+word) ← byte | | | |
| | | A, r Note 3 | 1 | 1 | – | A ← r | | | |
| | | r, A Note 3 | 1 | 1 | – | r ← A | | | |
| | | A, PSW | 2 | 1 | – | A ← PSW | | | |
| | | PSW, A | 2 | 3 | – | PSW ← A | × | × | × |
| | | A, CS | 2 | 1 | – | A ← CS | | | |
| | | CS, A | 2 | 1 | – | CS ← A | | | |
| | | A, ES | 2 | 1 | – | A ← ES | | | |
| | | ES, A | 2 | 1 | – | ES ← A | | | |
| | | A, !addr16 | 3 | 1 | 4 | A ← (addr16) | | | |
| | | A, ES:!addr16 | 4 | 2 | 5 | A ← (ES, addr16) | | | |
| | | !addr16, A | 3 | 1 | – | (addr16) ← A | | | |
| | | ES:!addr16, A | 4 | 2 | – | (ES, addr16) ← A | | | |
| | | A, saddr | 2 | 1 | – | A ← (saddr) | | | |
| | | saddr, A | 2 | 1 | – | (saddr) ← A | | | |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**3.** Except r = A

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (2/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag |
|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z AC CY |
| 8-bit data transfer | MOV | A, sfr | 2 | 1 | – | A ← sfr | |
| | | sfr, A | 2 | 1 | – | sfr ← A | |
| | | A, [DE] | 1 | 1 | 4 | A ← (DE) | |
| | | [DE], A | 1 | 1 | – | (DE) ← A | |
| | | A, ES:[DE] | 2 | 2 | 5 | A ← (ES, DE) | |
| | | ES:[DE], A | 2 | 2 | – | (ES, DE) ← A | |
| | | A, [HL] | 1 | 1 | 4 | A ← (HL) | |
| | | [HL], A | 1 | 1 | – | (HL) ← A | |
| | | A, ES:[HL] | 2 | 2 | 5 | A ← (ES, HL) | |
| | | ES:[HL], A | 2 | 2 | – | (ES, HL) ← A | |
| | | A, [DE+byte] | 2 | 1 | 4 | A ← (DE + byte) | |
| | | [DE+byte], A | 2 | 1 | – | (DE + byte) ← A | |
| | | A, ES:[DE+byte] | 3 | 2 | 5 | A ← ((ES, DE) + byte) | |
| | | ES:[DE+byte], A | 3 | 2 | – | ((ES, DE) + byte) ← A | |
| | | A, [HL+byte] | 2 | 1 | 4 | A ← (HL + byte) | |
| | | [HL+byte], A | 2 | 1 | – | (HL + byte) ← A | |
| | | A, ES:[HL+byte] | 3 | 2 | 5 | A ← ((ES, HL) + byte) | |
| | | ES:[HL+byte], A | 3 | 2 | – | ((ES, HL) + byte) ← A | |
| | | A, [SP+byte] | 2 | 1 | – | A ← (SP + byte) | |
| | | [SP+byte], A | 2 | 1 | – | (SP + byte) ← A | |
| | | A, word[B] | 3 | 1 | 4 | A ← (B + word) | |
| | | word[B], A | 3 | 1 | – | (B + word) ← A | |
| | | A, ES:word[B] | 4 | 2 | 5 | A ← ((ES, B) + word) | |
| | | ES:word[B], A | 4 | 2 | – | ((ES, B) + word) ← A | |
| | | A, word[C] | 3 | 1 | 4 | A ← (C + word) | |
| | | word[C], A | 3 | 1 | – | (C + word) ← A | |
| | | A, ES:word[C] | 4 | 2 | 5 | A ← ((ES, C) + word) | |
| | | ES:word[C], A | 4 | 2 | – | ((ES, C) + word) ← A | |
| | | A, word[BC] | 3 | 1 | 4 | A ← (BC + word) | |
| | | word[BC], A | 3 | 1 | – | (BC + word) ← A | |
| | | A, ES:word[BC] | 4 | 2 | 5 | A ← ((ES, BC) + word) | |
| | | ES:word[BC], A | 4 | 2 | – | ((ES, BC) + word) ← A | |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (3/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag |
|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z AC CY |
| 8-bit data transfer | MOV | A, [HL+B] | 2 | 1 | 4 | A ← (HL + B) | |
| | | [HL+B], A | 2 | 1 | – | (HL + B) ← A | |
| | | A, ES:[HL+B] | 3 | 2 | 5 | A ← ((ES, HL) + B) | |
| | | ES:[HL+B], A | 3 | 2 | – | ((ES, HL) + B) ← A | |
| | | A, [HL+C] | 2 | 1 | 4 | A ← (HL + C) | |
| | | [HL+C], A | 2 | 1 | – | (HL + C) ← A | |
| | | A, ES:[HL+C] | 3 | 2 | 5 | A ← ((ES, HL) + C) | |
| | | ES:[HL+C], A | 3 | 2 | – | ((ES, HL) + C) ← A | |
| | | X, !addr16 | 3 | 1 | 4 | X ← (addr16) | |
| | | X, ES:!addr16 | 4 | 2 | 5 | X ← (ES, addr16) | |
| | | X, saddr | 2 | 1 | – | X ← (saddr) | |
| | | B, !addr16 | 3 | 1 | 4 | B ← (addr16) | |
| | | B, ES:!addr16 | 4 | 2 | 5 | B ← (ES, addr16) | |
| | | B, saddr | 2 | 1 | – | B ← (saddr) | |
| | | C, !addr16 | 3 | 1 | 4 | C ← (addr16) | |
| | | C, ES:!addr16 | 4 | 2 | 5 | C ← (ES, addr16) | |
| | | C, saddr | 2 | 1 | – | C ← (saddr) | |
| | | ES, saddr | 3 | 1 | – | ES ← (saddr) | |
| | XCH | A, r Note 3 | 1 (r = X)<br>2 (other than r = X) | 1 | – | A ←→ r | |
| | | A, !addr16 | 4 | 2 | – | A ←→ (addr16) | |
| | | A, ES:!addr16 | 5 | 3 | – | A ←→ (ES, addr16) | |
| | | A, saddr | 3 | 2 | – | A ←→ (saddr) | |
| | | A, sfr | 3 | 2 | – | A ←→ sfr | |
| | | A, [DE] | 2 | 2 | – | A ←→ (DE) | |
| | | A, ES:[DE] | 3 | 3 | – | A ←→ (ES, DE) | |
| | | A, [HL] | 2 | 2 | – | A ←→ (HL) | |
| | | A, ES:[HL] | 3 | 3 | – | A ←→ (ES, HL) | |
| | | A, [DE+byte] | 3 | 2 | – | A ←→ (DE + byte) | |
| | | A, ES:[DE+byte] | 4 | 3 | – | A ←→ ((ES, DE) + byte) | |
| | | A, [HL+byte] | 3 | 2 | – | A ←→ (HL + byte) | |
| | | A, ES:[HL+byte] | 4 | 3 | – | A ←→ ((ES, HL) + byte) | |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**3.** Except r = A

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (4/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z | AC | CY |
| 8-bit data transfer | XCH | A, [HL+B] | 2 | 2 | – | A ←→ (HL+B) | | | |
| | | A, ES:[HL+B] | 3 | 3 | – | A ←→ ((ES, HL)+B) | | | |
| | | A, [HL+C] | 2 | 2 | – | A ←→ (HL+C) | | | |
| | | A, ES:[HL+C] | 3 | 3 | – | A ←→ ((ES, HL)+C) | | | |
| | ONEB | A | 1 | 1 | – | A ← 01H | | | |
| | | X | 1 | 1 | – | X ← 01H | | | |
| | | B | 1 | 1 | – | B ← 01H | | | |
| | | C | 1 | 1 | – | C ← 01H | | | |
| | | !addr16 | 3 | 1 | – | (addr16) ← 01H | | | |
| | | ES:!addr16 | 4 | 2 | – | (ES, addr16) ← 01H | | | |
| | | saddr | 2 | 1 | – | (saddr) ← 01H | | | |
| | CLRB | A | 1 | 1 | – | A ← 00H | | | |
| | | X | 1 | 1 | – | X ← 00H | | | |
| | | B | 1 | 1 | – | B ← 00H | | | |
| | | C | 1 | 1 | – | C ← 00H | | | |
| | | !addr16 | 3 | 1 | – | (addr16) ← 00H | | | |
| | | ES:!addr16 | 4 | 2 | – | (ES,addr16) ← 00H | | | |
| | | saddr | 2 | 1 | – | (saddr) ← 00H | | | |
| | MOVS | [HL+byte], X | 3 | 1 | – | (HL+byte) ← X | × | | × |
| | | ES:[HL+byte], X | 4 | 2 | – | (ES, HL+byte) ← X | × | | × |
| 16-bit data transfer | MOVW | rp, #word | 3 | 1 | – | rp ← word | | | |
| | | saddrp, #word | 4 | 1 | – | (saddrp) ← word | | | |
| | | sfrp, #word | 4 | 1 | – | sfrp ← word | | | |
| | | AX, rp Note 3 | 1 | 1 | – | AX ← rp | | | |
| | | rp, AX Note 3 | 1 | 1 | – | rp ← AX | | | |
| | | AX, !addr16 | 3 | 1 | 4 | AX ← (addr16) | | | |
| | | !addr16, AX | 3 | 1 | – | (addr16) ← AX | | | |
| | | AX, ES:!addr16 | 4 | 2 | 5 | AX ← (ES, addr16) | | | |
| | | ES:!addr16, AX | 4 | 2 | – | (ES, addr16) ← AX | | | |
| | | AX, saddrp | 2 | 1 | – | AX ← (saddrp) | | | |
| | | saddrp, AX | 2 | 1 | – | (saddrp) ← AX | | | |
| | | AX, sfrp | 2 | 1 | – | AX ← sfrp | | | |
| | | sfrp, AX | 2 | 1 | – | sfrp ← AX | | | |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**3.** Except rp = AX

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (5/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag |
|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z AC CY |
| 16-bit data transfer | MOVW | AX, [DE] | 1 | 1 | 4 | AX ← (DE) | |
| | | [DE], AX | 1 | 1 | – | (DE) ← AX | |
| | | AX, ES:[DE] | 2 | 2 | 5 | AX ← (ES, DE) | |
| | | ES:[DE], AX | 2 | 2 | – | (ES, DE) ← AX | |
| | | AX, [HL] | 1 | 1 | 4 | AX ← (HL) | |
| | | [HL], AX | 1 | 1 | – | (HL) ← AX | |
| | | AX, ES:[HL] | 2 | 2 | 5 | AX ← (ES, HL) | |
| | | ES:[HL], AX | 2 | 2 | – | (ES, HL) ← AX | |
| | | AX, [DE+byte] | 2 | 1 | 4 | AX ← (DE+byte) | |
| | | [DE+byte], AX | 2 | 1 | – | (DE+byte) ← AX | |
| | | AX, ES:[DE+byte] | 3 | 2 | 5 | AX ← ((ES, DE) + byte) | |
| | | ES:[DE+byte], AX | 3 | 2 | – | ((ES, DE) + byte) ← AX | |
| | | AX, [HL+byte] | 2 | 1 | 4 | AX ← (HL + byte) | |
| | | [HL+byte], AX | 2 | 1 | – | (HL + byte) ← AX | |
| | | AX, ES:[HL+byte] | 3 | 2 | 5 | AX ← ((ES, HL) + byte) | |
| | | ES:[HL+byte], AX | 3 | 2 | – | ((ES, HL) + byte) ← AX | |
| | | AX, [SP+byte] | 2 | 1 | – | AX ← (SP + byte) | |
| | | [SP+byte], AX | 2 | 1 | – | (SP + byte) ← AX | |
| | | AX, word[B] | 3 | 1 | 4 | AX ← (B + word) | |
| | | word[B], AX | 3 | 1 | – | (B+ word) ← AX | |
| | | AX, ES:word[B] | 4 | 2 | 5 | AX ← ((ES, B) + word) | |
| | | ES:word[B], AX | 4 | 2 | – | ((ES, B) + word) ← AX | |
| | | AX, word[C] | 3 | 1 | 4 | AX ← (C + word) | |
| | | word[C], AX | 3 | 1 | – | (C + word) ← AX | |
| | | AX, ES:word[C] | 4 | 2 | 5 | AX ← ((ES, C) + word) | |
| | | ES:word[C], AX | 4 | 2 | – | ((ES, C) + word) ← AX | |
| | | AX, word[BC] | 3 | 1 | 4 | AX ← (BC + word) | |
| | | word[BC], AX | 3 | 1 | – | (BC + word) ← AX | |
| | | AX, ES:word[BC] | 4 | 2 | 5 | AX ← ((ES, BC) + word) | |
| | | ES:word[BC], AX | 4 | 2 | – | ((ES, BC) + word) ← AX | |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (6/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z | AC | CY |
| 16-bit data transfer | MOVW | BC, !addr16 | 3 | 1 | 4 | BC ← (addr16) | | | |
| | | BC, ES:!addr16 | 4 | 2 | 5 | BC ← (ES, addr16) | | | |
| | | DE, !addr16 | 3 | 1 | 4 | DE ← (addr16) | | | |
| | | DE, ES:!addr16 | 4 | 2 | 5 | DE ← (ES, addr16) | | | |
| | | HL, !addr16 | 3 | 1 | 4 | HL ← (addr16) | | | |
| | | HL, ES:!addr16 | 4 | 2 | 5 | HL ← (ES, addr16) | | | |
| | | BC, saddrp | 2 | 1 | – | BC ← (saddrp) | | | |
| | | DE, saddrp | 2 | 1 | – | DE ← (saddrp) | | | |
| | | HL, saddrp | 2 | 1 | – | HL ← (saddrp) | | | |
| | XCHW | AX, rp Note 3 | 1 | 1 | – | AX ←→ rp | | | |
| | ONEW | AX | 1 | 1 | – | AX ← 0001H | | | |
| | | BC | 1 | 1 | – | BC ← 0001H | | | |
| | CLRW | AX | 1 | 1 | – | AX ← 0000H | | | |
| | | BC | 1 | 1 | – | BC ← 0000H | | | |
| 8-bit operation | ADD | A, #byte | 2 | 1 | – | A, CY ← A + byte | × | × | × |
| | | saddr, #byte | 3 | 2 | – | (saddr), CY ← (saddr)+byte | × | × | × |
| | | A, r Note 4 | 2 | 1 | – | A, CY ← A + r | × | × | × |
| | | r, A | 2 | 1 | – | r, CY ← r + A | × | × | × |
| | | A, !addr16 | 3 | 1 | 4 | A, CY ← A + (addr16) | × | × | × |
| | | A, ES:!addr16 | 4 | 2 | 5 | A, CY ← A + (ES, addr16) | × | × | × |
| | | A, saddr | 2 | 1 | – | A, CY ← A + (saddr) | × | × | × |
| | | A, [HL] | 1 | 1 | 4 | A, CY ← A+ (HL) | × | × | × |
| | | A, ES:[HL] | 2 | 2 | 5 | A,CY ← A + (ES, HL) | × | × | × |
| | | A, [HL+byte] | 2 | 1 | 4 | A, CY ← A + (HL+byte) | × | × | × |
| | | A, ES:[HL+byte] | 3 | 2 | 5 | A,CY ← A + ((ES, HL)+byte) | × | × | × |
| | | A, [HL+B] | 2 | 1 | 4 | A, CY ← A + (HL+B) | × | × | × |
| | | A, ES:[HL+B] | 3 | 2 | 5 | A,CY ← A+((ES, HL)+B) | × | × | × |
| | | A, [HL+C] | 2 | 1 | 4 | A, CY ← A + (HL+C) | × | × | × |
| | | A, ES:[HL+C] | 3 | 2 | 5 | A,CY ← A + ((ES, HL) + C) | × | × | × |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**3.** Except rp = AX

**4.** Except r = A

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (7/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z | AC | CY |
| 8-bit operation | ADDC | A, #byte | 2 | 1 | – | A, CY ← A+byte+CY | × | × | × |
| | | saddr, #byte | 3 | 2 | – | (saddr), CY ← (saddr) +byte+CY | × | × | × |
| | | A, rv Note 3 | 2 | 1 | – | A, CY ← A + r + CY | × | × | × |
| | | r, A | 2 | 1 | – | r, CY ← r + A + CY | × | × | × |
| | | A, !addr16 | 3 | 1 | 4 | A, CY ← A + (addr16)+CY | × | × | × |
| | | A, ES:!addr16 | 4 | 2 | 5 | A, CY ← A + (ES, addr16)+CY | × | × | × |
| | | A, saddr | 2 | 1 | – | A, CY ← A + (saddr)+CY | × | × | × |
| | | A, [HL] | 1 | 1 | 4 | A, CY ← A+ (HL) + CY | × | × | × |
| | | A, ES:[HL] | 2 | 2 | 5 | A,CY ← A+ (ES, HL) + CY | × | × | × |
| | | A, [HL+byte] | 2 | 1 | 4 | A, CY ← A+ (HL+byte) + CY | × | × | × |
| | | A, ES:[HL+byte] | 3 | 2 | 5 | A,CY ← A+ ((ES, HL)+byte) + CY | × | × | × |
| | | A, [HL+B] | 2 | 1 | 4 | A, CY ← A+ (HL+B) +CY | × | × | × |
| | | A, ES:[HL+B] | 3 | 2 | 5 | A,CY ← A+((ES, HL)+B)+CY | × | × | × |
| | | A, [HL+C] | 2 | 1 | 4 | A, CY ← A+ (HL+C)+CY | × | × | × |
| | | A, ES:[HL+C] | 3 | 2 | 5 | A,CY ← A+ ((ES, HL)+C)+CY | × | × | × |
| | SUB | A, #byte | 2 | 1 | – | A, CY ← A – byte | × | × | × |
| | | saddr, #byte | 3 | 2 | – | (saddr), CY ← (saddr) – byte | × | × | × |
| | | A, r Note 3 | 2 | 1 | – | A, CY ← A – r | × | × | × |
| | | r, A | 2 | 1 | – | r, CY ← r – A | × | × | × |
| | | A, !addr16 | 3 | 1 | 4 | A, CY ← A – (addr16) | × | × | × |
| | | A, ES:!addr16 | 4 | 2 | 5 | A, CY ← A – (ES, addr16) | × | × | × |
| | | A, saddr | 2 | 1 | – | A, CY ← A – (saddr) | × | × | × |
| | | A, [HL] | 1 | 1 | 4 | A, CY ← A – (HL) | × | × | × |
| | | A, ES:[HL] | 2 | 2 | 5 | A,CY ← A – (ES, HL) | × | × | × |
| | | A, [HL+byte] | 2 | 1 | 4 | A, CY ← A – (HL+byte) | × | × | × |
| | | A, ES:[HL+byte] | 3 | 2 | 5 | A,CY ← A – ((ES, HL)+byte) | × | × | × |
| | | A, [HL+B] | 2 | 1 | 4 | A, CY ← A – (HL+B) | × | × | × |
| | | A, ES:[HL+B] | 3 | 2 | 5 | A,CY ← A – ((ES, HL)+B) | × | × | × |
| | | A, [HL+C] | 2 | 1 | 4 | A, CY ← A – (HL+C) | × | × | × |
| | | A, ES:[HL+C] | 3 | 2 | 5 | A,CY ← A – ((ES, HL)+C) | × | × | × |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**3.** Except r = A

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (8/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z | AC | CY |
| 8-bit operation | SUBC | A, #byte | 2 | 1 | – | A, CY ← A – byte – CY | × | × | × |
| | | saddr, #byte | 3 | 2 | – | (saddr), CY ← (saddr) – byte – CY | × | × | × |
| | | A, r Note 3 | 2 | 1 | – | A, CY ← A – r – CY | × | × | × |
| | | r, A | 2 | 1 | – | r, CY ← r – A – CY | × | × | × |
| | | A, !addr16 | 3 | 1 | 4 | A, CY ← A – (addr16) – CY | × | × | × |
| | | A, ES:!addr16 | 4 | 2 | 5 | A, CY ← A – (ES, addr16) – CY | × | × | × |
| | | A, saddr | 2 | 1 | – | A, CY ← A – (saddr) – CY | × | × | × |
| | | A, [HL] | 1 | 1 | 4 | A, CY ← A – (HL) – CY | × | × | × |
| | | A, ES:[HL] | 2 | 2 | 5 | A,CY ← A – (ES, HL) – CY | × | × | × |
| | | A, [HL+byte] | 2 | 1 | 4 | A, CY ← A – (HL+byte) – CY | × | × | × |
| | | A, ES:[HL+byte] | 3 | 2 | 5 | A,CY ← A – ((ES, HL)+byte) – CY | × | × | × |
| | | A, [HL+B] | 2 | 1 | 4 | A, CY ← A – (HL+B) – CY | × | × | × |
| | | A, ES:[HL+B] | 3 | 2 | 5 | A,CY ← A – ((ES, HL)+B) – CY | × | × | × |
| | | A, [HL+C] | 2 | 1 | 4 | A, CY ← A – (HL+C) – CY | × | × | × |
| | | A, ES:[HL+C] | 3 | 2 | 5 | A, CY ← A – ((ES:HL)+C) – CY | × | × | × |
| | AND | A, #byte | 2 | 1 | – | A ← A ∧ byte | × | | |
| | | saddr, #byte | 3 | 2 | – | (saddr) ← (saddr) ∧ byte | × | | |
| | | A, r Note 3 | 2 | 1 | – | A ← A ∧ r | × | | |
| | | r, A | 2 | 1 | – | R ← r ∧ A | × | | |
| | | A, !addr16 | 3 | 1 | 4 | A ← A ∧ (addr16) | × | | |
| | | A, ES:!addr16 | 4 | 2 | 5 | A ← A ∧ (ES:addr16) | × | | |
| | | A, saddr | 2 | 1 | – | A ← A ∧ (saddr) | × | | |
| | | A, [HL] | 1 | 1 | 4 | A ← A ∧ (HL) | × | | |
| | | A, ES:[HL] | 2 | 2 | 5 | A ← A ∧ (ES:HL) | × | | |
| | | A, [HL+byte] | 2 | 1 | 4 | A ← A ∧ (HL+byte) | × | | |
| | | A, ES:[HL+byte] | 3 | 2 | 5 | A ← A ∧ ((ES:HL)+byte) | × | | |
| | | A, [HL+B] | 2 | 1 | 4 | A ← A ∧ (HL+B) | × | | |
| | | A, ES:[HL+B] | 3 | 2 | 5 | A ← A ∧ ((ES:HL)+B) | × | | |
| | | A, [HL+C] | 2 | 1 | 4 | A ← A ∧ (HL+C) | × | | |
| | | A, ES:[HL+C] | 3 | 2 | 5 | A ← A ∧ ((ES:HL)+C) | × | | |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**3.** Except r = A

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (9/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag |
|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z AC CY |
| 8-bit operation | OR | A, #byte | 2 | 1 | – | A ← A∨byte | × |
| | | saddr, #byte | 3 | 2 | – | (saddr) ← (saddr)∨byte | × |
| | | A, r Note 3 | 2 | 1 | – | A ← A∨r | × |
| | | r, A | 2 | 1 | – | r ← r∨A | × |
| | | A, !addr16 | 3 | 1 | 4 | A ← A∨(addr16) | × |
| | | A, ES:!addr16 | 4 | 2 | 5 | A ← A∨(ES:addr16) | × |
| | | A, saddr | 2 | 1 | – | A ← A∨(saddr) | × |
| | | A, [HL] | 1 | 1 | 4 | A ← A∨(H) | × |
| | | A, ES:[HL] | 2 | 2 | 5 | A ← A∨(ES:HL) | × |
| | | A, [HL+byte] | 2 | 1 | 4 | A ← A∨(HL+byte) | × |
| | | A, ES:[HL+byte] | 3 | 2 | 5 | A ← A∨((ES:HL)+byte) | × |
| | | A, [HL+B] | 2 | 1 | 4 | A ← A∨(HL+B) | × |
| | | A, ES:[HL+B] | 3 | 2 | 5 | A ← A∨((ES:HL)+B) | × |
| | | A, [HL+C] | 2 | 1 | 4 | A ← A∨(HL+C) | × |
| | | A, ES:[HL+C] | 3 | 2 | 5 | A ← A∨((ES:HL)+C) | × |
| | XOR | A, #byte | 2 | 1 | – | A ← A⊻byte | × |
| | | saddr, #byte | 3 | 2 | – | (saddr) ← (saddr)⊻byte | × |
| | | A, r Note 3 | 2 | 1 | – | A ← A⊻r | × |
| | | r, A | 2 | 1 | – | r ← r⊻A | × |
| | | A, !addr16 | 3 | 1 | 4 | A ← A⊻(addr16) | × |
| | | A, ES:!addr16 | 4 | 2 | 5 | A ← A⊻(ES:addr16) | × |
| | | A, saddr | 2 | 1 | – | A ← A⊻(saddr) | × |
| | | A, [HL] | 1 | 1 | 4 | A ← A⊻(HL) | × |
| | | A, ES:[HL] | 2 | 2 | 5 | A ← A⊻(ES:HL) | × |
| | | A, [HL+byte] | 2 | 1 | 4 | A ← A⊻(HL+byte) | × |
| | | A, ES:[HL+byte] | 3 | 2 | 5 | A ← A⊻((ES:HL)+byte) | × |
| | | A, [HL+B] | 2 | 1 | 4 | A ← A⊻(HL+B) | × |
| | | A, ES:[HL+B] | 3 | 2 | 5 | A ← A⊻((ES:HL)+B) | × |
| | | A, [HL+C] | 2 | 1 | 4 | A ← A⊻(HL+C) | × |
| | | A, ES:[HL+C] | 3 | 2 | 5 | A ← A⊻((ES:HL)+C) | × |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**3.** Except r = A

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (10/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z | AC | CY |
| 8-bit operation | CMP | A, #byte | 2 | 1 | – | A – byte | × | × | × |
| | | !addr16, #byte | 4 | 1 | 4 | (addr16) – byte | × | × | × |
| | | ES:!addr16, #byte | 5 | 2 | 5 | (ES:addr16) – byte | × | × | × |
| | | saddr, #byte | 3 | 1 | – | (saddr) – byte | × | × | × |
| | | A, r Note3 | 2 | 1 | – | A – r | × | × | × |
| | | r, A | 2 | 1 | – | r – A | × | × | × |
| | | A, !addr16 | 3 | 1 | 4 | A – (addr16) | × | × | × |
| | | A, ES:!addr16 | 4 | 2 | 5 | A – (ES:addr16) | × | × | × |
| | | A, saddr | 2 | 1 | – | A – (saddr) | × | × | × |
| | | A, [HL] | 1 | 1 | 4 | A – (HL) | × | × | × |
| | | A, ES:[HL] | 2 | 2 | 5 | A – (ES:HL) | × | × | × |
| | | A, [HL+byte] | 2 | 1 | 4 | A – (HL+byte) | × | × | × |
| | | A, ES:[HL+byte] | 3 | 2 | 5 | A – ((ES:HL)+byte) | × | × | × |
| | | A, [HL+B] | 2 | 1 | 4 | A – (HL+B) | × | × | × |
| | | A, ES:[HL+B] | 3 | 2 | 5 | A – ((ES:HL)+B) | × | × | × |
| | | A, [HL+C] | 2 | 1 | 4 | A – (HL+C) | × | × | × |
| | | A, ES:[HL+C] | 3 | 2 | 5 | A – ((ES:HL)+C) | × | × | × |
| | CMP0 | A | 1 | 1 | – | A – 00H | × | × | × |
| | | X | 1 | 1 | – | X – 00H | × | × | × |
| | | B | 1 | 1 | – | B – 00H | × | × | × |
| | | C | 1 | 1 | – | C – 00H | × | × | × |
| | | !addr16 | 3 | 1 | 4 | (addr16) – 00H | × | × | × |
| | | ES:!addr16 | 4 | 2 | 5 | (ES:addr16) – 00H | × | × | × |
| | | saddr | 2 | 1 | – | (saddr) – 00H | × | × | × |
| | CMPS | X, [HL+byte] | 3 | 1 | 4 | X – (HL+byte) | × | × | × |
| | | X, ES:[HL+byte] | 4 | 2 | 5 | X – ((ES:HL)+byte) | × | × | × |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**3.** Except r = A

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (11/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z | AC | CY |
| 16-bit operation | ADDW | AX, #word | 3 | 1 | – | AX, CY ← AX+word | × | × | × |
| | | AX, AX | 1 | 1 | – | AX, CY ← AX+AX | × | × | × |
| | | AX, BC | 1 | 1 | – | AX, CY ← AX+BC | × | × | × |
| | | AX, DE | 1 | 1 | – | AX, CY ← AX+DE | × | × | × |
| | | AX, HL | 1 | 1 | – | AX, CY ← AX+HL | × | × | × |
| | | AX, !addr16 | 3 | 1 | 4 | AX, CY ← AX+(addr16) | × | × | × |
| | | AX, ES:!addr16 | 4 | 2 | 5 | AX, CY ← AX+(ES:addr16) | × | × | × |
| | | AX, saddrp | 2 | 1 | – | AX, CY ← AX+(saddrp) | × | × | × |
| | | AX, [HL+byte] | 3 | 1 | 4 | AX, CY ← AX+(HL+byte) | × | × | × |
| | | AX, ES: [HL+byte] | 4 | 2 | 5 | AX, CY ← AX+((ES:HL)+byte) | × | × | × |
| | SUBW | AX, #word | 3 | 1 | – | AX, CY ← AX – word | × | × | × |
| | | AX, BC | 1 | 1 | – | AX, CY ← AX – BC | × | × | × |
| | | AX, DE | 1 | 1 | – | AX, CY ← AX – DE | × | × | × |
| | | AX, HL | 1 | 1 | – | AX, CY ← AX – HL | × | × | × |
| | | AX, !addr16 | 3 | 1 | 4 | AX, CY ← AX – (addr16) | × | × | × |
| | | AX, ES:!addr16 | 4 | 2 | 5 | AX, CY ← AX – (ES:addr16) | × | × | × |
| | | AX, saddrp | 2 | 1 | – | AX, CY ← AX – (saddrp) | × | × | × |
| | | AX, [HL+byte] | 3 | 1 | 4 | AX, CY ← AX – (HL+byte) | × | × | × |
| | | AX, ES: [HL+byte] | 4 | 2 | 5 | AX, CY ← AX – ((ES:HL)+byte) | × | × | × |
| | CMPW | AX, #word | 3 | 1 | – | AX – word | × | × | × |
| | | AX, BC | 1 | 1 | – | AX – BC | × | × | × |
| | | AX, DE | 1 | 1 | – | AX – DE | × | × | × |
| | | AX, HL | 1 | 1 | – | AX – HL | × | × | × |
| | | AX, !addr16 | 3 | 1 | 4 | AX – (addr16) | × | × | × |
| | | AX, ES:!addr16 | 4 | 2 | 5 | AX – (ES:addr16) | × | × | × |
| | | AX, saddrp | 2 | 1 | – | AX – (saddrp) | × | × | × |
| | | AX, [HL+byte] | 3 | 1 | 4 | AX – (HL+byte) | × | × | × |
| | | AX, ES: [HL+byte] | 4 | 2 | 5 | AX – ((ES:HL)+byte) | × | × | × |
| Multiply | MULU | X | 1 | 1 | – | AX ← A×X | | | |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (12/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Operation | Flag |
|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z AC CY |
| Multiply, Divide, Multiply & accumu-late | MULU | X | 1 | 1 | – | AX ← A × X | |
| | MULHU | | 3 | 2 | – | BCAX ← AX × BC (unsigned) | |
| | MULH | | 3 | 2 | – | BCAX ← AX × BC (signed) | |
| | DIVHU | | 3 | 9 | – | AX (quotient), DE (remainder) ← AX ÷ DE (unsigned) | |
| | DIVWU | | 3 | 17 | – | BCAX (quotient), HLDE (remainder) ← BCAX ÷ HLDE (unsigned) | |
| | MACHU | | 3 | 3 | – | MACR ← MACR + AX × BC (unsigned) | × × |
| | MACH | | 3 | 3 | – | MACR ← MACR + AX × BC(signed) | × × |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

<R> **Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine. Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.**

- **V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.6 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

**Remarks 1.** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**2.** MACR indicates the multiplication and accumulation register (MACRH, MACRL).

**Table 33-5. Operation List (13/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z | AC | CY |
| Increment/ decrement | INC | r | 1 | 1 | – | r ← r+1 | × | × | |
| | | !addr16 | 3 | 2 | – | (addr16) ← (addr16)+1 | × | × | |
| | | ES:!addr16 | 4 | 3 | – | (ES, addr16) ← (ES, addr16)+1 | × | × | |
| | | saddr | 2 | 2 | – | (saddr) ← (saddr)+1 | × | × | |
| | | [HL+byte] | 3 | 2 | – | (HL+byte) ← (HL+byte)+1 | × | × | |
| | | ES: [HL+byte] | 4 | 3 | – | ((ES:HL)+byte) ← ((ES:HL)+byte)+1 | × | × | |
| | DEC | r | 1 | 1 | – | r ← r – 1 | × | × | |
| | | !addr16 | 3 | 2 | – | (addr16) ← (addr16) – 1 | × | × | |
| | | ES:!addr16 | 4 | 3 | – | (ES, addr16) ← (ES, addr16) – 1 | × | × | |
| | | saddr | 2 | 2 | – | (saddr) ← (saddr) – 1 | × | × | |
| | | [HL+byte] | 3 | 2 | – | (HL+byte) ← (HL+byte) – 1 | × | × | |
| | | ES: [HL+byte] | 4 | 3 | – | ((ES:HL)+byte) ← ((ES:HL)+byte) – 1 | × | × | |
| | INCW | rp | 1 | 1 | – | rp ← rp+1 | | | |
| | | !addr16 | 3 | 2 | – | (addr16) ← (addr16)+1 | | | |
| | | ES:!addr16 | 4 | 3 | – | (ES, addr16) ← (ES, addr16)+1 | | | |
| | | saddrp | 2 | 2 | – | (saddrp) ← (saddrp)+1 | | | |
| | | [HL+byte] | 3 | 2 | – | (HL+byte) ← (HL+byte)+1 | | | |
| | | ES: [HL+byte] | 4 | 3 | – | ((ES:HL)+byte) ← ((ES:HL)+byte)+1 | | | |
| | DECW | rp | 1 | 1 | – | rp ← rp – 1 | | | |
| | | !addr16 | 3 | 2 | – | (addr16) ← (addr16) – 1 | | | |
| | | ES:!addr16 | 4 | 3 | – | (ES, addr16) ← (ES, addr16) – 1 | | | |
| | | saddrp | 2 | 2 | – | (saddrp) ← (saddrp) – 1 | | | |
| | | [HL+byte] | 3 | 2 | – | (HL+byte) ← (HL+byte) – 1 | | | |
| | | ES: [HL+byte] | 4 | 3 | – | ((ES:HL)+byte) ← ((ES:HL)+byte) – 1 | | | |
| Shift | SHR | A, cnt | 2 | 1 | – | (CY ← $A_0$, $A_{m-1}$ ← $A_m$, $A_7$ ← 0) ×cnt | | | × |
| | SHRW | AX, cnt | 2 | 1 | – | (CY ← $AX_0$, $AX_{m-1}$ ← $AX_m$, $AX_{15}$ ← 0) ×cnt | | | × |
| | SHL | A, cnt | 2 | 1 | – | (CY ← $A_7$, $A_m$ ← $A_{m-1}$, $A_0$ ← 0) ×cnt | | | × |
| | | B, cnt | 2 | 1 | – | (CY ← $B_7$, $B_m$ ← $B_{m-1}$, $B_0$ ← 0) ×cnt | | | × |
| | | C, cnt | 2 | 1 | – | (CY ← $C_7$, $C_m$ ← $C_{m-1}$, $C_0$ ← 0) ×cnt | | | × |
| | SHLW | AX, cnt | 2 | 1 | – | (CY ← $AX_{15}$, $AX_m$ ← $AX_{m-1}$, $AX_0$ ← 0) ×cnt | | | × |
| | | BC, cnt | 2 | 1 | – | (CY ← $BC_{15}$, $BC_m$ ← $BC_{m-1}$, $BC_0$ ← 0) ×cnt | | | × |
| | SAR | A, cnt | 2 | 1 | – | (CY ← $A_0$, $A_{m-1}$ ← $A_m$, $A_7$ ← $A_7$) ×cnt | | | × |
| | SARW | AX, cnt | 2 | 1 | – | (CY ← $AX_0$, $AX_{m-1}$ ← $AX_m$, $AX_{15}$ ← $AX_{15}$) ×cnt | | | × |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**Remarks 1.** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**2.** cnt indicates the bit shift count.

**Table 33-5. Operation List (14/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z | AC | CY |
| Rotate | ROR | A, 1 | 2 | 1 | – | (CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$)×1 | | | × |
| | ROL | A, 1 | 2 | 1 | – | (CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$)×1 | | | × |
| | RORC | A, 1 | 2 | 1 | – | (CY $\leftarrow A_0$, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$)×1 | | | × |
| | ROLC | A, 1 | 2 | 1 | – | (CY $\leftarrow A_7$, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$)×1 | | | × |
| | ROLWC | AX,1 | 2 | 1 | – | (CY $\leftarrow AX_{15}$, $AX_0 \leftarrow$ CY, $AX_{m+1} \leftarrow AX_m$) ×1 | | | × |
| | | BC,1 | 2 | 1 | – | (CY $\leftarrow BC_{15}$, $BC_0 \leftarrow$ CY, $BC_{m+1} \leftarrow BC_m$) ×1 | | | × |
| Bit manipulate | MOV1 | CY, A.bit | 2 | 1 | – | CY ← A.bit | | | × |
| | | A.bit, CY | 2 | 1 | – | A.bit ← CY | | | |
| | | CY, PSW.bit | 3 | 1 | – | CY ← PSW.bit | | | × |
| | | PSW.bit, CY | 3 | 4 | – | PSW.bit ← CY | × | × | |
| | | CY, saddr.bit | 3 | 1 | – | CY ← (saddr).bit | | | × |
| | | saddr.bit, CY | 3 | 2 | – | (saddr).bit ← CY | | | |
| | | CY, sfr.bit | 3 | 1 | – | CY ← sfr.bit | | | × |
| | | sfr.bit, CY | 3 | 2 | – | sfr.bit ← CY | | | |
| | | CY,[HL].bit | 2 | 1 | 4 | CY ← (HL).bit | | | × |
| | | [HL].bit, CY | 2 | 2 | – | (HL).bit ← CY | | | |
| | | CY, ES:[HL].bit | 3 | 2 | 5 | CY ← (ES, HL).bit | | | × |
| | | ES:[HL].bit, CY | 3 | 3 | – | (ES, HL).bit ← CY | | | |
| | AND1 | CY, A.bit | 2 | 1 | – | CY ← CY ∧ A.bit | | | × |
| | | CY, PSW.bit | 3 | 1 | – | CY ← CY ∧ PSW.bit | | | × |
| | | CY, saddr.bit | 3 | 1 | – | CY ← CY ∧ (saddr).bit | | | × |
| | | CY, sfr.bit | 3 | 1 | – | CY ← CY ∧ sfr.bit | | | × |
| | | CY,[HL].bit | 2 | 1 | 4 | CY ← CY ∧ (HL).bit | | | × |
| | | CY, ES:[HL].bit | 3 | 2 | 5 | CY ← CY ∧ (ES, HL).bit | | | × |
| | OR1 | CY, A.bit | 2 | 1 | – | CY ← CY ∨ A.bit | | | × |
| | | CY, PSW.bit | 3 | 1 | – | CYX ← CY ∨ ∨PSW.bit | | | × |
| | | CY, saddr.bit | 3 | 1 | – | CY ← CY ∨ (saddr).bit | | | × |
| | | CY, sfr.bit | 3 | 1 | – | CY ← CY ∨ sfr.bit | | | × |
| | | CY, [HL].bit | 2 | 1 | 4 | CY ← CY ∨ (HL).bit | | | × |
| | | CY, ES:[HL].bit | 3 | 2 | 5 | CY ← CY ∨ (ES, HL).bit | | | × |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (15/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z | AC | CY |
| Bit manipulate | XOR1 | CY, A.bit | 2 | 1 | – | CY ← CY ⩛ A.bit | | | × |
| | | CY, PSW.bit | 3 | 1 | – | CY ← CY ⩛ PSW.bit | | | × |
| | | CY, saddr.bit | 3 | 1 | – | CY ← CY ⩛ (saddr).bit | | | × |
| | | CY, sfr.bit | 3 | 1 | – | CY ← CY ⩛ sfr.bit | | | × |
| | | CY, [HL].bit | 2 | 1 | 4 | CY ← CY ⩛ (HL).bit | | | × |
| | | CY, ES:[HL].bit | 3 | 2 | 5 | CY ← CY ⩛ (ES, HL).bit | | | × |
| | SET1 | A.bit | 2 | 1 | – | A.bit ← 1 | | | |
| | | PSW.bit | 3 | 4 | – | PSW.bit ← 1 | × | × | × |
| | | !addr16.bit | 4 | 2 | – | (addr16).bit ← 1 | | | |
| | | ES:!addr16.bit | 5 | 3 | – | (ES, addr16).bit ← 1 | | | |
| | | saddr.bit | 3 | 2 | – | (saddr).bit ← 1 | | | |
| | | sfr.bit | 3 | 2 | – | sfr.bit ← 1 | | | |
| | | [HL].bit | 2 | 2 | – | (HL).bit ← 1 | | | |
| | | ES:[HL].bit | 3 | 3 | – | (ES, HL).bit ← 1 | | | |
| | CLR1 | A.bit | 2 | 1 | – | A.bit ← 0 | | | |
| | | PSW.bit | 3 | 4 | – | PSW.bit ← 0 | × | × | × |
| | | !addr16.bit | 4 | 2 | – | (addr16).bit ← 0 | | | |
| | | ES:!addr16.bit | 5 | 3 | – | (ES, addr16).bit ← 0 | | | |
| | | saddr.bit | 3 | 2 | – | (saddr.bit) ← 0 | | | |
| | | sfr.bit | 3 | 2 | – | sfr.bit ← 0 | | | |
| | | [HL].bit | 2 | 2 | – | (HL).bit ← 0 | | | |
| | | ES:[HL].bit | 3 | 3 | – | (ES, HL).bit ← 0 | | | |
| | SET1 | CY | 2 | 1 | – | CY ← 1 | | | 1 |
| | CLR1 | CY | 2 | 1 | – | CY ← 0 | | | 0 |
| | NOT1 | CY | 2 | 1 | – | CY ← $\overline{CY}$ | | | × |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (16/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag |
|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z AC CY |
| Call/ return | CALL | rp | 2 | 3 | – | (SP – 2) ← (PC+2)S, (SP – 3) ← (PC+2)H, (SP – 4) ← (PC+2)L, PC ← CS, rp, SP ← SP – 4 | |
| | | $!addr20 | 3 | 3 | – | (SP – 2) ← (PC+3)S, (SP – 3) ← (PC+3)H, (SP – 4) ← (PC+3)L, PC ← PC+3+jdisp16, SP ← SP – 4 | |
| | | !addr16 | 3 | 3 | – | (SP – 2) ← (PC+3)S, (SP – 3) ← (PC+3)H, (SP – 4) ← (PC+3)L, PC ← 0000, addr16, SP ← SP – 4 | |
| | | !!addr20 | 4 | 3 | – | (SP – 2) ← (PC+4)S, (SP – 3) ← (PC+4)H, (SP – 4) ← (PC+4)L, PC ← addr20, SP ← SP – 4 | |
| | CALLT | [addr5] | 2 | 5 | – | (SP – 2) ← (PC+2)S , (SP – 3) ← (PC+2)H, (SP – 4) ← (PC+2)L , PCS ← 0000, PCH ← (0000, addr5+1), PCL ← (0000, addr5), SP ← SP – 4 | |
| | BRK | - | 2 | 5 | – | (SP – 1) ← PSW, (SP – 2) ← (PC+2)S, (SP – 3) ← (PC+2)H, (SP – 4) ← (PC+2)L, PCS ← 0000, PCH ← (0007FH), PCL ← (0007EH), SP ← SP – 4, IE ← 0 | |
| | RET | - | 1 | 6 | – | PCL ← (SP), PCH ← (SP+1), PCS ← (SP+2), SP ← SP+4 | |
| | RETI | - | 2 | 6 | – | PCL ← (SP), PCH ← (SP+1), PCS ← (SP+2), PSW← (SP+3), SP ← SP+4 | R R R |
| | RETB | - | 2 | 6 | – | PCL ← (SP), PCH ← (SP+1), PCS ← (SP+2), PSW ← (SP+3), SP ← SP+4 | R R R |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (17/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z | AC | CY |
| Stack manipulate | PUSH | PSW | 2 | 1 | – | (SP – 1) ← PSW, (SP – 2) ← 00H, SP ← SP–2 | | | |
| | | rp | 1 | 1 | – | (SP – 1) ← $rp_H$, (SP – 2) ← $rp_L$, SP ← SP – 2 | | | |
| | POP | PSW | 2 | 3 | – | PSW ← (SP+1), SP ← SP + 2 | R | R | R |
| | | rp | 1 | 1 | – | $rp_L$ ←(SP), $rp_H$ ← (SP+1), SP ← SP + 2 | | | |
| | MOVW | SP, #word | 4 | 1 | – | SP ← word | | | |
| | | SP, AX | 2 | 1 | – | SP ← AX | | | |
| | | AX, SP | 2 | 1 | – | AX ← SP | | | |
| | | HL, SP | 3 | 1 | – | HL ← SP | | | |
| | | BC, SP | 3 | 1 | – | BC ← SP | | | |
| | | DE, SP | 3 | 1 | – | DE ← SP | | | |
| | ADDW | SP, #byte | 2 | 1 | – | SP ← SP + byte | | | |
| | SUBW | SP, #byte | 2 | 1 | – | SP ← SP – byte | | | |
| Unconditional branch | BR | AX | 2 | 3 | – | PC ← CS, AX | | | |
| | | $addr20 | 2 | 3 | – | PC ← PC + 2 + jdisp8 | | | |
| | | $!addr20 | 3 | 3 | – | PC ← PC + 3 + jdisp16 | | | |
| | | !addr16 | 3 | 3 | – | PC ← 0000, addr16 | | | |
| | | !!addr20 | 4 | 3 | – | PC ← addr20 | | | |
| Conditional branch | BC | $addr20 | 2 | 2/4 Note3 | – | PC ← PC + 2 + jdisp8 if CY = 1 | | | |
| | BNC | $addr20 | 2 | 2/4 Note3 | – | PC ← PC + 2 + jdisp8 if CY = 0 | | | |
| | BZ | $addr20 | 2 | 2/4 Note3 | – | PC ← PC + 2 + jdisp8 if Z = 1 | | | |
| | BNZ | $addr20 | 2 | 2/4 Note3 | – | PC ← PC + 2 + jdisp8 if Z = 0 | | | |
| | BH | $addr20 | 3 | 2/4 Note3 | – | PC ← PC + 3 + jdisp8 if (Z∨CY)=0 | | | |
| | BNH | $addr20 | 3 | 2/4 Note3 | – | PC ← PC + 3 + jdisp8 if (Z∨CY)=1 | | | |
| | BT | saddr.bit, $addr20 | 4 | 3/5 Note3 | – | PC ← PC + 4 + jdisp8 if (saddr).bit = 1 | | | |
| | | sfr.bit, $addr20 | 4 | 3/5 Note3 | – | PC ← PC + 4 + jdisp8 if sfr.bit = 1 | | | |
| | | A.bit, $addr20 | 3 | 3/5 Note3 | – | PC ← PC + 3 + jdisp8 if A.bit = 1 | | | |
| | | PSW.bit, $addr20 | 4 | 3/5 Note3 | – | PC ← PC + 4 + jdisp8 if PSW.bit = 1 | | | |
| | | [HL].bit, $addr20 | 3 | 3/5 Note3 | 6/7 | PC ← PC + 3 + jdisp8 if (HL).bit = 1 | | | |
| | | ES:[HL].bit, $addr20 | 4 | 4/6 Note3 | 7/8 | PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 | | | |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**3.** This indicates the number of clocks “when condition is not met/when condition is met”.

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

**Table 33-5. Operation List (18/18)**

| Instruction Group | Mnemonic | Operands | Bytes | Clocks | | Clocks | Flag |
|---|---|---|---|---|---|---|---|
| | | | | Note 1 | Note 2 | | Z AC CY |
| Conditional branch | BF | saddr.bit, $addr20 | 4 | 3/5 Note3 | – | PC ← PC + 4 + jdisp8 if (saddr).bit = 0 | |
| | | sfr.bit, $addr20 | 4 | 3/5 Note3 | – | PC ← PC + 4 + jdisp8 if sfr.bit = 0 | |
| | | A.bit, $addr20 | 3 | 3/5 Note3 | – | PC ← PC + 3 + jdisp8 if A.bit = 0 | |
| | | PSW.bit, $addr20 | 4 | 3/5 Note3 | – | PC ← PC + 4 + jdisp8 if PSW.bit = 0 | |
| | | [HL].bit, $addr20 | 3 | 3/5 Note3 | 6/7 | PC ← PC + 3 + jdisp8 if (HL).bit = 0 | |
| | | ES:[HL].bit, $addr20 | 4 | 4/6 Note3 | 7/8 | PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0 | |
| | BTCLR | saddr.bit, $addr20 | 4 | 3/5 Note3 | – | PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit | |
| | | sfr.bit, $addr20 | 4 | 3/5 Note3 | – | PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit | |
| | | A.bit, $addr20 | 3 | 3/5 Note3 | – | PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit | |
| | | PSW.bit, $addr20 | 4 | 3/5 Note3 | – | PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit | × × × |
| | | [HL].bit, $addr20 | 3 | 3/5 Note3 | – | PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit | |
| | | ES:[HL].bit, $addr20 | 4 | 4/6 Note3 | – | PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit | |
| Conditional skip | SKC | – | 2 | 1 | – | Next instruction skip if CY = 1 | |
| | SKNC | – | 2 | 1 | – | Next instruction skip if CY = 0 | |
| | SKZ | – | 2 | 1 | – | Next instruction skip if Z = 1 | |
| | SKNZ | – | 2 | 1 | – | Next instruction skip if Z = 0 | |
| | SKH | – | 2 | 1 | – | Next instruction skip if (Z∨CY)=0 | |
| | SKNH | – | 2 | 1 | – | Next instruction skip if (Z∨CY)=1 | |
| CPU control | SEL Note4 | RBn | 2 | 1 | – | RBS[1:0] ← n | |
| | NOP | – | 1 | 1 | – | No Operation | |
| | EI | – | 3 | 4 | – | IE ← 1 (Enable Interrupt) | |
| | DI | – | 3 | 4 | – | IE ← 0 (Disable Interrupt) | |
| | HALT | – | 2 | 3 | – | Set HALT Mode | |
| | STOP | – | 2 | 3 | – | Set STOP Mode | |

**Notes 1.** Number of CPU clocks ($f_{CLK}$) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

**2.** Number of CPU clocks ($f_{CLK}$) when the program memory area is accessed.

**3.** This indicates the number of clocks “when condition is not met/when condition is met”.

**4.** n indicates the number of register banks (n = 0 to 3)

**Remark** Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

# CHAPTER 34 ELECTRICAL SPECIFICATIONS (GRADE L)

**Cautions 1. RL78/F13 and RL78/F14 have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.**

**2. With products not provided with an $EV_{DD0}$, $EV_{DD1}$, $EV_{SS0}$, or $EV_{SS1}$ pin, replace $EV_{DD0}$ and $EV_{DD1}$ with $V_{DD}$, or replace $EV_{SS0}$ and $EV_{SS1}$ with $V_{SS}$.**

**3. The pins mounted depending on the product. For details, refer to 1.5 Pin Configurations and 2.1 Pin Function List.**

**4. The products are classified into the following five groups according to the product type, pin count, and code flash memory size. In this chapter, the products are referred to by group names depending on the content. In this case, refer to the following classification.**

**Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory**

**Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory**

**Group C: RL78/F13 (CAN and LIN incorporated) products with 30, 32, 48, 64, or 80 pins and 32 Kbytes to 128 Kbytes of code flash memory**

**Group D: RL78/F14 products with 30, 32, 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory**

**Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory**

## 34.1 Absolute Maximum Ratings

(1/2)

| Parameter | Symbol | Conditions | Ratings | Unit |
|---|---|---|---|---|
| Supply voltage | $V_{DD}$ | | -0.5 to +6.5 | V |
| | $EV_{DD0}$, $EV_{DD1}$ | $EV_{DD0} = EV_{DD1}$ | -0.5 to +6.5 | V |
| | $V_{SS}$ | | -0.5 to +0.3 | V |
| | $EV_{SS0}$, $EV_{SS1}$ | $EV_{SS0} = EV_{SS1}$ | -0.5 to +0.3 | V |
| REGC pin input voltage | $V_{IREGC}$ | REGC | -0.3 to +2.8 and -0.3 to $V_{DD}$+0.3 Note 1 | V |
| Input voltage | $V_{I1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97 Note 4, P106, P107, P120, P125 to P127, P140, P150 to P157 | -0.3 to $EV_{DD0}$+0.3 and -0.3 to $V_{DD}$+0.3 Note 2 | V |
| | $V_{I2}$ | P33, P34, P80 to P87, P90 to P97 Note 4, P100 to P105, P121 to P124, P137, $\overline{\text{RESET}}$ | -0.3 to $V_{DD}$+0.3 Note 2 | V |
| Output voltage | $V_{O1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97 Note 4, P106, P107, P120, P125 to P127, P130, P140, P150 to P157 | -0.3 to $EV_{DD0}$+0.3 and -0.3 to $V_{DD}$+0.3 Note 2 | V |
| | $V_{O2}$ | P33, P34, P80 to P87, P90 to P97 Note 4, P100 to P105 | -0.3 to $V_{DD}$+0.3 | V |
| Analog input voltage | $V_{AI1}$ | ANI24 to ANI30 | -0.3 to $EV_{DD0}$+0.3 and -0.3 to $AV_{REF(+)}$+0.3 Notes 2, 3 | V |
| | $V_{AI2}$ | ANI0 to ANI23 | -0.3 to $V_{DD}$+0.3 and -0.3 to $AV_{REF(+)}$+0.3 Notes 2, 3 | V |

**Notes 1.** Connect the REGC pin to $V_{SS}$ via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

**2.** Must be 6.5 V or lower.

**3.** For pins to be used in A/D conversion, the voltage should not exceed the value $AV_{REF(+)}$ + 0.3.

**4.** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(2/2)

| Parameter | Symbol | Conditions | | Ratings | Unit |
|---|---|---|---|---|---|
| Output current, high | $I_{OH1}$ | Per pin | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P130, P140, P150 to P157 | -40 | mA |
| | | Total of all pins -170 mA | P01, P02, P40 to P47, P92 to P97[Note], P120, P125 to P127, P150 to P153 | -70 | mA |
| | | | P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P106, P107, P130, P140, P154 to P157 | -100 | mA |
| | $I_{OH2}$ | Per pin | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105 | -0.5 | mA |
| | | Total of all pins | | -2 | mA |
| Output current, low | $I_{OL1}$ | Per pin | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P130, P140, P150 to P157 | 40 | mA |
| | | Total of all pins 170 mA | P01, P02, P40 to P47, P92 to P97[Note], P120, P125 to P127, P150 to P153 | 70 | mA |
| | | | P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P106, P107, P130, P140, P154 to P157 | 100 | mA |
| | $I_{OL2}$ | Per pin | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105 | 1 | mA |
| | | Total of all pins | | 5 | mA |
| Operating ambient temperature | $T_A$ | In normal operation mode | | -40 to +105 | °C |
| | | In flash memory programming mode | | | |
| Storage temperature | $T_{stg}$ | | | -65 to +150 | °C |

**Note** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

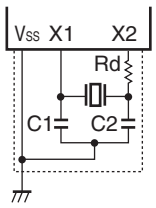
**Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

## 34.2 Oscillator Characteristics

### 34.2.1 Main System Clock Oscillator Characteristics

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Resonator | Recommended Circuit | Parameter | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Ceramic resonator/ Crystal resonator |  | X1 clock oscillation frequency (fx) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | 1.0 | | 20.0 | MHz |

**Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.**

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as $V_{SS}$.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

**2. Customers are requested to consult the resonator manufacturer to select an appropriate resonator and to determine the proper oscillation constant. Customers are also requested to adequately evaluate the oscillation on their system. Determine the X1 clock oscillation stabilization time using the oscillation stabilization time of the oscillation stabilization time counter status register (OSTC) and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.**

### 34.2.2 On-chip Oscillator Characteristics

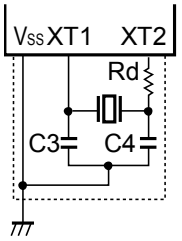
**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Oscillators | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| High-speed on-chip oscillator clock frequency[Note] | $f_{IH}$ | | 1 | | 64 | MHz |
| High-speed on-chip oscillator clock frequency accuracy | - | | -2 | | +2 | % |
| Low-speed on-chip oscillator clock frequency | $f_{IL}$, $f_{WDT}$ | | | 15 | | kHz |
| Low-speed on-chip oscillator clock frequency accuracy | - | | -15 | | +15 | % |

**Note** High-speed on-chip oscillator frequency is selected with bits 0 to 4 of the option byte (000C2H/020C2H) and bits 0 to 2 of the HOCODIV register.

### 34.2.3 Subsystem Clock Oscillator Characteristics

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Resonator | Recommended Circuit | Item | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Crystal resonator |  | XT1 clock oscillation frequency ($f_{XT}$) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | 29.0 | 32.768 | 35.0 | kHz |

**Cautions 1. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.**

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as $V_{SS}$.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

**2. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption and thus required to be adequately evaluated on the system. Customers are requested to consult the resonator manufacturer to select an appropriate resonator and to determine the proper oscillation constant.**

### 34.2.4 PLL Circuit Characteristics

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Resonator | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| PLL input enable clock frequency[Note 1] | $f_{PLLI}$ | PLLMUL = 0 | PLLDIV0 = 0 | 3.92 | 4.0 | 4.08 | MHz |
| | | | PLLDIV0 = 1 | 7.84 | 8.0 | 8.16 | MHz |
| | | PLLMUL = 1 | PLLDIV0 = 0 | 3.92 | 4.0 | 4.08 | MHz |
| | | | PLLDIV0 = 1 | 7.84 | 8.0 | 8.16 | MHz |
| PLL output frequency (center value) | $f_{PLL}$ | PLLMUL = 0 | PLLDIV0 = 0 | $f_{PLLI}$ × 12/2 | | | MHz |
| | | | PLLDIV0 = 1 | $f_{PLLI}$ × 12/4 | | | MHz |
| | | PLLMUL = 1 | PLLDIV0 = 0 | $f_{PLLI}$ × 16/2 | | | MHz |
| | | | PLLDIV0 = 1 | $f_{PLLI}$ × 16/4 | | | MHz |
| Long-term jitter[Notes 2, 3] | $t_{LJ}$ | $f_{PLL}$ = 24 MHz (480 counts) | | -2 | | +2 | ns |
| | | $f_{PLL}$ = 32 MHz (640 counts) | | -2 | | +2 | ns |
| | | $f_{PLL}$ = 48 MHz (960 counts) | | -2 | | +2 | ns |
| | | $f_{PLL}$ = 64 MHz (1280 counts) | | -2 | | +2 | ns |

**Notes 1.** If the high-speed on-chip oscillator clock is to be selected as the PLL input clock, the minimum and maximum values will reflect the range of accuracy of the oscillation frequency by the high-speed on-chip oscillator clock.

**2.** Guaranteed by design, but not tested before shipment.

**3.** Indicates 20 μs.

## 34.3 DC Characteristics

### 34.3.1 Pin Characteristics

For the relationship between the port pins shown in the following tables and the products, refer to **CHAPTER 4 PORT FUNCTIONS**.

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (1/4)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Output current, high[Note 1] | $I_{OH1}$ | Per pin for P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note 3], P106, P107, P120, P125 to P127, P130, P140, P150 to 157 | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -5.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -3.0 | mA |
| | | Per pin for P10, P12, P14, P30,P120,P140 (special slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -0.6 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -0.2 | mA |
| | | Total of P01, P02, P40 to P47, P92 to P97[Note 3], P120, P125 to P127, P150 to P153 (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -20.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -10.0 | mA |
| | | Total of P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P106, P107, P130, P140, P154 to P157 (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -30.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -19.0 | mA |
| | | Total of all pins (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -50.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -29.0 | mA |
| | $I_{OH2}$ | Per pin for P33, P34, P80 to P87, P90 to P97[Note 3], P100 to P105 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | -0.1 | mA |
| | | Total of all pins (for duty factors ≤ 70%[Note 2]) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | -2.0 | mA |

<R> **Notes 1.** Value of current at which the device operation is guaranteed even if the current flows from pins $EV_{DD0}$, $EV_{DD1}$ and $V_{DD}$ to an output pin.

**2.** These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins ($I_{OH}$ × 0.7)/(n × 0.01)

<Example> Where n = 80% and $I_{OH}$ = -10.0 mA

Total output current of pins = (-10.0 × 0.7)/(80 × 0.01) ≈ -8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

**3.** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Caution P10 to P17, P60 to P63, P70 to P72, and P120 do not output high level in N-ch open-drain mode. P10 to P12 and P70 to P72 of the Group A products do not support N-ch open-drain mode.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (2/4)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Output current, low[Note 1] | $I_{OL1}$ | Per pin for P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note 3], P106, P107, P120, P125 to P127, P130, P140, P150 to 157 | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 8.5 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 4.0 | mA |
| | | Per pin for P10, P12, P14, P30, P120, P140 (special slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 0.59 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 0.07 | mA |
| | | Total of P01, P02, P40 to P47, P92 to P97[Note 3], P120, P125 to P127, P150 to P153 (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 20.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 15.0 | mA |
| | | Total of P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P106, P107, P130, P140, P154 to P157 (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 45.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 35.0 | mA |
| | | Total of all pins (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 65.0 | mA |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | | 50.0 | mA |
| | $I_{OL2}$ | Per pin for P33, P34, P80 to P87, P90 to P97[Note 3], P100 to P105 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | 0.4 | mA |
| | | Total of all pins (for duty factors ≤ 70%[Note 2]) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | 5.0 | mA |

<R> **Notes 1.** Value of current at which the device operation is guaranteed even if the current flows to the $EV_{SS0}$, $EV_{SS1}$ and $V_{SS}$ pins from an output pin.

**2.** These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins ($I_{OL}$ × 0.7)/(n × 0.01)

<Example> Where n = 80% and $I_{OL}$ = 10.0 mA

Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≈ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

**3.** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (3/4)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Input voltage, high | $V_{IH1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P106, P107, P120, P125 to P127, P140, P150 to P157 (Schmitt 1 mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0.65 $EV_{DD0}$ | | $EV_{DD0}$ Note 1 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0.7 $EV_{DD0}$ | | $EV_{DD0}$ Note 1 | V |
| | $V_{IH2}$ | P10, P11, P13, P14, P16, P17, P30, P43, P50, P52 to P54, P60 to P63, P70, P71, P73, P75 to P77, P107, P125, P150, P152, P153 (Schmitt 3 mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0.8 $EV_{DD0}$ | | $EV_{DD0}$ Note 1 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0.85 $EV_{DD0}$ | | $EV_{DD0}$ Note 1 | V |
| | $V_{IH3}$ | P10, P11, P13, P14, P16, P17, P30, P54, P62, P63, P70, P71, P73, P125 (TTL mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 2.2 | | $EV_{DD0}$ Note 1 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 2.0 | | $EV_{DD0}$ Note 1 | V |
| | $V_{IH4}$ Note 2 | P33, P34, P80 to P87, P90 to P97, P100 to P105, P137 (fixed to Schmitt 3 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0.8 $V_{DD}$ | | $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0.85 $V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH5}$ | $\overline{\text{RESET}}$ (fixed to Schmitt 1 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0.65 $V_{DD}$ | | $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0.7 $V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH6}$ | P121 to P124, EXCLK, EXCLKS (fixed to Schmitt 2 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0.8 $V_{DD}$ | | $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0.8 $V_{DD}$ | | $V_{DD}$ | V |

**Notes 1.** The maximum value of $V_{IH}$ of the pins P10 to P17, P60 to P63, P70 to P72, and P120 is $EV_{DD0}$, even in N-ch open-drain mode.

**2.** P92 to P96 of the Group A products are fixed to Schmitt 1 mode.
P96 and P97 of the Group B, C, and D products are fixed to Schmitt 1 mode.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (4/4)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Input voltage, low | $V_{IL1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P106, P107, P120, P125 to P127, P140, P150 to P157 (Schmitt 1 mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0 | | 0.35 $EV_{DD0}$ | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0 | | 0.3 $EV_{DD0}$ | V |
| | $V_{IL2}$ | P10, P11, P13, P14, P16, P17, P30, P43, P50, P52 to P54, P60 to P63, P70, P71, P73, P75 to P77, P107, P125, P150, P152, P153 (Schmitt 3 mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0 | | 0.5 $EV_{DD0}$ | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0 | | 0.4 $EV_{DD0}$ | V |
| | $V_{IL3}$ | P10, P11, P13, P14, P16, P17, P30, P54, P62, P63, P70, P71, P73, P125 (TTL mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0 | | 0.8 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0 | | 0.5 | V |
| | $V_{IL4}$ Note | P33, P34, P80 to P87, P90 to P97, P100 to P105, P137 (fixed to Schmitt 3 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0 | | 0.5 $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0 | | 0.4 $V_{DD}$ | V |
| | $V_{IL5}$ | $\overline{\text{RESET}}$ (fixed to Schmitt 1 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0 | | 0.35 $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0 | | 0.3 $V_{DD}$ | V |
| | $V_{IL6}$ | P121 to P124, EXCLK, EXCLKS (fixed to Schmitt 2 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0 | | 0.2 $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0 | | 0.2 $V_{DD}$ | V |

**Note** P92 to P96 of the Group A products are fixed to Schmitt 1 mode.
P96 and P97 of the Group B, C, and D products are fixed to Schmitt 1 mode.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (1/2)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Output voltage, high | $V_{OH1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P130, P140, P150 to P157 (normal slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH1}$ = -5.0 mA | $EV_{DD0}$-0.9 | | | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH1}$ = -3.0 mA | $EV_{DD0}$-0.7 | | | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH1}$ = -1.0 mA | $EV_{DD0}$-0.5 | | | V |
| | $V_{OH2}$ | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OH2}$ = -100 μA | $V_{DD}$-0.5 | | | V |
| | $V_{OH3}$ | P10, P12, P14, P30, P120, P140 (special slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH3}$ = -0.6 mA | $EV_{DD0}$-0.8 | | | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH3}$ = -0.2 mA | $EV_{DD0}$-0.5 | | | V |
| Output voltage, low | $V_{OL1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P130, P140, P150 to P157 (normal slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL1}$ = 8.5 mA | | | 0.7 | V |
| | | | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL1}$ = 4.0 mA | | | 0.4 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL1}$ = 4.0 mA | | | 0.7 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL1}$ = 1.5 mA | | | 0.4 | V |
| | $V_{OL2}$ | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OL2}$ = 400 μA | | | 0.4 | V |
| | $V_{OL3}$ | P10, P12, P14, P30, P120, P140 (special slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL3}$ = 0.6 mA | | | 0.8 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL3}$ = 0.07 mA | | | 0.5 | V |

**Note** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Caution P10 to P17, P60 to P63, P70 to P72, and P120 do not output high level in N-ch open-drain mode.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (2/2)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Input leakage current, high | $I_{LIH1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P140, P150 to P157 | $V_I = EV_{DD0}$ | | | 1 | μA |
| | $I_{LIH2}$ | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105, P137, $\overline{RESET}$ | $V_I = V_{DD}$ | | | 1 | μA |
| | $I_{LIH3}$ | P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS) | $V_I = V_{DD}$ In input port or external clock input | | | 1 | μA |
| | | | In resonator connection | | | 10 | μA |
| Input leakage current, low | $I_{LIL1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P140, P150 to P157 | $V_I = EV_{SS0}$ | | | -1 | μA |
| | $I_{LIL2}$ | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105, P137, $\overline{RESET}$ | $V_I = V_{SS}$ | | | -1 | μA |
| | $I_{LIL3}$ | P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS) | $V_I = V_{SS}$ In input port or external clock input | | | -1 | μA |
| | | | In resonator connection | | | -10 | μA |
| On-chip pull-up resistance | $R_U$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97, P100 to P107, P120, P125 to P127, P140, P150 to P157 | $V_I = EV_{SS0}$, in input port | 10 | 20 | 100 | kΩ |

**Note** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Caution P10 to P17, P60 to P63, P70 to P72, and P120 do not output high level in N-ch open-drain mode.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

### 34.3.2 Supply Current Characteristics

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (1/3)**

| Items | Symbol | Conditions | | | | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| Supply current Note 1 | $I_{DD1}$ | Operating mode | Normal operation Note 2 | High-speed on-chip oscillator clock operation | $f_{IH}$ = 64 MHz | $f_{CLK}$ = 32 MHz Notes 3, 4 | | 6.5 | 14.0 | mA |
| | | | | | $f_{IH}$ = 32 MHz | $f_{CLK}$ = $f_{IH}$ Notes 3, 4 | | 6.1 | 13.0 | mA |
| | | | | | $f_{IH}$ = 1 MHz | $f_{CLK}$ = $f_{IH}$ Notes 3, 4 | | 1.0 | 2.5 | mA |
| | | | | Resonator operation | $f_{MX}$ = 20 MHz | $f_{CLK}$ = $f_{MX}$ Notes 3, 5 | | 4.2 | 9.0 | mA |
| | | | | | $f_{MX}$ = 1 MHz | $f_{CLK}$ = $f_{MX}$ Notes 3, 5 | | 0.9 | 2.5 | mA |
| | | | | Resonator operation (PLL operation) (PLL input clock = $f_{MX}$) | $f_{PLL}$ = 64 MHz, $f_{MX}$ = 8 MHz | $f_{CLK}$ = 32 MHz Notes 3, 6 | | 6.4 | 14.0 | mA |
| | | | | | $f_{PLL}$ = 32 MHz, $f_{MX}$ = 8 MHz | $f_{CLK}$ = 32 MHz Notes 3, 6 | | 6.3 | 13.5 | mA |
| | | | | | $f_{PLL}$ = 32 MHz, $f_{MX}$ = 4 MHz | $f_{CLK}$ = 32 MHz Notes 3, 6 | | 6.1 | 13.0 | mA |
| | | | | Subsystem clock operation | $f_{SUB}$ = 32.768 kHz | $f_{CLK}$ = $f_{SUB}$ Note 7 | | | | |
| | | | | | | Groups A to D | | 6.0 | 50.0 | μA |
| | | | | | | Group E | | 6.0 | 70.0 | μA |
| | | | | Low-speed on-chip oscillator clock operation | $f_{IL}$ = 15 kHz | $f_{CLK}$ = $f_{IL}$ Note 8 | | | | |
| | | | | | | Groups A to D | | 3.0 | 40.0 | μA |
| | | | | | | Group E | | 3.0 | 60.0 | μA |

**Notes 1.** Total current flowing into $V_{DD}$ and $EV_{DD0}$, including the input leakage current flowing when the level of the input pin is fixed to $V_{DD}$, $EV_{DD0}$, $V_{SS}$, or $EV_{SS0}$. However, not including the current flowing into the I/O buffer and on-chip pull-up/pull-down resistors.

**2.** Current drawn when all the CPU instructions are executed.

**3.** The values below the MAX. column include the peripheral operation current (except for background operation (BGO)). However, the LVD circuit, A/D converter, D/A converter, and comparator are stopped.

**4.** When high-speed system clock, subsystem clock, PLL clock, and low-speed on-chip oscillator clock are stopped.

**5.** When subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

**6.** When subsystem clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

**7.** When high-speed system clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator are stopped.

**8.** When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped.

**Remarks 1.** $f_{MX}$: High-speed system clock frequency

**2.** $f_{SUB}$: Subsystem clock frequency

**3.** $f_{PLL}$: PLL clock frequency

**4.** $f_{IH}$: High-speed on-chip oscillator clock frequency

**5.** $f_{IL}$: Low-speed on-chip oscillator clock frequency

**6.** $f_{CLK}$: CPU/peripheral hardware clock frequency

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (2/3)**

| Items | Symbol | Conditions | | | | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| Supply current Notes 1, 3 | $I_{DD2}$ | HALT mode Note 2 | Normal operation Note 4 | High-speed on-chip oscillator clock operation | $f_{IH}$ = 64 MHz | $f_{CLK}$ = 32 MHz Note 6 | | 1.2 | 10.0 | mA |
| | | | | | $f_{IH}$ = 32 MHz | $f_{CLK}$ = $f_{IH}$ Note 6 | | 1.0 | 9.0 | mA |
| | | | | | $f_{IH}$ = 1 MHz | $f_{CLK}$ = $f_{IH}$ Note 6 | | 0.3 | 1.5 | mA |
| | | | | Resonator operation | $f_{MX}$ = 20 MHz | $f_{CLK}$ = $f_{MX}$ Note 7 | | 0.6 | 6.0 | mA |
| | | | | | $f_{MX}$ = 1 MHz | $f_{CLK}$ = $f_{MX}$ Note 7 | | 0.2 | 1.5 | mA |
| | | | | Resonator operation (PLL operation) (PLL input clock = $f_{MX}$) | $f_{PLL}$ = 64 MHz, $f_{MX}$ = 8 MHz | $f_{CLK}$ = 32 MHz Note 8 | | 1.1 | 10.0 | mA |
| | | | | | $f_{PLL}$ = 32 MHz, $f_{MX}$ = 8 MHz | $f_{CLK}$ = 32 MHz Note 8 | | 1.0 | 9.5 | mA |
| | | | | | $f_{PLL}$ = 32 MHz, $f_{MX}$ = 4 MHz | $f_{CLK}$ = 32 MHz Note 8 | | 0.8 | 9.0 | mA |
| | | | | Subsystem clock operation | $f_{SUB}$ = 32.768 kHz | $f_{CLK}$ = $f_{SUB}$ Note 9 | | | | |
| | | | | | Groups A to D | | | 0.7 | 45.0 | μA |
| | | | | | Group E | | | 0.7 | 65.0 | |
| | | | | Low-speed on-chip oscillator clock operation | $f_{IL}$ = 15 kHz | $f_{CLK}$ = $f_{IL}$ Note 10 | | | | |
| | | | | | Groups A to D | | | 0.7 | 35.0 | μA |
| | | | | | Group E | | | 0.7 | 55.0 | |
| | $I_{DD3}$ | STOP mode Note 5 | | $T_A$ = +25°C | Groups A to D | | | 0.5 | | μA |
| | | | | | Group E | | | 0.5 | | |
| | | | | $T_A$ = +50°C | Groups A to D | | | | 2.5 | |
| | | | | | Group E | | | | 4.5 | |
| | | | | $T_A$ = +70°C | Groups A to D | | | | 4.5 | |
| | | | | | Group E | | | | 8.0 | |
| | | | | $T_A$ = +105°C | Groups A to D | | | | 30.0 | |
| | | | | | Group E | | | | 50.0 | |

**Notes 1.** Total current flowing into $V_{DD}$ and $EV_{DD0}$, including the input leakage current flowing when the level of the input pin is fixed to $V_{DD}$, $EV_{DD0}$, $V_{SS}$, or $EV_{SS0}$. However, not including the current flowing into the I/O buffer and on-chip pull-up/pull-down resistors.

**2.** When HALT mode is entered during fetch from the flash memory.

**3.** The values below the MAX. column include the peripheral operation current and STOP leakage current. However, the watchdog timer, LVD circuit, A/D converter, D/A converter, and comparator are stopped

**4.** Current flowing when all the instructions are executed by the CPU.

**5.** When high-speed system clock, subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

**6.** When high-speed system clock, subsystem clock, PLL clock, and low-speed on-chip oscillator clock are stopped.

**7.** When subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

**8.** When subsystem clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

**9.** When high-speed system clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

**10.** When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped.

**Remarks 1.** $f_{MX}$: High-speed system clock frequency

**2.** $f_{SUB}$: Subsystem clock frequency

**3.** $f_{PLL}$: PLL clock frequency

**4.** $f_{IH}$: High-speed on-chip oscillator clock frequency

**5.** $f_{IL}$: Low-speed on-chip oscillator clock frequency

**6.** $f_{CLK}$: CPU/peripheral hardware clock frequency

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (3/3)**

| Items | Symbol | Conditions | | | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|---|---|
| Supply current[Notes 1, 2] | $I_{SNOZ}$ | SNOOZE mode | A/D converter operation | During mode transition | | | 1.0 | 1.2 | mA |
| | | | | During conversion | Low-voltage mode $AV_{REFP}$ = $V_{DD}$ = 5.0 V | | 2.1 | 2.5 | mA |
| | | | DTC operation | | | | 4.5 | | mA |

**Notes 1.** Total current flowing into $V_{DD}$ and $EV_{DD0}$, including the input leakage current flowing when the level of the input pin is fixed to $V_{DD}$, $EV_{DD0}$, $V_{SS}$, or $EV_{SS0}$. However, not including the current flowing into the I/O buffer and on-chip pull-up/pull-down resistors.

**2.** The values below the MAX. column include the STOP leakage current.

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Window watchdog timer operating current | $I_{WDT}$ Notes 1, 2 | $f_{IL}$ = 15 kHz | | | 0.22 | | μA |
| A/D converter operating current | $I_{ADC}$ Note 3 | When conversion at maximum speed | Normal mode, $AV_{REFP}$ = $V_{DD}$ = 5.0 V | | 1.3 | 1.7 | mA |
| | | When internal reference voltage is selected Note 5 | | | 75.0 | | μA |
| LVD operating current | $I_{LVD}$ Note 4 | | | | 0.08 | | μA |
| Temperature sensor operating current | $I_{TMPS}$ | | | | 75.0 | | μA |
| D/A converter operating current | $I_{DAC}$ | Per channel | | | 0.8 | 1.5 | mA |
| Comparator operating current | $I_{CMP}$ | | | | 50.0 | | μA |
| BGO operating current | $I_{BGO}$ Note 6 | | | | 2.50 | 12.20 | mA |

**Notes 1.** When the high-speed on-chip oscillator clock and high-speed system clock are stopped.

**2.** Current flowing only to the watchdog timer (including the operation current of the 1.5 kHz on-chip oscillator). The current value is the sum of $I_{DD1}$, $I_{DD2}$, or $I_{DD3}$ and $I_{WDT}$ when the watchdog timer operates in STOP mode.

**3.** Current flowing only to the A/D converter. The current value is the sum of $I_{DD1}$ or $I_{DD2}$ and $I_{ADC}$ when the A/D converter operates in operation mode or HALT mode.

**4.** Current flowing only to the LVD circuit. The current value is the sum of $I_{DD1}$, $I_{DD2}$, or $I_{DD3}$ and $I_{LVD}$ when the LVD circuit operates in operation mode, HALT mode, or STOP mode.

**5.** Operating current that increases when the internal reference voltage is selected. This current flows even when conversion is stopped.

**6.** Current increased by the BGO operation. The current value is the sum of $I_{DD1}$ or $I_{DD2}$ and $I_{BGO}$ when the BGO operates in operation mode or HALT mode.

## 34.4 AC Characteristics

### 34.4.1 Basic Operation

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (1/2)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Instruction cycle (minimum instruction execution time) | $T_{CY}$ | High-speed on-chip oscillator clock operation | | 0.03125 | | 1 | μs |
| | | High-speed system clock operation | | 0.05 | | 1 | μs |
| | | PLL clock operation | | 0.03125 | | 1 | μs |
| | | Subsystem clock operation | | 28.5 | 30.5 | 34.5 | μs |
| | | Low-speed on-chip oscillator clock operation | | | 66.6 | | μs |
| | | In self programming mode | | 0.03125 | | 1 | μs |
| CPU/peripheral hardware clock frequency | $f_{CLK}$ | | | 0.03125 | | 66.6 | μs |
| External system clock frequency | $f_{EX}$ | | | 1.0 | | 20.0 | MHz |
| | $f_{EXS}$ | | | 29 | | 35 | kHz |
| External system clock input high-level width, low-level width | $t_{EXH}$, $t_{EXL}$ | | | 24 | | | ns |
| | $t_{EXHS}$, $t_{EXLS}$ | | | 13.7 | | | μs |
| TI00 to TI07, TI10 to TI17 input high-level width, low-level width | $t_{TIH}$, $t_{TIL}$ | | | $1/f_{MCK}$+10 | | | ns |
| TO00 to TO07, TO10 to TO17 output frequency | $f_{TO}$ | All TO pins, Normal slew rate, C = 30 pF | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 16 | MHz |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 8 | MHz |
| | | TO01, TO06, TO07, TO11, TO13 only, Special slew rate, C = 30 pF | | | | 2 | MHz |
| PCLBUZ0 output frequency | $f_{PCL}$ | Normal slew rate C = 30 pF | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 16 | MHz |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 8 | MHz |
| | | Special slew rate C = 30 pF | | | | 2 | MHz |
| Timer RJ input cycle | $t_C$ | TRJIO0 | | 100 | | | ns |
| Timer RJ input high-level width, low-level width | $t_{WH}$, $t_{WL}$ | TRJIO0 | | 40 | | | ns |
| Interrupt input high-level width, low-level width | $t_{INTH}$, $t_{INTL}$ | INTP0 to INTP13 Note | | 1 | | | μs |
| KR0 to KR7 key interrupt tinput low-level width | $t_{KR}$ | | | 250 | | | ns |
| $\overline{RESET}$ low-level width | $t_{RSL}$ | | | 10 | | | μs |

<R> **Note** Pins $\overline{RESET}$, INTP0 to INTP3, INTP12, and INTP13 have noise filters for transient levels lasting less than 100 ns.

**Caution Excluding the error in oscillation frequency accuracy**.

**Remark** $f_{MCK}$: Timer array unit operation clock frequency

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (2/2)**

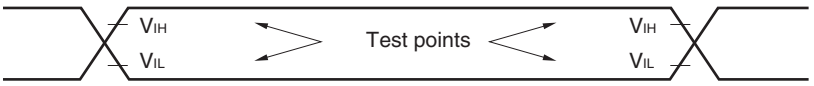
| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Port output rise time, port output fall time | $t_{RO}$, $t_{FO}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P96, P97, P106, P107, P120, P125 to P127, P130, P140, P150 to 157 (normal slew rate) C = 30 pF | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 25 | ns |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 55 | ns |
| | | P10, P12, P14, P30, P120, P140 (special slew rate) C = 30 pF | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | 25 Note | 60 | ns |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 100 | ns |

**Note** $T_A$ = +25°C, $EV_{DD0}$ = 5.0 V

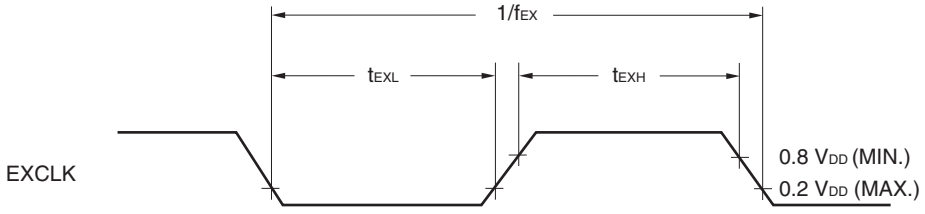
**Caution Excluding the error in oscillation frequency accuracy**.

**Remark** $f_{MCK}$: Timer array unit operation clock frequency

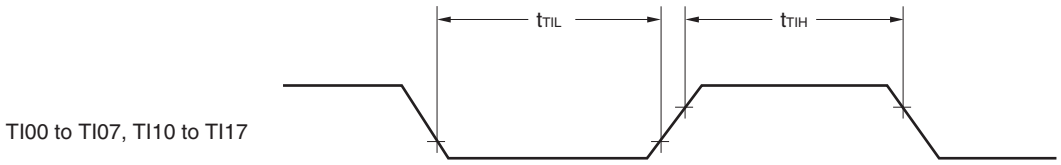
AC Timing Test Points

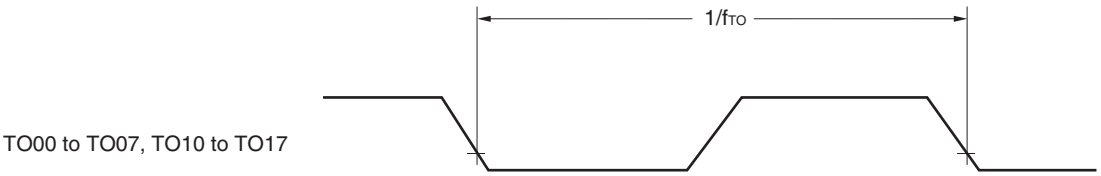



External System Clock Timing

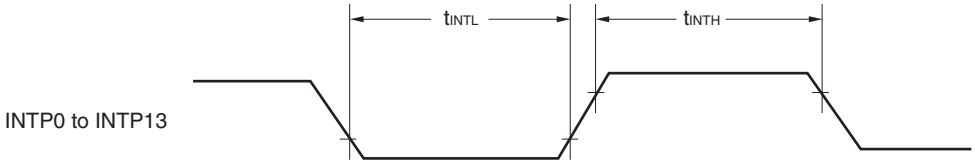


TI/TO Timing

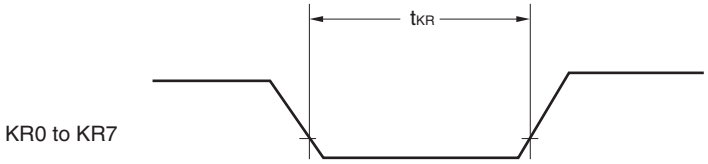


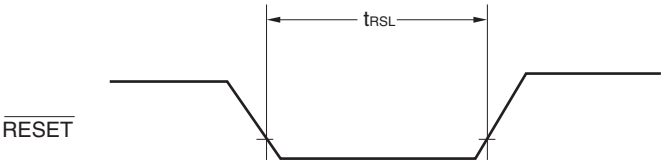

Interrupt Request Input Timing



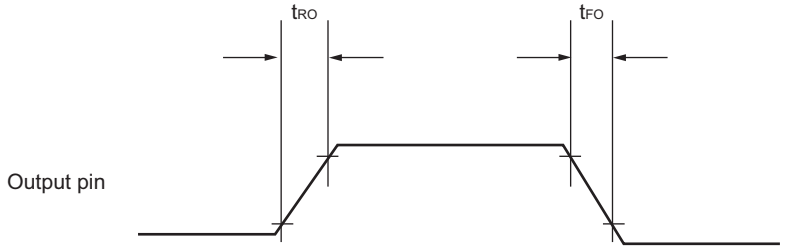

Key Interrupt Input Timing




$\overline{\text{RESET}}$ Input Timing




Output Rising and Falling Timing



## 34.5 Peripheral Functions Characteristics

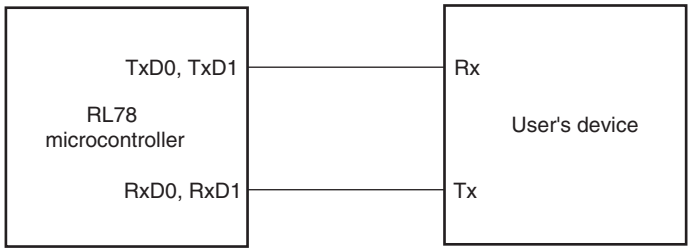
### 34.5.1 Serial Array Unit

**(1) During communication at same potential (UART mode) (dedicated baud rate generator output)**

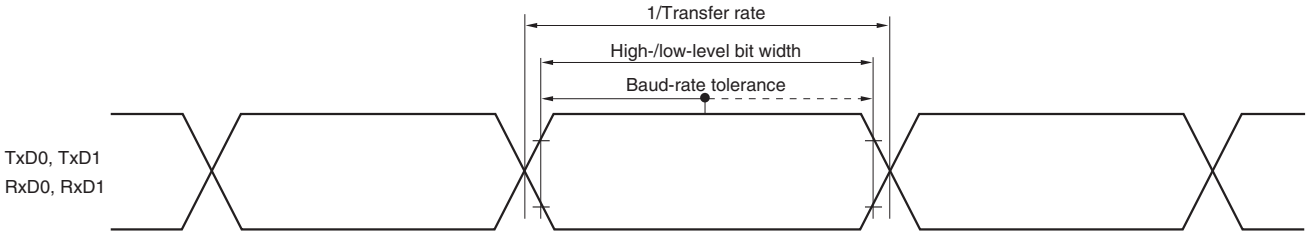
**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Transfer rate | - | | | | | $f_{MCK}$/6 | bps |
| | | $f_{CLK}$ = 32 MHz, $f_{MCK}$ = $f_{CLK}$ | Normal slew rate | | | 5.3 | Mbps |
| | | | Special slew rate | | | 2 | Mbps |

UART mode connection diagram (during communication at same potential)




UART mode bit width (during communication at same potential) (reference)




**Caution Select the normal input buffer for the RxD0 pin and RxD1 pin and normal output mode for the TxD0 pin and TxD1 pin.**

**Remark** $f_{MCK}$: Serial array unit operation clock frequency

### (2) During communication at same potential (CSI mode) (master mode, $\overline{SCKp}$ … internal clock output, normal slew rate)

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY1}$ | | 125 Note 5 | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH1}$, $t_{KL1}$ | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | $t_{KCY1}$/2 – 12 | | | ns |
| | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | $t_{KCY1}$/2 – 18 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑) Note 1 | $t_{SIK1}$ | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 44 | | | ns |
| | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 55 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑) Note 2 | $t_{KSI1}$ | | 30 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output Note 3 | $t_{KSO1}$ | C = 30 pF Note 4 | | | 40 | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**4.** C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

**5.** $t_{KCY1} \geq 4/f_{CLK}$ must also be satisfied.

**Caution Select the normal input buffer for the SIp pin and normal output mode for the SOp pin and $\overline{SCKp}$ pin.**

**Remark** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

### (3) During communication at same potential (CSI mode) (master mode, $\overline{SCKp}$ … internal clock output, special slew rate)

**($T_A$ = -40 to +105°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY1}$ | | 500 Note 5 | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH1}$, $t_{KL1}$ | | $t_{KCY1}$/2 – 60 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑) Note 1 | $t_{SIK1}$ | | 120 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑) Note 2 | $t_{KSI1}$ | | 80 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output Note 3 | $t_{KSO1}$ | C = 30 pF Note 4 | | | 90 | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**4.** C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

**5.** $t_{KCY1} \geq 4/f_{CLK}$ must also be satisfied.

**Caution Select the normal input buffer for the SIp pin and normal output mode and special slew rate for the SOp pin and $\overline{SCKp}$ pin.**

**Remark** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

### (4) During communication at same potential (CSI mode) (slave mode, $\overline{SCKp}$ … external clock input, normal slew rate)

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY2}$ | | | $8/f_{MCK}$ | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH2}$, $t_{KL2}$ | | | $t_{KCY2}/2$ | | | ns |
| SIp setup time (to $\overline{SCKp}$↑)[Note 1] | $t_{SIK2}$ | | | $1/f_{MCK}$ + 20 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑)[Note 2] | $t_{KSI2}$ | | | $1/f_{MCK}$ + 31 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output[Note 3] | $t_{KSO2}$ | C = 30 pF[Note 4] | 4.0V ≤ $V_{DD}$ = $EV_{DD0}$ = $EV_{DD1}$ ≤ 5.5V | | | $2/f_{MCK}$ + 44 | ns |
| | | | 2.7V ≤ $V_{DD}$ = $EV_{DD0}$ = $EV_{DD1}$ < 4.0V | | | $2/f_{MCK}$ + 57 | ns |
| $\overline{SSIp}$ setup time | $t_{SSIK}$ | DAP = 0 | | 120 | | | ns |
| | | DAP = 1 | | $1/f_{MCK}$ + 120 | | | ns |
| $\overline{SSIp}$ hold time | $t_{KSSI}$ | DAP = 0 | | $1/f_{MCK}$ + 120 | | | ns |
| | | DAP = 1 | | 120 | | | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**4.** C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

**Caution Select the normal input buffer for the SIp, $\overline{SCKp}$ and $\overline{SSIp}$ pins and normal output mode for the SOp pin.**

**Remarks 1.** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

**2.** $f_{MCK}$: Serial array unit operation clock frequency

### (5) During communication at same potential (CSI mode) (slave mode, $\overline{SCKp}$ … external clock input, special slew rate)

**($T_A$ = -40 to +105°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY2}$ | 20 MHz < $f_{MCK}$ | 10/$f_{MCK}$ | | | ns |
| | | 10 MHz < $f_{MCK}$ ≤ 20 MHz | 8/$f_{MCK}$ | | | ns |
| | | $f_{MCK}$ ≤ 10 MHz | 6/$f_{MCK}$ | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH2}$, $t_{KL2}$ | | $t_{KCY2}$/2 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑)Note1 | $t_{SIK2}$ | | 1/$f_{MCK}$ + 50 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑)Note 2 | $t_{KSI2}$ | | 1/$f_{MCK}$ + 50 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp outputNote 3 | $t_{KSO2}$ | C = 30 pFNote 4 | | | 2/$f_{MCK}$ + 80 | ns |
| $\overline{SSIp}$ setup time | $t_{SSIK}$ | DAP = 0 | 120 | | | ns |
| | | DAP = 1 | 1/$f_{MCK}$ + 120 | | | ns |
| $\overline{SSIp}$ hold time | $t_{KSSI}$ | DAP = 0 | 1/$f_{MCK}$ + 120 | | | ns |
| | | DAP = 1 | 120 | | | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from $\overline{SCKp}$↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

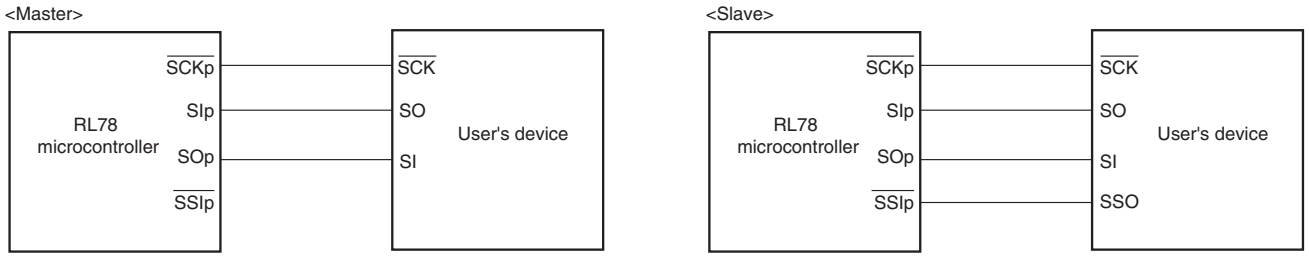
**4.** C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

**Caution Select the normal input buffer for the SIp, $\overline{SCKp}$ and $\overline{SSIp}$ pins and normal output mode and special slew rate for the SOp pin.**

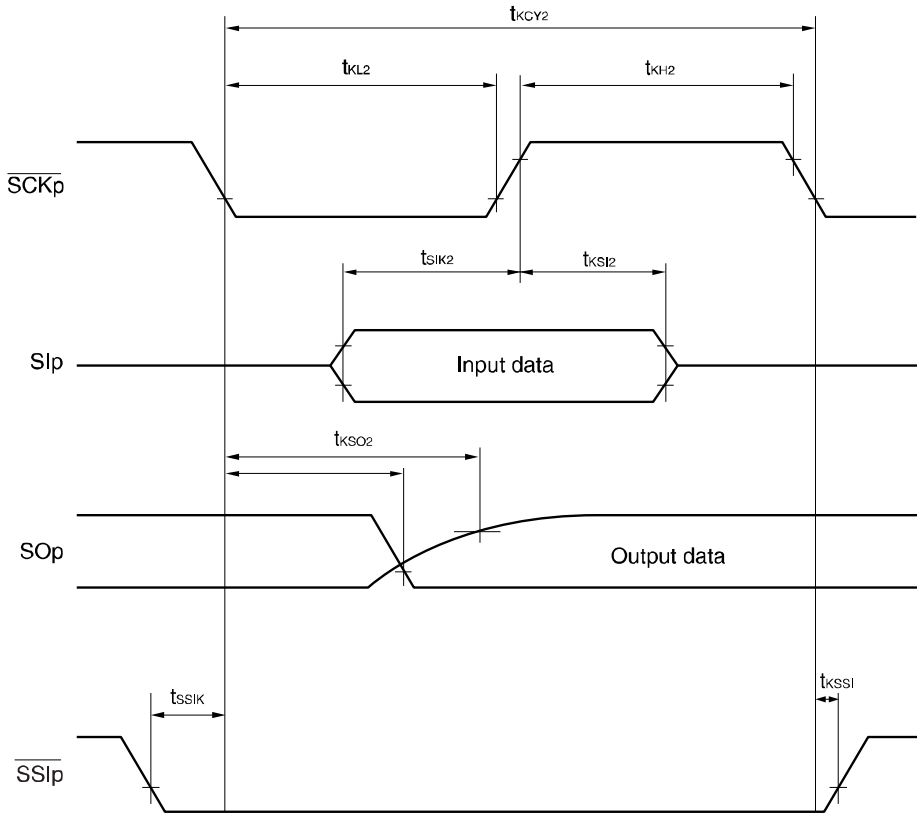
**Remarks 1.** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

**2.** $f_{MCK}$: Serial array unit operation clock frequency

CSI mode connection diagram (during communication at same potential)

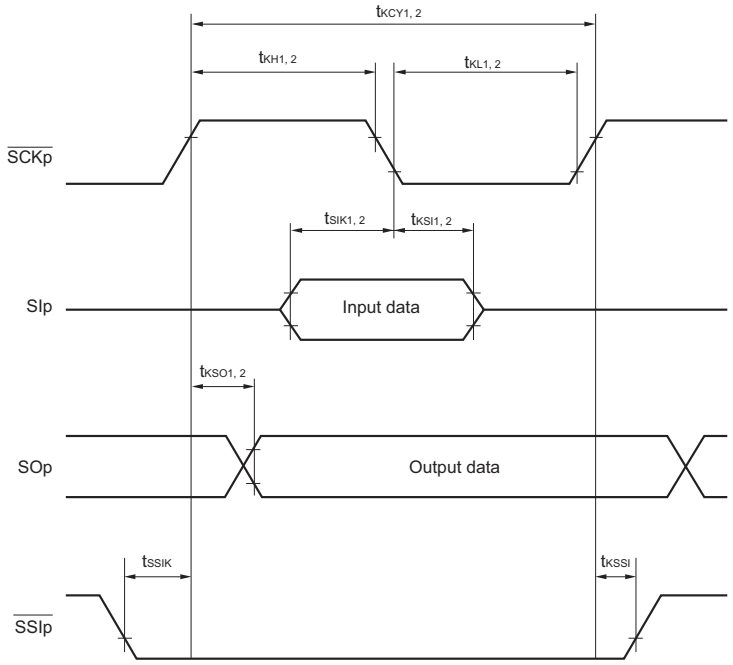



CSI mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)




**Remark** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

CSI mode serial transfer timing (during communication at same potential)
(When DAPmn= 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)




**Remark** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

### (6) During communication at same potential (simplified $I^2C$ mode)

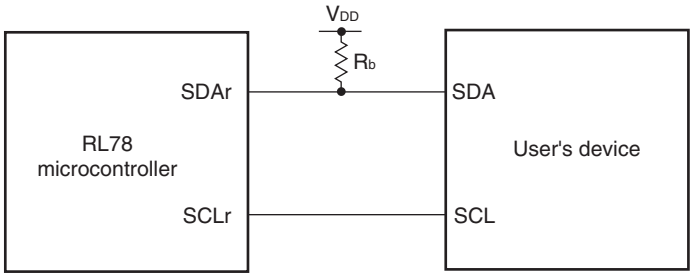
**(SDAr: N-ch open-drain output ($EV_{DD0}$ tolerance) mode, SCLr: normal output mode)**

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

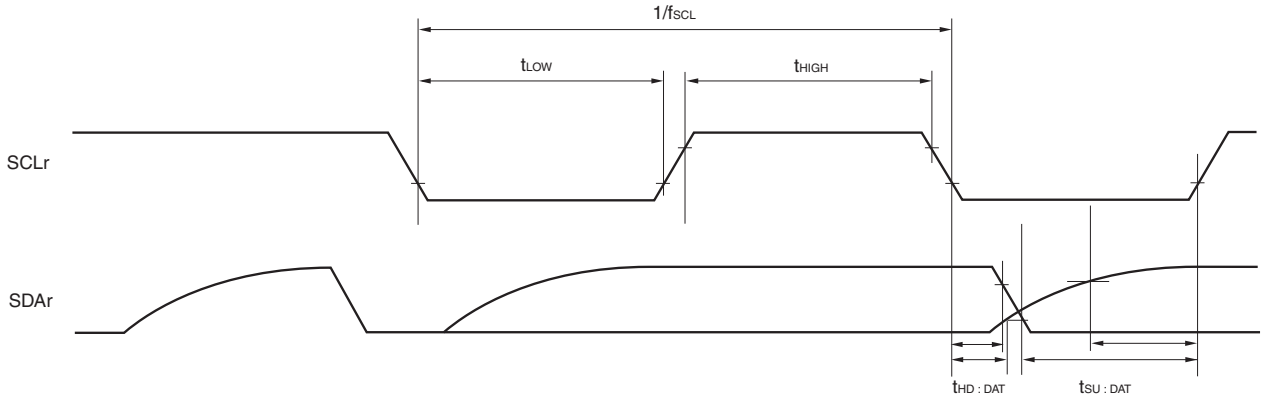
| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| SCLr clock frequency | $f_{SCL}$ | | | | 1000[Note] | kHz |
| Hold time when SCLr = "L" | $t_{LOW}$ | | 475 | | | ns |
| Hold time when SCLr = "H" | $t_{HIGH}$ | | 475 | | | ns |
| Data setup time (reception) | $t_{SU:DAT}$ | | 1/$f_{MCK}$ + 85 | | | ns |
| Data hold time (transmission) | $t_{HD:DAT}$ | $C_b$ = 50 pF, $R_b$ = 2.7 kΩ | 0 | | 305 | ns |

**Note** $f_{CLK} \le f_{MCK}/4$ must also be satisfied.

Simplified $I^2C$ mode connection diagram (during communication at same potential)




Simplified $I^2C$ mode serial transfer timing (during communication at same potential)




**Caution Select the normal input buffer and N-ch open-drain output mode for the SDAr pin and normal output mode for the SCLr pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (SDAr) pull-up resistance, $C_b$ [F]: Communication line (SCLr, SDAr) load capacitance

**2.** r: IICr (r = 00, 01, 10, 11)

**3.** $f_{MCK}$: Serial array unit operation clock frequency

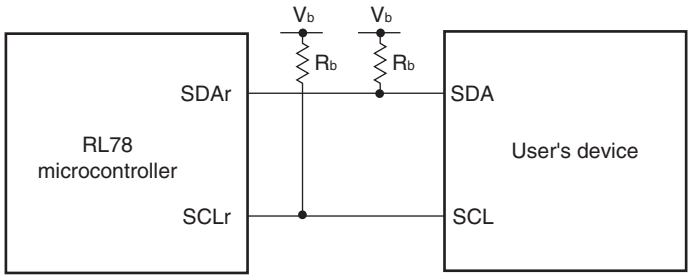
### (7) During communication at same potential (simplified I²C mode) (SDAr and SCLr: N-ch open-drain output (EVDD0 tolerance) mode)

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| SCLr clock frequency | $f_{SCL}$ | | | 400[Note] | kHz |
| Hold time when SCLr = "L" | $t_{LOW}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $C_b$ = 100 pF, $R_b$ = 1.7 kΩ | 1300 | | ns |
| <R> | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $C_b$ = 100 pF, $R_b$ = 2.7 kΩ | | | |
| Hold time when SCLr = "H" | $t_{HIGH}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $C_b$ = 100 pF, $R_b$ = 1.7 kΩ | 600 | | ns |
| <R> | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $C_b$ = 100 pF, $R_b$ = 2.7 kΩ | | | |
| Data setup time (reception) | $t_{SU:DAT}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $C_b$ = 100 pF, $R_b$ = 1.7 kΩ | 1/$f_{MCK}$ + 120 | | ns |
| <R> | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $C_b$ = 100 pF, $R_b$ = 2.7 kΩ | 1/$f_{MCK}$ + 270 | | ns |
| Data hold time (transmission) | $t_{HD:DAT}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $C_b$ = 100 pF, $R_b$ = 1.7 kΩ | 0 | 300 | ns |
| <R> | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $C_b$ = 100 pF, $R_b$ = 2.7 kΩ | | | |

**Note** $f_{CLK}$ ≤ $f_{MCK}$/4 must also be satisfied.

Simplified I²C mode connection diagram (during communication at same potential)



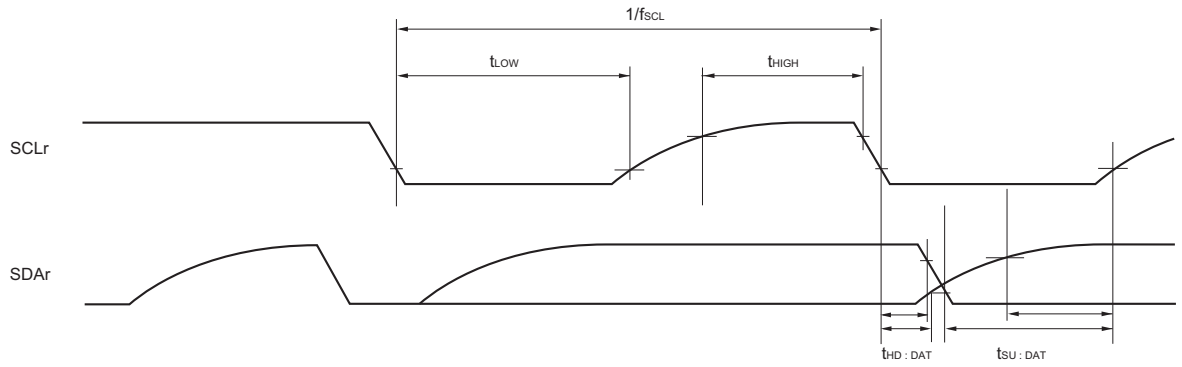

**Caution Select the normal input buffer and N-ch open-drain output mode for the SDAr pin and SCLr pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (SDAr, SCLr) pull-up resistance, $C_b$ [F]: Communication line (SDAr, SCLr) load capacitance, $V_b$[V]: Communication line voltage

**2.** r: IICr (r = 00, 01, 10, 11)

**3.** $f_{MCK}$: Serial array unit operation clock frequency

Simplified $I^2C$ mode serial transfer timing (during communication at same potential)




**Remark** r: IICr (r = 00, 01, 10, 11)

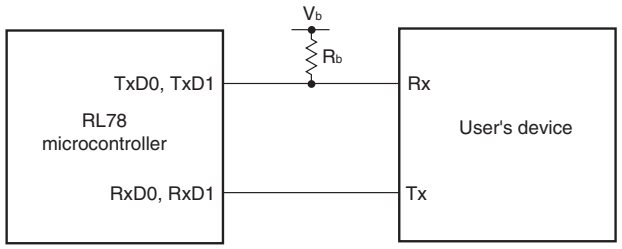
### (8) Communication at different potential (UART mode) (TxD output buffer: N-ch open-drain, RxD input buffer: TTL)

**($T_A$ = -40 to +105°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

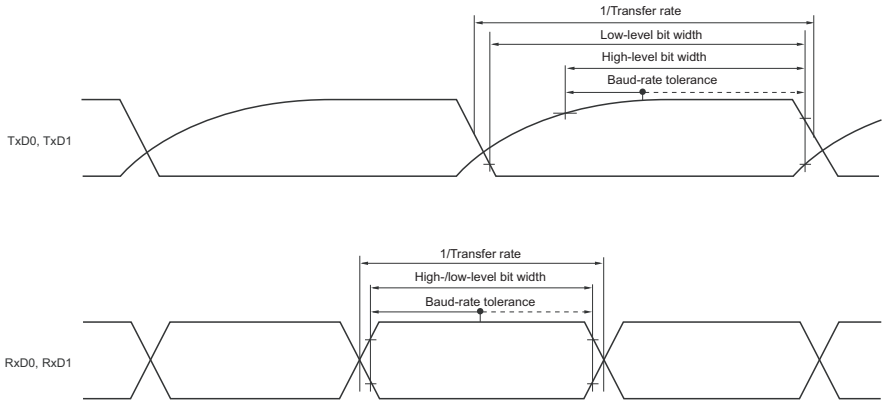
| Parameter | Symbol | Conditions | | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|---|
| Transfer rate | – | Reception | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $V_{IH}$ = 2.2 V, $V_{IL}$ = 0.8 V | | | | $f_{MCK}$/6 | bps |
| | | | | Theoretical value of the maximum transfer rate Note ($C_b$ = 30 pF) | | | 5.3 | Mbps |
| | | Transmission | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $V_{OH}$ = 2.2 V, $V_{OL}$ = 0.8 V | | | | Smaller number of the values given by $f_{MCK}$/6 and expression 1 is applicable. | bps |
| | | | | Theoretical value of the maximum transfer rate Note ($C_b$ = 30 pF) Normal slew rate | | | 5.3 | Mbps |

**Note** Expression 1: Maximum transfer rate = $1 / [\{-C_b \times R_b \times \ln (1 - 2.2/V_b)\} \times 3]$

UART mode connection diagram (during communication at different potential)




UART mode bit width (during communication at different potential) (reference)




**Caution Select the TTL input buffer for the RxD0 pin and RxD1 pin and N-ch open-drain output mode for the TxD0 pin and TxD1 pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (TxD) pull-up resistance, $C_b$ [F]: Communication line (TxD) load capacitance, $V_b$ [V]: Communication line voltage

**2.** $f_{MCK}$: Serial array unit operation clock frequency

### (9) During communication at different potential (3-V supply system) (CSI mode) (master mode, $\overline{SCKp}$ … internal clock output, normal slew rate)

**($T_A$ = -40 to +105°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

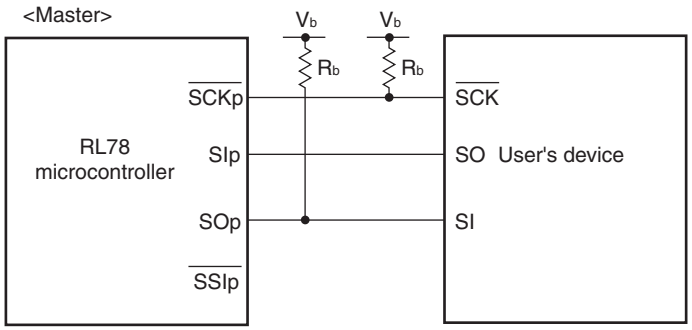
| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 400 Note3 | | | ns |
| $\overline{SCKp}$ high-level width | $t_{KH1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | $t_{KCY1}$/2 – 75 | | | ns |
| $\overline{SCKp}$ low-level width | $t_{KL1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | $t_{KCY1}$/2 – 20 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑) Note 1 | $t_{SIK1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 150 | | | ns |
| SIp setup time (to $\overline{SCKp}$↓) Note 2 | $t_{SIK1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 70 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑) Note 1 | $t_{KSI1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 30 | | | ns |
| SIp hold time (from $\overline{SCKp}$↓) Note 2 | $t_{KSI1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 30 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output Note1 | $t_{KSO1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | | | 120 | ns |
| Delay time from $\overline{SCKp}$↑ to SOp output Note2 | $t_{KSO1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | | | 40 | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

**2.** When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**3.** $t_{KCY1} \geq 4/f_{CLK}$ must also be satisfied.

CSI mode connection diagram (during communication at different potential)




**Caution Select the TTL input buffer for the SIp pin and N-ch open-drain output mode for the SOp pin and $\overline{\text{SCKp}}$ pin.**

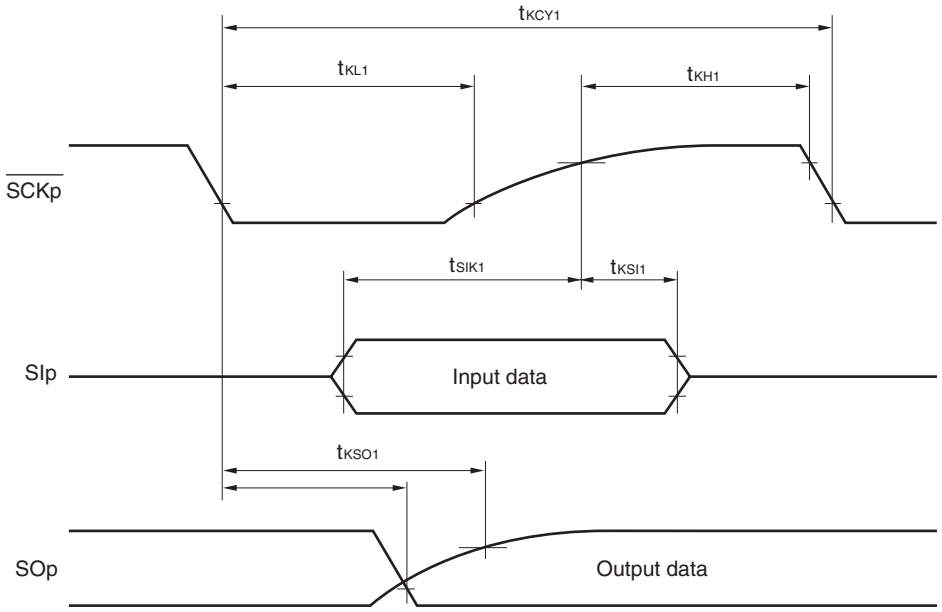
**Remarks 1.** $R_b$ [Ω]: Communication line ($\overline{\text{SCKp}}$, SOp) pull-up resistance, $C_b$ [F]: Communication line (SOp, $\overline{\text{SCKp}}$) load capacitance, $V_b$ [V]: Communication line voltage

**2.** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

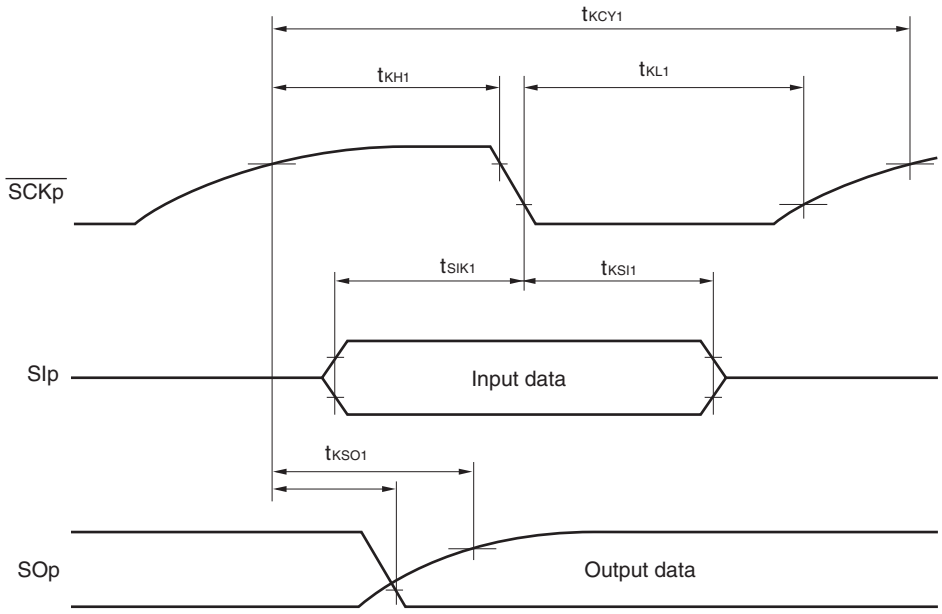
**3.** AC characteristics of the serial array unit during communication at different potential in CSI mode are measured with the $V_{IH}$ and $V_{IL}$ below:

When 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, 2.7 V ≤ $V_b$ ≤ 4.0 V: $V_{IH}$ = 2.2 V, $V_{IL}$ = 0.8 V

CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)




CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn= 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



### (10) During communication at different potential (3-V supply system) (CSI mode) (slave mode, $\overline{SCKp}$ ... external clock input, normal slew rate)

**($T_A$ = -40 to +105°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

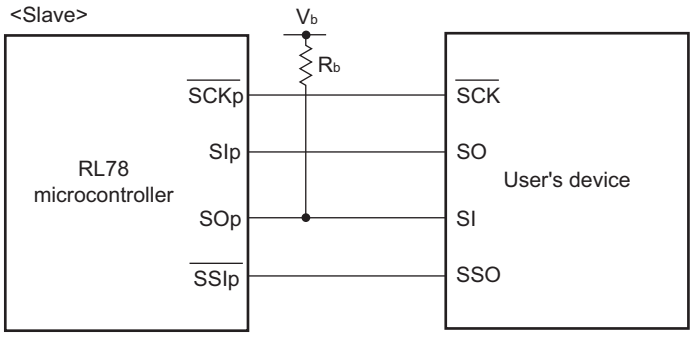
| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY2}$ | 2.7 V ≤ $V_b$ ≤ $V_{DD}$ | 24 MHz < $f_{MCK}$ | 14/$f_{MCK}$ | | | ns |
| | | | 20 MHz < $f_{MCK}$ ≤ 24 MHz | 12/$f_{MCK}$ | | | ns |
| | | | 8 MHz < $f_{MCK}$ ≤ 20 MHz | 10/$f_{MCK}$ | | | ns |
| | | | 4 MHz < $f_{MCK}$ ≤ 8 MHz | 8/$f_{MCK}$ | | | ns |
| | | | $f_{MCK}$ ≤ 4 MHz | 6/$f_{MCK}$ | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH2}$, $t_{KL2}$ | 2.7 V ≤ $V_b$ ≤ $V_{DD}$ | | $t_{KCY2}$/2 – 20 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑)[Note 1] | $t_{SIK2}$ | | | 90 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑)[Note 2] | $t_{KSI2}$ | | | 1/$f_{MCK}$ + 50 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output[Note 3] | $t_{KSO2}$ | 2.7 V ≤ $V_b$ ≤ $V_{DD}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | | | | 2/$f_{MCK}$ + 120 | ns |
| $\overline{SSIp}$ setup time | $t_{SSIK}$ | DAP = 0 | | 120 | | | ns |
| | | DAP = 1 | | 1/$f_{MCK}$ + 120 | | | ns |
| $\overline{SSIp}$ hold time | $t_{KSSI}$ | DAP = 0 | | 1/$f_{MCK}$ + 120 | | | ns |
| | | DAP = 1 | | 120 | | | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

CSI mode connection diagram (during communication at different potential)



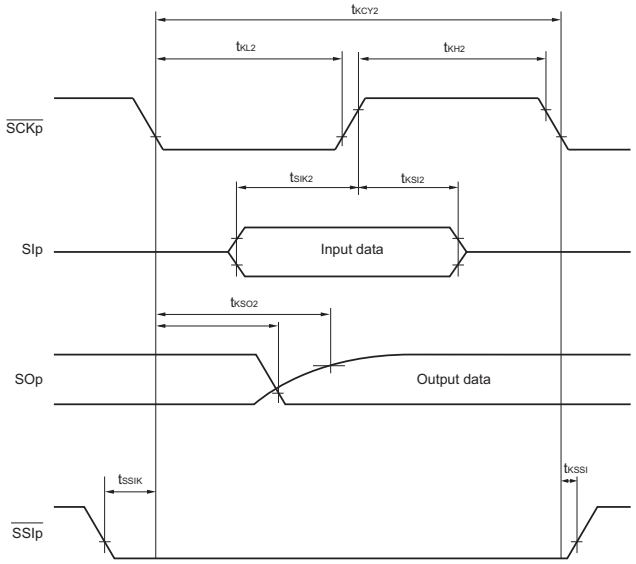

**Caution Select the TTL input buffer for the SIp, $\overline{\text{SCKp}}$ and $\overline{\text{SSIp}}$ pins and N-ch open-drain output mode for the SOp pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (SOp) pull-up resistance, $C_b$ [F]: Communication line (SOp) load capacitance, $V_b$ [V]: Communication line voltage

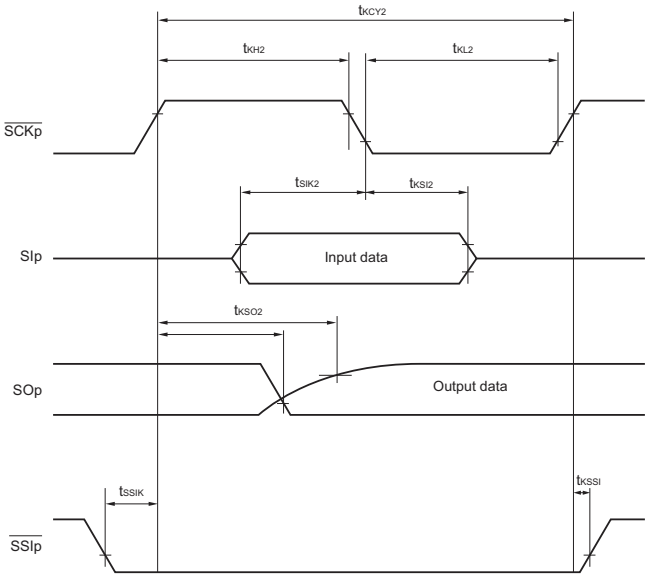
**2.** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

**3.** AC characteristics of the serial array unit during communication at different potential in CSI mode are measured with the $V_{IH}$ and $V_{IL}$ below:
When 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, 2.7 V ≤ $V_b$ ≤ 4.0 V: $V_{IH}$ = 2.2 V, $V_{IL}$ = 0.8 V

CSI mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)




CSI mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn= 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



**(11) During communication at different potential (3-V supply system) (simplified $I^2C$ mode)**

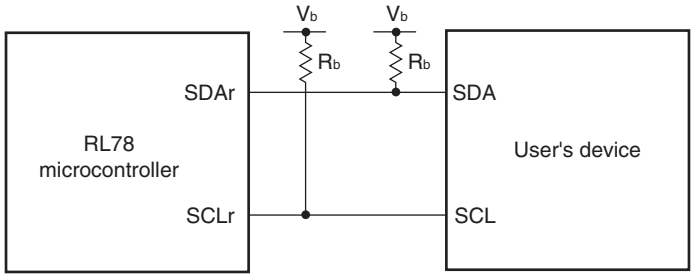
**(SDAr: TTL input buffer mode or N-ch open-drain output ($EV_{DD0}$ tolerance) mode, SCLr: N-ch open-drain output ($EV_{DD0}$ tolerance) mode)**

**($T_A$ = -40 to +105°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

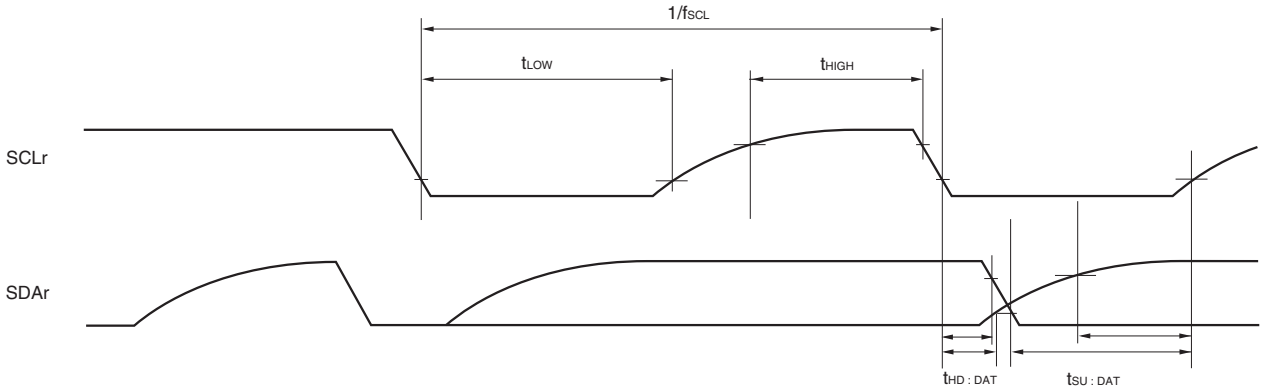
| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| SCLr clock frequency | $f_{SCL}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | | 400[Note] | kHz |
| Hold time when SCLr = "L" | $t_{LOW}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | 1200 | | ns |
| Hold time when SCLr = "H" | $t_{HIGH}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | 600 | | ns |
| Data setup time (reception) | $t_{SU:DAT}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | 135 + 1/$f_{MCK}$ | | ns |
| Data hold time (transmission) | $t_{HD:DAT}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | 0 | 140 | ns |

**Note** $f_{SCL}$ ≤ $f_{MCK}$/4 must also be satisfied.

Simplified I²C mode connection diagram (during communication at different potential)




Simplified I²C mode serial transfer timing (during communication at different potential)




**Caution Select the TTL input buffer and the N-ch open-drain output mode for the SDAr pin and N-ch open-drain output mode for the SCLr pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (SDAr, SCLr) pull-up resistance, $C_b$ [F]: Communication line (SDAr, SCLr) load capacitance, $V_b$ [V]: Communication line voltage

**2.** $f_{MCK}$: Serial array unit operation clock frequency

### 34.5.2 Serial Interface IICA

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | Normal Mode | | Fast Mode | | Fast Mode Plus | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| SCLA0 clock frequency | $f_{SCL}$ | Fast mode plus: 10 MHz ≤ $f_{CLK}$ | | | | | 0 | 1000 | kHz |
| | | Fast mode: 3.5 MHz ≤ $f_{CLK}$ | | | 0 | 400 | | | kHz |
| | | Normal mode: 1 MHz ≤ $f_{CLK}$ | 0 | 100 | | | | | kHz |
| Setup time of restart condition[Note 1] | $t_{SU:STA}$ | | 4.7 | | 0.6 | | 0.26 | | µs |
| Hold time | $t_{HD:STA}$ | | 4.0 | | 0.6 | | 0.26 | | µs |
| Hold time when SCLA0 = "L" | $t_{LOW}$ | | 4.7 | | 1.3 | | 0.5 | | µs |
| Hold time when SCLA0 = "H" | $t_{HIGH}$ | | 4.0 | | 0.6 | | 0.26 | | µs |
| Data setup time (reception) | $t_{SU:DAT}$ | | 250 | | 100 | | 50 | | ns |
| Data hold time (transmission)[Note 2] | $t_{HD:DAT}$ | | 0 | 3.45 | 0 | 0.9 | 0 | | µs |
| Setup time of stop condition | $t_{SU:STO}$ | | 4.0 | | 0.6 | | 0.26 | | µs |
| Bus-free time | $t_{BUF}$ | | 4.7 | | 1.3 | | 0.5 | | µs |

**Notes 1.** The first clock pulse is generated after this period when the start/restart condition is detected.

**2.** The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a wait state is inserted in the $\overline{ACK}$ (acknowledge) timing.

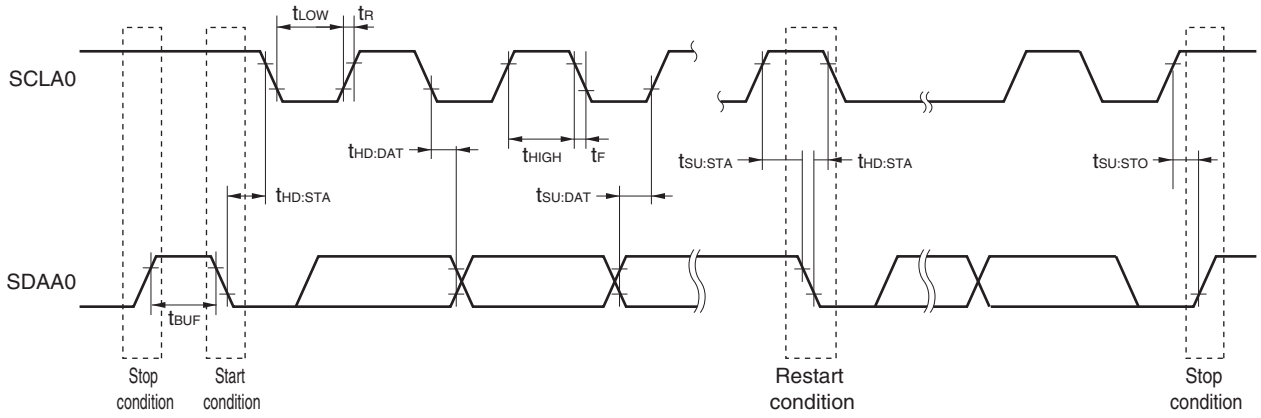
**Remark** The maximum value of $C_b$ (communication line capacitance) and the value of $R_b$ (communication line pull-up resistor) at that time in each mode are as follows.

Standard mode: $C_b$ = 400 pF, $R_b$ = 2.7 kΩ

Fast mode: $C_b$ = 320 pF, $R_b$ = 1.1 kΩ

Fast mode plus: $C_b$ = 120 pF, $R_b$ = 1.1 kΩ

IICA serial transfer timing



### 34.5.3 On-chip Debug (UART)

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Transfer rate | - | | 115.2 k | | 1 M | bps |

### 34.5.4 LIN/UART Module (RLIN3) UART Mode

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Transfer rate | - | Operation mode, HALT mode | LIN communication clock source ($f_{CLK}$ or $f_{MX}$): 4 to 32 MHz | | | 5333 | kbps |
| | | SNOOZE mode | LIN communication clock source ($f_{CLK}$): 1 to 32 MHz FRQSEL4 = 0 in the user option byte (000C2H/020C2H) | | | 4.8 | |
| | | | LIN communication clock source ($f_{CLK}$): 1 to 32 MHz FRQSEL4 = 1 in the user option byte (000C2H/020C2H) | | | 2.4 | |

## 34.6 Analog Characteristics

### 34.6.1 A/D Converter Characteristics

**(1) When $AV_{REF}$ (+) = $AV_{REFP}$/ANI0 (ADREFP1 = 0, ADREFP0 = 1), $AV_{REF}$ (-) = $AV_{REFM}$/ANI1 (ADREFM = 1), target ANI pin: ANI2 to ANI23 (power supply: $V_{DD}$)**

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V, Reference voltage (+) = $AV_{REFP}$, Reference voltage (-) = $AV_{REFM}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | 8 | | 10 | bit |
| Overall error[Note 1] | AINL | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.2 | ±3.0 | LSB |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.2 | ±3.5 | LSB |
| Conversion time | $t_{CONV}$ | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 2.125 | | 39 | μs |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 3.1875 | | 39 | μs |
| Zero-scale error[Notes 1, 2] | EZS | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.25 | %FSR |
| Full-scale error[Notes 1, 2] | EFS | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.25 | %FSR |
| Integral linearity error[Note 1] | ILE | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.5 | LSB |
| Differential linearity error[Note 1] | DLE | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±1.5 | LSB |
| Reference voltage (+) | $AV_{REFP}$ | | | 2.7 | | $V_{DD}$ | V |
| Analog input voltage | $V_{AIN}$ | | | 0 | | $AV_{REFP}$ | V |
| Internal reference voltage (+) | $V_{BGR}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.38 | 1.45 | 1.5 | V |

**Notes 1.** Excludes quantization error (±1/2 LSB).

**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

**(2) When $AV_{REF}$ (+) = $AV_{REFP}$/ANI0 (ADREFP1 = 0, ADREFP0 = 1), $AV_{REF}$ (-) = $AV_{REFM}$/ANI1 (ADREFM = 1), target ANI pin: ANI24 to ANI30 (power supply: $EV_{DD0}$)**

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V, Reference voltage (+) = $AV_{REFP}$, Reference voltage (-) = $AV_{REFM}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | 8 | | 10 | bit |
| Overall error[Note 1] | AINL | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.2 | ±4.5 | LSB |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.2 | ±5.0 | LSB |
| Conversion time | $t_{CONV}$ | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 2.125 | | 39 | μs |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 3.1875 | | 39 | μs |
| Zero-scale error[Notes 1, 2] | EZS | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.35 | %FSR |
| Full-scale error[Notes 1, 2] | EFS | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.35 | %FSR |
| Integral linearity error[Note 1] | ILE | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±3.5 | LSB |
| Differential linearity error[Note 1] | DLE | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.0 | LSB |
| Reference voltage (+) | $AV_{REFP}$ | | | 2.7 | | $V_{DD}$ | V |
| Analog input voltage | $V_{AIN}$ | | | 0 | | $AV_{REFP}$ and $EV_{DD0}$ | V |
| Internal reference voltage (+) | $V_{BGR}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.38 | 1.45 | 1.5 | V |

**Notes 1.** Excludes quantization error (±1/2 LSB).

**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

**(3) When $AV_{REF}$ (+) = $V_{DD}$ (ADREFP1 = 0, ADREFP0 = 0), $AV_{REF}$ (-) = $V_{SS}$ (ADREFM = 0), target ANI pin: ANI0 to ANI23, ANI24 to ANI30**

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V, Reference voltage (+) = $V_{DD}$, Reference voltage (-) = $V_{SS}$)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | 8 | | 10 | bit |
| Overall error[Note 1] | AINL | 10-bit resolution ANI0 to ANI23 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.2 | ±5.0 | LSB |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.2 | ±5.5 | LSB |
| | | 10-bit resolution ANI24 to ANI30 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.2 | ±6.5 | LSB |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.2 | ±7.0 | LSB |
| Conversion time | $t_{CONV}$ | 10-bit resolution | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 2.125 | | 39 | μs |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 3.1875 | | 39 | μs |
| Zero-scale error[Notes 1, 2] | EZS | 10-bit resolution ANI0 to ANI23 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.50 | %FSR |
| | | 10-bit resolution ANI24 to ANI30 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.60 | %FSR |
| Full-scale error[Notes 1, 2] | EFS | 10-bit resolution ANI0 to ANI23 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.50 | %FSR |
| | | 10-bit resolution ANI24 to ANI30 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.60 | %FSR |
| Integral linearity error[Note 1] | ILE | 10-bit resolution ANI0 to ANI23 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±3.5 | LSB |
| | | 10-bit resolution ANI24 to ANI30 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±4.0 | LSB |
| Differential linearity error[Note 1] | DLE | 10-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.0 | LSB |
| Analog input voltage | $V_{AIN}$ | ANI0 to ANI23[Note 3] | | 0 | | $V_{DD}$ | V |
| | | ANI24 to ANI30[Note 3] | | $EV_{SS}$ | | $EV_{DD0}$ | V |
| Internal reference voltage (+) | $V_{BGR}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.38 | 1.45 | 1.5 | V |

**Notes 1.** Excludes quantization error (±1/2 LSB).

**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

**3.** The number of pins depends on the product. For details, refer to **2.1 Pin Function List**.

**(4) When $AV_{REF}$ (+) = internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), $AV_{REF}$ (-) = $AV_{REFM}$/ANI1 (ADREFM = 1), target ANI pin: ANI0 to ANI23, ANI24 to ANI30**

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V, Reference voltage (+) = $V_{BGR}$, Reference voltage (-) = $AV_{REFM}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | | 8 | | bit |
| Conversion time | $t_{CONV}$ | 8-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | 17 | | 39 | μs |
| Zero-scale error[Notes 1, 2] | EZS | 8-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.60 | %FSR |
| Integral linearity error[Note 1] | ILE | 8-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.0 | LSB |
| Differential linearity error[Note 1] | DLE | 8-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±1.0 | LSB |
| Reference voltage (+) | $V_{BGR}$ | | | 1.38 | 1.45 | 1.5 | V |
| Analog input voltage | $V_{AIN}$ | | | 0 | | $V_{BGR}$ | V |

**Notes 1.** Excludes quantization error (±1/2 LSB).

**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

### 34.6.2 Temperatures Sensor Characteristics

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Temperature sensor output voltage | $V_{TMPS25}$ | Setting ADS register = 80H, $T_A$ = +25°C | | 1.1 | | V |
| Reference output voltage | $V_{CONST}$ | Setting ADS register = 81H | 1.38 | 1.45 | 1.5 | V |
| Temperature coefficient | $F_{VTMPS}$ | Temperature sensor that depends on the temperature | | -3.3 | | mV/°C |
| Operation stabilization wait time | $t_{AMP}$ | | 5 | | | μs |

### 34.6.3 D/A Converter Characteristics

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | | | 8 | bit |
| Overall error | AINL | Rload = 4 MΩ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.5 | LSB |
| | | Rload = 8 MΩ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.5 | LSB |
| Settling time | $t_{SET}$ | Cload = 20 pF | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | 3 | μs |

### 34.6.4 Comparator Characteristics

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input offset voltage | $V_{IOCMP}$ | | | ±5 | ±40 | mV |
| Input voltage range | $V_{ICMP}$ | | 0 | | $V_{DD}$ | V |
| Response time | $t_{CR}$, $t_{CF}$ | Input amplitude ±100 mV | | 70 | 200 | ns |
| Stabilization wait time during input channel switching[Note 1] | $t_{WAIT}$ | Input amplitude ±100 mV | 300 | | | ns |
| Operation stabilization wait time[Note 2] | $t_{CMP}$ | 3.3 V ≤ $V_{DD}$ ≤ 5.5 V | 1 | | | μs |
| | | 2.7 V ≤ $V_{DD}$ < 3.3 V | 3 | | | μs |

**Notes 1.** Period of time from when the comparator input channel is switched until the comparator is switched to output

**2.** Period of time from when the comparator operation is enabled (HCMPON bit in CMPCTL is set to 1) until the comparator satisfies the DC/AC characteristics.

### 34.6.5 POR Circuit Characteristics

**($T_A$ = -40 to +105°C, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Detection voltage[Note] | $V_{POR}$ | Power supply rise time | 1.48 | 1.56 | 1.62 | V |
| | $V_{PDR}$ | Power supply fall time | 1.47 | 1.55 | 1.61 | V |
| Minimum pulse width | $T_{PW}$ | | 300 | | | μs |
| Detection delay time | $T_{PD}$ | | | | 350 | μs |

**Note** This indicates the POR circuit characteristics, and normal operation is not guaranteed under the condition of less than lower limit operation voltage (2.7 V).

### 34.6.6 LVD Circuit Characteristics

#### (1) LVD detection voltage of interrupt mode or reset mode

**($T_A$ = -40 to +105°C, $V_{PDR} \le EV_{DD0} = EV_{DD1} = V_{DD} \le 5.5$ V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)**

| Parameter | | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Detection voltage | Supply voltage level | $V_{LVD0}$ | Power supply rise time | 4.62 | 4.74 | 4.84 | V |
| | | | Power supply fall time | 4.52 | 4.64 | 4.74 | V |
| | | $V_{LVD1}$ | Power supply rise time | 4.50 | 4.62 | 4.72 | V |
| | | | Power supply fall time | 4.40 | 4.52 | 4.62 | V |
| | | $V_{LVD2}$ | Power supply rise time | 4.30 | 4.42 | 4.51 | V |
| | | | Power supply fall time | 4.21 | 4.32 | 4.41 | V |
| | | $V_{LVD3}$ | Power supply rise time | 3.13 | 3.22 | 3.29 | V |
| | | | Power supply fall time | 3.07 | 3.15 | 3.22 | V |
| | | $V_{LVD4}$ | Power supply rise time | 2.95 | 3.02 | 3.09 | V |
| | | | Power supply fall time | 2.89 | 2.96 | 3.02 | V |
| | | $V_{LVD5}$ | Power supply rise time | 2.74 | 2.81 | 2.87 | V |
| | | | Power supply fall time | 2.68 Note | 2.75 | 2.81 | V |
| Minimum pulse width | | $t_{LW}$ | | 300 | | | μs |
| Detection delay time | | $t_{LD}$ | | | | 300 | μs |

**Note** The minimum value exceeds below the lower limit operation voltage (2.7 V), however, in reset mode, normal operation (same behavior when $V_{DD}$ = 2.7 V) is possible until a reset is effected at the power supply falling time.

#### (2) LVD detection voltage of interrupt & reset mode

**($T_A$ = -40 to +105°C, $V_{PDR} \le EV_{DD0} = EV_{DD1} = V_{DD} \le 5.5$ V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Interrupt and reset mode | $V_{LVD5}$ | VPOC2, VPOC1, VPOC0 = 0, 0, 1 Note 1, falling reset voltage: 2.75 V | | 2.68 Note 2 | 2.75 | 2.81 | V |
| | $V_{LVD2}$ | LVIS1, LVIS0 = 1, 0 | Rising release reset voltage | 4.30 | 4.42 | 4.51 | V |
| | | | Falling interrupt voltage | 4.21 | 4.32 | 4.41 | V |
| | $V_{LVD5}$ | VPOC2, VPOC1, VPOC0 = 0, 1, 0 Note 1, falling reset voltage: 2.75 V | | 2.68 Note 2 | 2.75 | 2.81 | V |
| | $V_{LVD1}$ | LVIS1, LVIS0 = 0, 0 | Rising release reset voltage | 4.50 | 4.62 | 4.72 | V |
| | | | Falling interrupt voltage | 4.40 | 4.52 | 4.62 | V |
| | $V_{LVD5}$ | VPOC2, VPOC1, VPOC0 = 0, 1, 1 Note 1, falling reset voltage: 2.75 V | | 2.68 Note 2 | 2.75 | 2.81 | V |
| | $V_{LVD3}$ | LVIS1, LVIS0 = 0, 1 | Rising release reset voltage | 3.13 | 3.22 | 3.29 | V |
| | | | Falling interrupt voltage | 3.07 | 3.15 | 3.22 | V |
| | $V_{LVD0}$ | LVIS1, LVIS0 = 0, 0 | Rising release reset voltage | 4.62 | 4.74 | 4.84 | V |
| | | | Falling interrupt voltage | 4.52 | 4.64 | 4.74 | V |

**Notes 1.** These values indicate setting values of option bytes.

**2.** The minimum value exceeds below the lower limit operation voltage (2.7 V), however, in reset mode, normal operation (same behavior when $V_{DD}$ = 2.7 V) is possible until a reset is effected at the power supply falling time.

## 34.7 Power Supply Voltage Rising Time

**($T_A$ = -40 to +105°C, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Maximum power supply voltage rising slope | $S_{vrmax}$ | 0 V → $V_{DD}$ (VPOC2 = 0 or 1[Note 2]) | | | 50[Note 3] | V/ms |
| Minimum power supply voltage rising slope [Note 1] | $S_{vrmin}$ | 0 V → 2.7 V | 6.5 | | | V/ms |

**Notes 1.** The minimum power supply voltage rising slope is applied only under the following condition.
When the voltage detection (LVD) circuit is not used (VPOC2 = 1) and an external reset circuit is not used or when a reset is not effected until $V_{DD}$ = 2.7 V.

**2.** These values indicate setting values of option bytes.

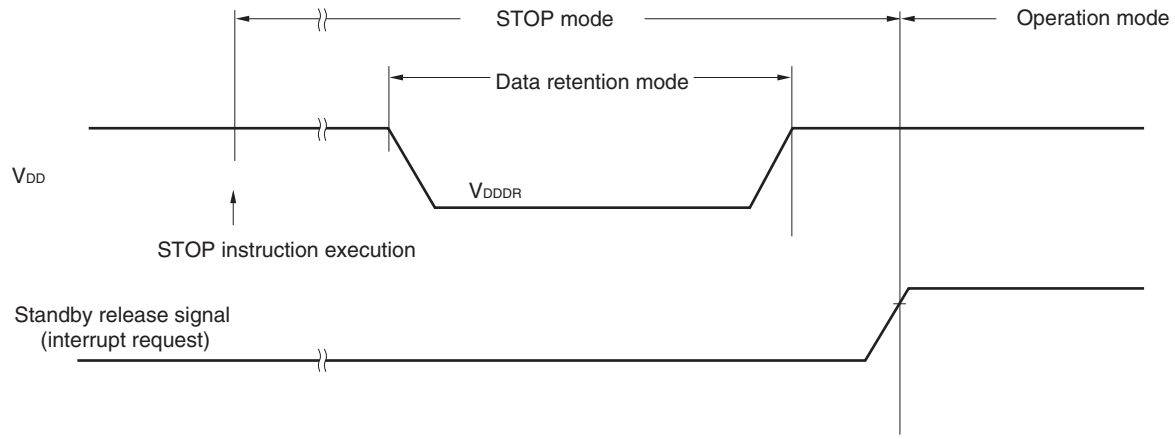
**3.** If the power supply drops below $V_{PDR}$ and a POR reset is effected, this specification is also applied when the power supply is recovered without dropping to 0 V.

## 34.8 STOP Mode Memory Retention Characteristics

**($T_A$ = -40 to +105°C, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Data retention supply voltage | $V_{DDDR}$ | | 1.47[Note] | | 5.5 | V |

**Note** The value depends on the POR detection voltage. When the voltage drops, the data is retained before a POR reset is effected, but data is not retained when a POR reset is effected.



## 34.9 Flash Memory Programming Characteristics

**($T_A$ = -40 to +105°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| System clock frequency | $f_{CLK}$ | | 1 | | 32 | MHz |
| Number of code flash rewrites Notes 1, 2, 3 | $C_{erwr}$ | Retained for 20 years (after rewrite) $T_A$ = +85°C Note 4 | 1,000 | | | Times |
| Number of data flash rewrites Notes 1, 2, 3 | | Retained for 20 years (after rewrite) $T_A$ = +85°C Note 4 | 10,000 | | | |
| | | Retained for 5 years (after rewrite) $T_A$ = +85°C Note 4 | 100,000 | | | |
| <R> Erase time | $T_{erasa}$ | Block erase | 5 | | | ms |
| Write time | $T_{wrwa}$ | 1 word write | 10 | | | μs |

**Notes 1.** 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

**2.** When using flash memory programmer and Renesas Electronics self programming library

**3.** These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

**4.** The specified data retention time is given under the condition that the average temperature ($T_A$) is 85°C or below.

# CHAPTER 35 ELECTRICAL SPECIFICATIONS (GRADE K)

**Cautions 1. RL78/F13 and RL78/F14 have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.**

**2. With products not provided with an $EV_{DD0}$, $EV_{DD1}$, $EV_{SS0}$, or $EV_{SS1}$ pin, replace $EV_{DD0}$ and $EV_{DD1}$ with $V_{DD}$, or replace $EV_{SS0}$ and $EV_{SS1}$ with $V_{SS}$.**

**3. The pins mounted depending on the product. For details, refer to 1.5 Pin Configurations and 2.1 Pin Function List.**

**4. The products are classified into the following five groups according to the product type, pin count, and code flash memory size. In this chapter, the products are referred to by group names depending on the content. In this case, refer to the following classification.**

**Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory**

**Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory**

**Group C: RL78/F13 (CAN and LIN incorporated) products with 30, 32, 48, 64, or 80 pins and 32 Kbytes to 128 Kbytes of code flash memory**

**Group D: RL78/F14 products with 30, 32, 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory**

**Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory**

## 35.1 Absolute Maximum Ratings

(1/2)

| Parameter | Symbol | Conditions | Ratings | Unit |
|---|---|---|---|---|
| Supply voltage | $V_{DD}$ | | -0.5 to +6.5 | V |
| | $EV_{DD0}$, $EV_{DD1}$ | $EV_{DD0} = EV_{DD1}$ | -0.5 to +6.5 | V |
| | $V_{SS}$ | | -0.5 to +0.3 | V |
| | $EV_{SS0}$, $EV_{SS1}$ | $EV_{SS0} = EV_{SS1}$ | -0.5 to +0.3 | V |
| REGC pin input voltage | $V_{IREGC}$ | REGC | -0.3 to +2.8 and -0.3 to $V_{DD}$+0.3 Note 1 | V |
| Input voltage | $V_{I1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97 Note 4, P106, P107, P120, P125 to P127, P140, P150 to P157 | -0.3 to $EV_{DD0}$+0.3 and -0.3 to $V_{DD}$+0.3 Note 2 | V |
| | $V_{I2}$ | P33, P34, P80 to P87, P90 to P97 Note 4, P100 to P105, P121 to P124, P137, $\overline{\text{RESET}}$ | -0.3 to $V_{DD}$+0.3 Note 2 | V |
| Output voltage | $V_{O1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97 Note 4, P106, P107, P120, P125 to P127, P130, P140, P150 to P157 | -0.3 to $EV_{DD0}$+0.3 and -0.3 to $V_{DD}$+0.3 Note 2 | V |
| | $V_{O2}$ | P33, P34, P80 to P87, P90 to P97 Note 4, P100 to P105 | -0.3 to $V_{DD}$+0.3 | V |
| Analog input voltage | $V_{AI1}$ | ANI24 to ANI30 | -0.3 to $EV_{DD0}$+0.3 and -0.3 to $AV_{REF(+)}$+0.3 Notes 2, 3 | V |
| | $V_{AI2}$ | ANI0 to ANI23 | -0.3 to $V_{DD}$+0.3 and -0.3 to $AV_{REF(+)}$+0.3 Notes 2, 3 | V |

**Notes 1.** Connect the REGC pin to $V_{SS}$ via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

**2.** Must be 6.5 V or lower.

**3.** For pins to be used in A/D conversion, the voltage should not exceed the value $AV_{REF(+)}$ + 0.3.

**4.** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(2/2)

| Parameter | Symbol | Conditions | | Ratings | Unit |
|---|---|---|---|---|---|
| Output current, high | $I_{OH1}$ | Per pin | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P130, P140, P150 to P157 | -40 | mA |
| | | Total of all pins -170 mA | P01, P02, P40 to P47, P92 to P97[Note], P120, P125 to P127, P150 to P153 | -70 | mA |
| | | | P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P106, P107, P130, P140, P154 to P157 | -100 | mA |
| | $I_{OH2}$ | Per pin | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105 | -0.5 | mA |
| | | Total of all pins | | -2 | mA |
| Output current, low | $I_{OL1}$ | Per pin | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P130, P140, P150 to P157 | 40 | mA |
| | | Total of all pins 170 mA | P01, P02, P40 to P47, P92 to P97[Note], P120, P125 to P127, P150 to P153 | 70 | mA |
| | | | P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P106, P107, P130, P140, P154 to P157 | 100 | mA |
| | $I_{OL2}$ | Per pin | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105 | 1 | mA |
| | | Total of all pins | | 5 | mA |
| Operating ambient temperature | $T_A$ | In normal operation mode | | -40 to +125 | °C |
| | | In flash memory programming mode | | | |
| Storage temperature | $T_{stg}$ | | | -65 to +150 | °C |

**Note** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

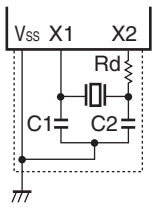
**Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

## 35.2 Oscillator Characteristics

### 35.2.1 Main System Clock Oscillator Characteristics

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Resonator | Recommended Circuit | Parameter | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Ceramic resonator/ Crystal resonator |  | X1 clock oscillation frequency (fx) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | 1.0 | | 20.0 | MHz |

**Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.**

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as $V_{SS}$.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

**2. Customers are requested to consult the resonator manufacturer to select an appropriate resonator and to determine the proper oscillation constant. Customers are also requested to adequately evaluate the oscillation on their system. Determine the X1 clock oscillation stabilization time using the oscillation stabilization time of the oscillation stabilization time counter status register (OSTC) and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.**

### 35.2.2 On-chip Oscillator Characteristics

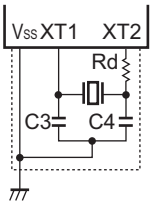
**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Oscillators | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| High-speed on-chip oscillator clock frequency[Note] | $f_{IH}$ | | 1 | | 48 | MHz |
| High-speed on-chip oscillator clock frequency accuracy | - | | -3 | | +3 | % |
| Low-speed on-chip oscillator clock frequency | $f_{IL}$, $f_{WDT}$ | | | 15 | | kHz |
| Low-speed on-chip oscillator clock frequency accuracy | - | | -15 | | +15 | % |

**Note** High-speed on-chip oscillator frequency is selected with bits 0 to 4 of the option byte (000C2H/020C2H) and bits 0 to 2 of the HOCODIV register.

### 35.2.3 Subsystem Clock Oscillator Characteristics

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Resonator | Recommended Circuit | Item | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Crystal resonator |  | XT1 clock oscillation frequency ($f_{XT}$) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | 29.0 | 32.768 | 35.0 | kHz |

**Cautions 1. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.**

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as $V_{SS}$.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

**2. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption and thus required to be adequately evaluated on the system. Customers are requested to consult the resonator manufacturer to select an appropriate resonator and to determine the proper oscillation constant.**

### 35.2.4 PLL Circuit Characteristics

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Resonator | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| PLL input enable clock frequency[Note 1] | $f_{PLLI}$ | PLLMUL = 0 | PLLDIV0 = 0 | 3.92 | 4.0 | 4.08 | MHz |
| | | | PLLDIV0 = 1 | 7.84 | 8.0 | 8.16 | MHz |
| | | PLLMUL = 1 | PLLDIV0 = 0 | 3.92 | 4.0 | 4.08 | MHz |
| | | | PLLDIV0 = 1 | 7.84 | 8.0 | 8.16 | MHz |
| PLL output frequency (center value) | $f_{PLL}$ | PLLMUL = 0 | PLLDIV0 = 0 | $f_{PLLI}$ × 12/2 | | | MHz |
| | | | PLLDIV0 = 1 | $f_{PLLI}$ × 12/4 | | | MHz |
| | | PLLMUL = 1 [Note 4] | PLLDIV0 = 0 [Note 4] | $f_{PLLI}$ × 16/2 | | | MHz |
| | | | PLLDIV0 = 1 | $f_{PLLI}$ × 16/4 | | | MHz |
| Long-term jitter[Notes 2, 3] | $t_{LJ}$ | $f_{PLL}$ = 24 MHz (480 counts) | | -2 | | +2 | ns |
| | | $f_{PLL}$ = 32 MHz (640 counts) | | -2 | | +2 | ns |
| | | $f_{PLL}$ = 48 MHz (960 counts) | | -2 | | +2 | ns |

**Notes 1.** If the high-speed on-chip oscillator clock is to be selected as the PLL input clock, the minimum and maximum values will reflect the range of accuracy of the oscillation frequency by the high-speed on-chip oscillator clock.

**2.** Guaranteed by design, but not tested before shipment.

**3.** Indicates 20 μs.

**4.** Setting of PLLMUL = 1 and PLLDIV0 = 0 is prohibited when $f_{PLLI}$ > 6 MHz.

## 35.3 DC Characteristics

### 35.3.1 Pin Characteristics

For the relationship between the port pins shown in the following tables and the products, refer to **CHAPTER 4 PORT FUNCTIONS**.

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (1/4)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Output current, high[Note 1] | $I_{OH1}$ | Per pin for P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note 3], P106, P107, P120, P125 to P127, P130, P140, P150 to 157 | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -5.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -3.0 | mA |
| | | Per pin for P10, P12, P14, P30,P120,P140 (special slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -0.6 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -0.2 | mA |
| | | Total of P01, P02, P40 to P47, P92 to P97[Note 3], P120, P125 to P127, P150 to P153 (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -20.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -10.0 | mA |
| | | Total of P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P106, P107, P130, P140, P154 to P157 (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -30.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -19.0 | mA |
| | | Total of all pins (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -42.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -29.0 | mA |
| | $I_{OH2}$ | Per pin for P33, P34, P80 to P87, P90 to P97[Note 3], P100 to P105 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | -0.1 | mA |
| | | Total of all pins (for duty factors ≤ 70%[Note 2]) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | -2.0 | mA |

<R> **Notes 1.** Value of current at which the device operation is guaranteed even if the current flows from pins $EV_{DD0}$, $EV_{DD1}$ and $V_{DD}$ to an output pin.

**2.** These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins ($I_{OH}$ × 0.7)/(n × 0.01)

<Example> Where n = 80% and $I_{OH}$ = -10.0 mA

Total output current of pins = (-10.0 × 0.7)/(80 × 0.01) ≈ -8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

**3.** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Caution P10 to P17, P60 to P63, P70 to P72, and P120 do not output high level in N-ch open-drain mode. P10 to P12 and P70 to P72 of the Group A products.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (2/4)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Output current, low[Note 1] | $I_{OL1}$ | Per pin for P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note 3], P106, P107, P120, P125 to P127, P130, P140, P150 to 157 | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 8.5 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 4.0 | mA |
| | | Per pin for P10, P12, P14, P30, P120, P140 (special slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 0.59 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 0.07 | mA |
| | | Total of P01, P02, P40 to P47, P92 to P97[Note 3], P120, P125 to P127, P150 to P153 (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 20.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 15.0 | mA |
| | | Total of P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P106, P107, P130, P140, P154 to P157 (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 45.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 35.0 | mA |
| | | Total of all pins (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 65.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 50.0 | mA |
| | $I_{OL2}$ | Per pin for P33, P34, P80 to P87, P90 to P97[Note 3], P100 to P105 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | 0.4 | mA |
| | | Total of all pins (for duty factors ≤ 70%[Note 2]) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | 5.0 | mA |

<R> **Notes 1.** Value of current at which the device operation is guaranteed even if the current flows to the $EV_{SS0}$, $EV_{SS1}$ and $V_{SS}$ pins from an output pin.

**2.** These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins ($I_{OL}$ × 0.7)/(n × 0.01)

<Example> Where n = 80% and $I_{OL}$ = 10.0 mA

Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≈ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

**3.** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (3/4)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Input voltage, high | $V_{IH1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P106, P107, P120, P125 to P127, P140, P150 to P157 (Schmitt 1 mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0.65 $EV_{DD0}$ | | $EV_{DD0}$ Note 1 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0.7 $EV_{DD0}$ | | $EV_{DD0}$ Note 1 | V |
| | $V_{IH2}$ | P10, P11, P13, P14, P16, P17, P30, P43, P50, P52 to P54, P60 to P63, P70, P71, P73, P75 to P77, P107, P125, P150, P152, P153 (Schmitt 3 mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0.8 $EV_{DD0}$ | | $EV_{DD0}$ Note 1 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0.85 $EV_{DD0}$ | | $EV_{DD0}$ Note 1 | V |
| | $V_{IH3}$ | P10, P11, P13, P14, P16, P17, P30, P54, P62, P63, P70, P71, P73, P125 (TTL mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 2.2 | | $EV_{DD0}$ Note 1 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 2.0 | | $EV_{DD0}$ Note 1 | V |
| | $V_{IH4}$ Note 2 | P33, P34, P80 to P87, P90 to P97, P100 to P105, P137 (fixed to Schmitt 3 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0.8 $V_{DD}$ | | $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0.85 $V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH5}$ | $\overline{\text{RESET}}$ (fixed to Schmitt 1 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0.65 $V_{DD}$ | | $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0.7 $V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH6}$ | P121 to P124, EXCLK, EXCLKS (fixed to Schmitt 2 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0.8 $V_{DD}$ | | $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0.8 $V_{DD}$ | | $V_{DD}$ | V |

**Notes 1.** The maximum value of $V_{IH}$ of the pins P10 to P17, P60 to P63, P70 to P72, and P120 is $EV_{DD0}$, even in N-ch open-drain mode.

**2.** P92 to P96 of the Group A products are fixed to Schmitt 1 mode.
P96 and P97 of the Group B, C, and D products are fixed to Schmitt 1 mode.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (4/4)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Input voltage, low | $V_{IL1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P106, P107, P120, P125 to P127, P140, P150 to P157 (Schmitt 1 mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0 | | 0.35 $EV_{DD0}$ | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0 | | 0.3 $EV_{DD0}$ | V |
| | $V_{IL2}$ | P10, P11, P13, P14, P16, P17, P30, P43, P50, P52 to P54, P60 to P63, P70, P71, P73, P75 to P77, P107, P125, P150, P152, P153 (Schmitt 3 mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0 | | 0.5 $EV_{DD0}$ | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0 | | 0.4 $EV_{DD0}$ | V |
| | $V_{IL3}$ | P10, P11, P13, P14, P16, P17, P30, P54, P62, P63, P70, P71, P73, P125 (TTL mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0 | | 0.8 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0 | | 0.5 | V |
| | $V_{IL4}$ Note | P33, P34, P80 to P87, P90 to P97, P100 to P105, P137 (fixed to Schmitt 3 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0 | | 0.5 $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0 | | 0.4 $V_{DD}$ | V |
| | $V_{IL5}$ | $\overline{\text{RESET}}$ (fixed to Schmitt 1 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0 | | 0.35 $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0 | | 0.3 $V_{DD}$ | V |
| | $V_{IL6}$ | P121 to P124, EXCLK, EXCLKS (fixed to Schmitt 2 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0 | | 0.2 $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0 | | 0.2 $V_{DD}$ | V |

**Note** P92 to P96 of the Group A products are fixed to Schmitt 1 mode.
P96 and P97 of the Group B, C, and D products are fixed to Schmitt 1 mode.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (1/2)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Output voltage, high | $V_{OH1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P130, P140, P150 to P157 (normal slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH1}$ = -5.0 mA | $EV_{DD0}$-0.9 | | | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH1}$ = -3.0 mA | $EV_{DD0}$-0.7 | | | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH1}$ = -1.0 mA | $EV_{DD0}$-0.5 | | | V |
| | $V_{OH2}$ | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OH2}$ = -100 μA | $V_{DD}$-0.5 | | | V |
| | $V_{OH3}$ | P10, P12, P14, P30, P120, P140 (special slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH3}$ = -0.6 mA | $EV_{DD0}$-0.8 | | | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH3}$ = -0.2 mA | $EV_{DD0}$-0.5 | | | V |
| Output voltage, low | $V_{OL1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P130, P140, P150 to P157 (normal slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL1}$ = 8.5 mA | | | 0.7 | V |
| | | | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL1}$ = 4.0 mA | | | 0.4 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL1}$ = 4.0 mA | | | 0.7 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL1}$ = 1.5 mA | | | 0.4 | V |
| | $V_{OL2}$ | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OL2}$ = 400 μA | | | 0.4 | V |
| | $V_{OL3}$ | P10, P12, P14, P30, P120, P140 (special slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL3}$ = 0.6 mA | | | 0.8 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL3}$ = 0.07 mA | | | 0.5 | V |

**Note** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Caution P10 to P17, P60 to P63, P70 to P72, and P120 do not output high level in N-ch open-drain mode.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (2/2)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Input leakage current, high | $I_{LIH1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P140, P150 to P157 | $V_I = EV_{DD0}$ | | | 1 | μA |
| | $I_{LIH2}$ | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105, P137, $\overline{RESET}$ | $V_I = V_{DD}$ | | | 1 | μA |
| | $I_{LIH3}$ | P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS) | $V_I = V_{DD}$ In input port or external clock input | | | 1 | μA |
| | | | $V_I = V_{DD}$ In resonator connection | | | 10 | μA |
| Input leakage current, low | $I_{LIL1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P140, P150 to P157 | $V_I = EV_{SS0}$ | | | -1 | μA |
| | $I_{LIL2}$ | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105, P137, $\overline{RESET}$ | $V_I = V_{SS}$ | | | -1 | μA |
| | $I_{LIL3}$ | P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS) | $V_I = V_{SS}$ In input port or external clock input | | | -1 | μA |
| | | | $V_I = V_{SS}$ In resonator connection | | | -10 | μA |
| On-chip pull-up resistance | $R_U$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97, P100 to P107, P120, P125 to P127, P140, P150 to P157 | $V_I = EV_{SS0}$, in input port | 10 | 20 | 100 | kΩ |

**Note** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Caution P10 to P17, P60 to P63, P70 to P72, and P120 do not output high level in N-ch open-drain mode.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

### 35.3.2 Supply Current Characteristics

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (1/3)**

| Items | Symbol | Conditions | | | | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| Supply current Note 1 | $I_{DD1}$ | Operating mode | Normal operation Note 2 | High-speed on-chip oscillator clock operation | $f_{IH}$ = 48 MHz | $f_{CLK}$ = 24 MHz Notes 3, 4 | | 5.1 | 12.0 | mA |
| | | | | | $f_{IH}$ = 24 MHz | $f_{CLK}$ = $f_{IH}$ Notes 3, 4 | | 4.8 | 11.0 | mA |
| | | | | | $f_{IH}$ = 1 MHz | $f_{CLK}$ = $f_{IH}$ Notes 3, 4 | | 1.0 | 2.5 | mA |
| | | | | Resonator operation | $f_{MX}$ = 20 MHz | $f_{CLK}$ = $f_{MX}$ Notes 3, 5 | | 4.2 | 9.0 | mA |
| | | | | | $f_{MX}$ = 1 MHz | $f_{CLK}$ = $f_{MX}$ Notes 3, 5 | | 0.9 | 2.5 | mA |
| | | | | Resonator operation (PLL operation) (PLL input clock = $f_{MX}$) | $f_{PLL}$ = 48 MHz, $f_{MX}$ = 8 MHz | $f_{CLK}$ = 24 MHz Notes 3, 6 | | 5.0 | 12.0 | mA |
| | | | | | $f_{PLL}$ = 24 MHz, $f_{MX}$ = 8 MHz | $f_{CLK}$ = 24 MHz Notes 3, 6 | | 4.9 | 11.0 | mA |
| | | | | | $f_{PLL}$ = 24 MHz, $f_{MX}$ = 4 MHz | $f_{CLK}$ = 24 MHz Notes 3, 6 | | 4.7 | 11.0 | mA |
| | | | | Subsystem clock operation | $f_{SUB}$ = 32.768 kHz | $f_{CLK}$ = $f_{SUB}$ Note 7 | | | | |
| | | | | | | Groups A to D | | 6.0 | 80.0 | μA |
| | | | | | | Group E | | 6.0 | 120.0 | μA |
| | | | | Low-speed on-chip oscillator clock operation | $f_{IL}$ = 15 kHz | $f_{CLK}$ = $f_{IL}$ Note 8 | | | | |
| | | | | | | Groups A to D | | 3.0 | 70.0 | μA |
| | | | | | | Group E | | 3.0 | 110.0 | μA |

**Notes 1.** Total current flowing into $V_{DD}$ and $EV_{DD0}$, including the input leakage current flowing when the level of the input pin is fixed to $V_{DD}$, $EV_{DD0}$, $V_{SS}$, or $EV_{SS0}$. However, not including the current flowing into the I/O buffer and on-chip pull-up/pull-down resistors.

**2.** Current drawn when all the CPU instructions are executed.

**3.** The values below the MAX. column include the peripheral operation current (except for background operation (BGO)). However, the LVD circuit, A/D converter, D/A converter, and comparator are stopped.

**4.** When high-speed system clock, subsystem clock, PLL clock, and low-speed on-chip oscillator clock are stopped.

**5.** When subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

**6.** When subsystem clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

**7.** When high-speed system clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator are stopped.

**8.** When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped.

**Remarks 1.** $f_{MX}$: High-speed system clock frequency

**2.** $f_{SUB}$: Subsystem clock frequency

**3.** $f_{PLL}$: PLL clock frequency

**4.** $f_{IH}$: High-speed on-chip oscillator clock frequency

**5.** $f_{IL}$: Low-speed on-chip oscillator clock frequency

**6.** $f_{CLK}$: CPU/peripheral hardware clock frequency

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (2/3)**

| Items | Symbol | Conditions | | | | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| Supply current Notes1, 3 | $I_{DD2}$ | HALT mode Note 2 | Normal operation Note 4 | High-speed on-chip oscillator clock operation | $f_{IH}$ = 48 MHz | $f_{CLK}$ = 24 MHz Note 6 | | 0.9 | 8.0 | mA |
| | | | | | $f_{IH}$ = 24 MHz | $f_{CLK}$ = $f_{IH}$ Note 6 | | 0.7 | 7.0 | mA |
| | | | | | $f_{IH}$ = 1 MHz | $f_{CLK}$ = $f_{IH}$ Note 6 | | 0.3 | 1.5 | mA |
| | | | | Resonator operation | $f_{MX}$ = 20 MHz | $f_{CLK}$ = $f_{MX}$ Note 7 | | 0.6 | 6.0 | mA |
| | | | | | $f_{MX}$ = 1 MHz | $f_{CLK}$ = $f_{MX}$ Note 7 | | 0.2 | 1.5 | mA |
| | | | | Resonator operation (PLL operation) (PLL input clock = $f_{MX}$) | $f_{PLL}$ = 48 MHz, $f_{MX}$ = 8 MHz | $f_{CLK}$ = 24 MHz Note 8 | | 0.9 | 8.0 | mA |
| | | | | | $f_{PLL}$ = 24 MHz, $f_{MX}$ = 8 MHz | $f_{CLK}$ = 24 MHz Note 8 | | 0.8 | 7.0 | mA |
| | | | | | $f_{PLL}$ = 24 MHz, $f_{MX}$ = 4 MHz | $f_{CLK}$ = 24 MHz Note 8 | | 0.6 | 7.0 | mA |
| | | | | Subsystem clock operation | $f_{SUB}$ = 32.768 kHz | $f_{CLK}$ = $f_{SUB}$ Note 9 | | | | |
| | | | | | Groups A to D | | | 0.7 | 75.0 | μA |
| | | | | | Group E | | | 0.7 | 115.0 | |
| | | | | Low-speed on-chip oscillator clock operation | $f_{IL}$ = 15 kHz | $f_{CLK}$ = $f_{IL}$ Note 10 | | | | |
| | | | | | Groups A to D | | | 0.7 | 65.0 | μA |
| | | | | | Group E | | | 0.7 | 105.0 | |
| | $I_{DD3}$ | STOP mode Note 5 | | $T_A$ = +25°C | Groups A to D | | | 0.5 | | μA |
| | | | | | Group E | | | 0.5 | | |
| | | | | $T_A$ = +50°C | Groups A to D | | | | 2.5 | |
| | | | | | Group E | | | | 4.5 | |
| | | | | $T_A$ = +70°C | Groups A to D | | | | 4.5 | |
| | | | | | Group E | | | | 8.0 | |
| | | | | $T_A$ = +105°C | Groups A to D | | | | 30.0 | |
| | | | | | Group E | | | | 50.0 | |
| | | | | $T_A$ = +125°C | Groups A to D | | | | 60.0 | |
| | | | | | Group E | | | | 100.0 | |

**Notes 1.** Total current flowing into $V_{DD}$ and $EV_{DD0}$, including the input leakage current flowing when the level of the input pin is fixed to $V_{DD}$, $EV_{DD0}$, $V_{SS}$, or $EV_{SS0}$. However, not including the current flowing into the I/O buffer and on-chip pull-up/pull-down resistors.

**2.** When HALT mode is entered during fetch from the flash memory.

**3.** The values below the MAX. column include the peripheral operation current and STOP leakage current. However, the watchdog timer, LVD circuit, A/D converter, D/A converter, and comparator are stopped

**4.** Current flowing when all the instructions are executed by the CPU.

**5.** When high-speed system clock, subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

**6.** When high-speed system clock, subsystem clock, PLL clock, and low-speed on-chip oscillator clock are stopped.

**7.** When subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

8. When subsystem clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.
9. When high-speed system clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.
10. When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped.

**Remarks 1.** $f_{MX}$: High-speed system clock frequency
**2.** $f_{SUB}$: Subsystem clock frequency
**3.** $f_{PLL}$: PLL clock frequency
**4.** $f_{IH}$: High-speed on-chip oscillator clock frequency
**5.** $f_{IL}$: Low-speed on-chip oscillator clock frequency
**6.** $f_{CLK}$: CPU/peripheral hardware clock frequency

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (3/3)**

| Items | Symbol | Conditions | | | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|---|---|
| Supply current[Notes 1, 2] | $I_{SNOZ}$ | SNOOZE mode | A/D converter operation | During mode transition | | | 1.0 | 1.2 | mA |
| | | | | During conversion | Low-voltage mode $AV_{REFP}$ = $V_{DD}$ = 5.0 V | | 2.1 | 2.5 | mA |
| | | | DTC operation | | | | 4.5 | | mA |

**Notes 1.** Total current flowing into $V_{DD}$ and $EV_{DD0}$, including the input leakage current flowing when the level of the input pin is fixed to $V_{DD}$, $EV_{DD0}$, $V_{SS}$, or $EV_{SS0}$. However, not including the current flowing into the I/O buffer and on-chip pull-up/pull-down resistors.

**2.** The values below the MAX. column include the STOP leakage current.

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Window watchdog timer operating current | $I_{WDT}$ Notes 1, 2 | $f_{IL}$ = 15 kHz | | | 0.22 | | μA |
| A/D converter operating current | $I_{ADC}$ Note 3 | When conversion at maximum speed | Normal mode, $AV_{REFP}$ = $V_{DD}$ = 5.0 V | | 1.3 | 1.7 | mA |
| | | When internal reference voltage is selected Note 5 | | | 75.0 | | μA |
| LVD operating current | $I_{LVD}$ Note 4 | | | | 0.08 | | μA |
| Temperature sensor operating current | $I_{TMPS}$ | | | | 75.0 | | μA |
| D/A converter operating current | $I_{DAC}$ | Per channel | | | 0.8 | 1.5 | mA |
| Comparator operating current | $I_{CMP}$ | | | | 50.0 | | μA |
| BGO operating current | $I_{BGO}$ Note 6 | | | | 2.50 | 12.20 | mA |

**Notes 1.** When the high-speed on-chip oscillator clock and high-speed system clock are stopped.

**2.** Current flowing only to the watchdog timer (including the operation current of the 1.5 kHz on-chip oscillator). The current value is the sum of $I_{DD1}$, $I_{DD2}$, or $I_{DD3}$ and $I_{WDT}$ when the watchdog timer operates in STOP mode.

**3.** Current flowing only to the A/D converter. The current value is the sum of $I_{DD1}$ or $I_{DD2}$ and $I_{ADC}$ when the A/D converter operates in operation mode or HALT mode.

**4.** Current flowing only to the LVD circuit. The current value is the sum of $I_{DD1}$, $I_{DD2}$, or $I_{DD3}$ and $I_{LVD}$ when the LVD circuit operates in operation mode, HALT mode, or STOP mode.

**5.** Operating current that increases when the internal reference voltage is selected. This current flows even when conversion is stopped.

**6.** Current increased by the BGO operation. The current value is the sum of $I_{DD1}$ or $I_{DD2}$ and $I_{BGO}$ when the BGO operates in operation mode or HALT mode.

## 35.4 AC Characteristics

### 35.4.1 Basic Operation

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (1/2)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Instruction cycle (minimum instruction execution time) | $T_{CY}$ | High-speed on-chip oscillator clock operation | | 0.04166 | | 1 | μs |
| | | High-speed system clock operation | | 0.05 | | 1 | μs |
| | | PLL clock operation | | 0.04166 | | 1 | μs |
| | | Subsystem clock operation | | 28.5 | 30.5 | 34.5 | μs |
| | | Low-speed on-chip oscillator clock operation | | | 66.6 | | μs |
| | | In self programming mode | | 0.04166 | | 1 | μs |
| CPU/peripheral hardware clock frequency | $f_{CLK}$ | | | 0.04166 | | 66.6 | μs |
| External system clock frequency | $f_{EX}$ | | | 1.0 | | 20.0 | MHz |
| | $f_{EXS}$ | | | 29 | | 35 | kHz |
| External system clock input high-level width, low-level width | $t_{EXH}$, $t_{EXL}$ | | | 24 | | | ns |
| | $t_{EXHS}$, $t_{EXLS}$ | | | 13.7 | | | μs |
| TI00 to TI07, TI10 to TI17 input high-level width, low-level width | $t_{TIH}$, $t_{TIL}$ | | | $1/f_{MCK}$+10 | | | ns |
| TO00 to TO07, TO10 to TO17 output frequency | $f_{TO}$ | All TO pins, Normal slew rate, C = 30 pF | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 12 | MHz |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 6 | MHz |
| | | TO01, TO06, TO07, TO11, TO13 only, Special slew rate, C = 30 pF | | | | 2 | MHz |
| PCLBUZ0 output frequency | $f_{PCL}$ | Normal slew rate C = 30 pF | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 12 | MHz |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 6 | MHz |
| | | Special slew rate C = 30 pF | | | | 2 | MHz |
| Timer RJ input cycle | $t_C$ | TRJIO0 | | 100 | | | ns |
| Timer RJ input high-level width, low-level width | $t_{WH}$, $t_{WL}$ | TRJIO0 | | 40 | | | ns |
| Interrupt input high-level width, low-level width | $t_{INTH}$, $t_{INTL}$ | INTP0 to INTP13 Note | | 1 | | | μs |
| KR0 to KR7 key interrupt tinput low-level width | $t_{KR}$ | | | 250 | | | ns |
| $\overline{\text{RESET}}$ low-level width | $t_{RSL}$ | | | 10 | | | μs |

<R> **Note** Pins $\overline{\text{RESET}}$, INTP0 to INTP3, INTP12, and INTP13 have noise filters for transient levels lasting less than 100 ns.

**Caution Excluding the error in oscillation frequency accuracy**.

**Remark** $f_{MCK}$: Timer array unit operation clock frequency

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (2/2)**

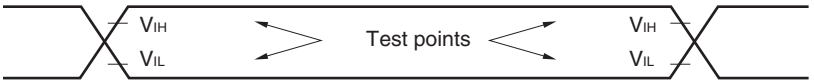
| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Port output rise time, port output fall time | $t_{RO}$, $t_{FO}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P96, P97, P106, P107, P120, P125 to P127, P130, P140, P150 to 157 (normal slew rate) C = 30 pF | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 25 | ns |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 55 | ns |
| | | P10, P12, P14, P30, P120, P140 (special slew rate) C = 30 pF | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | 25 Note | 60 | ns |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 100 | ns |

**Note** $T_A$ = +25°C, $EV_{DD0}$ = 5.0 V

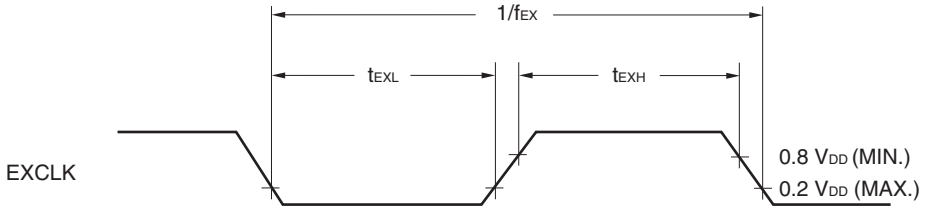
**Caution Excluding the error in oscillation frequency accuracy**.

**Remark** $f_{MCK}$: Timer array unit operation clock frequency

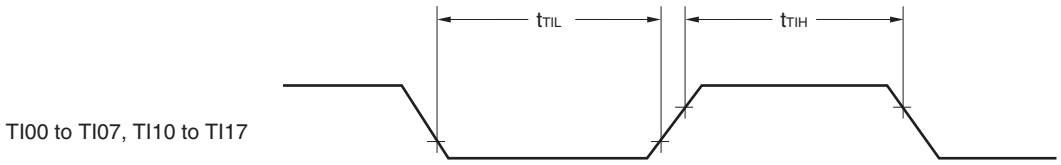
AC Timing Test Points

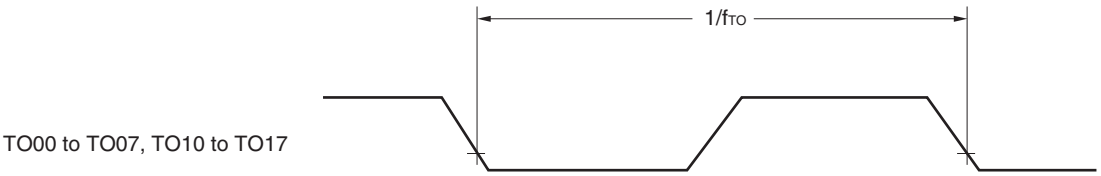



External System Clock Timing

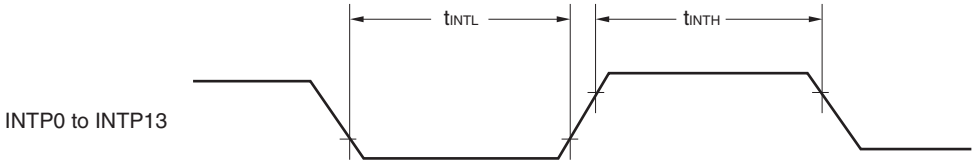


TI/TO Timing

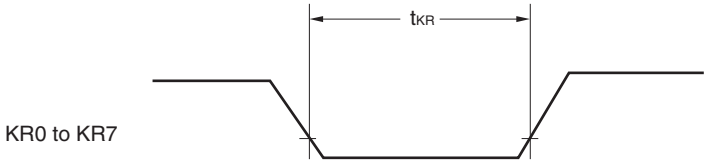


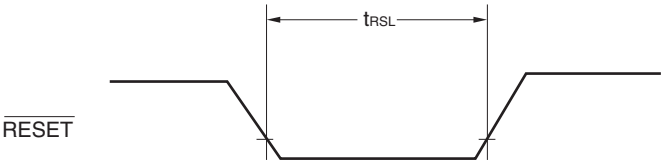

Interrupt Request Input Timing



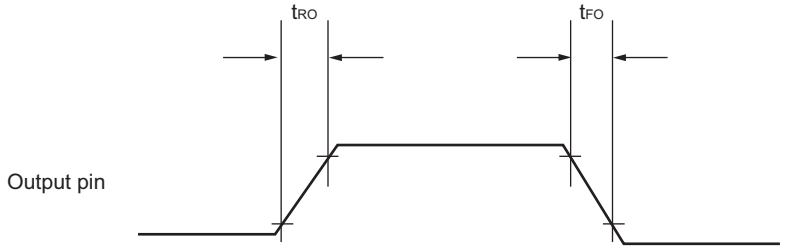

Key Interrupt Input Timing




$\overline{\text{RESET}}$ Input Timing




Output Rising and Falling Timing



## 35.5 Peripheral Functions Characteristics

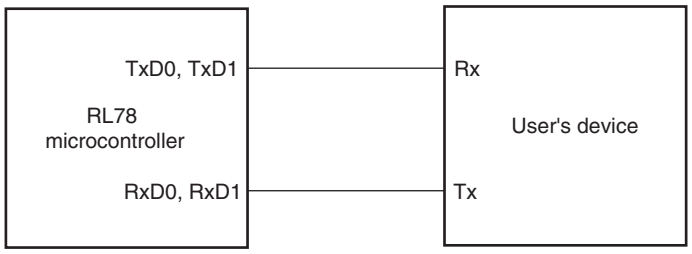
### 35.5.1 Serial Array Unit

**(1) During communication at same potential (UART mode) (dedicated baud rate generator output)**

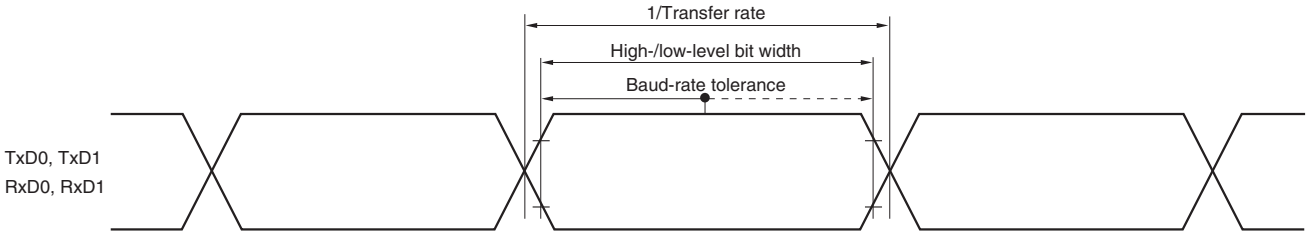
**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Transfer rate | - | | | | | $f_{MCK}$/6 | bps |
| | | $f_{CLK}$ = 24 MHz, $f_{MCK}$ = $f_{CLK}$ | Normal slew rate | | | 4 | Mbps |
| | | | Special slew rate | | | 2 | Mbps |

UART mode connection diagram (during communication at same potential)




UART mode bit width (during communication at same potential) (reference)




**Caution Select the normal input buffer for the RxD0 pin and RxD1 pin and normal output mode for the TxD0 pin and TxD1 pin.**

**Remark** $f_{MCK}$: Serial array unit operation clock frequency

### (2) During communication at same potential (CSI mode) (master mode, $\overline{SCKp}$ … internal clock output, normal slew rate)

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY1}$ | | 166.6 Note 5 | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH1}$, $t_{KL1}$ | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | $t_{KCY1}$/2 – 12 | | | ns |
| | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | $t_{KCY1}$/2 – 18 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑) Note 1 | $t_{SIK1}$ | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 55 | | | ns |
| | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 66 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑) Note 2 | $t_{KSI1}$ | | 30 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output Note 3 | $t_{KSO1}$ | C = 30 pF Note 4 | | | 40 | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**4.** C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

**5.** $t_{KCY1} \geq 4/f_{CLK}$ must also be satisfied.

**Caution Select the normal input buffer for the SIp pin and normal output mode for the SOp pin and $\overline{SCKp}$ pin.**

**Remark** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

### (3) During communication at same potential (CSI mode) (master mode, $\overline{SCKp}$ … internal clock output, special slew rate)

**($T_A$ = -40 to +125°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY1}$ | | 500 Note 5 | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH1}$, $t_{KL1}$ | | $t_{KCY1}$/2 – 60 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑) Note 1 | $t_{SIK1}$ | | 120 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑) Note 2 | $t_{KSI1}$ | | 80 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output Note 3 | $t_{KSO1}$ | C = 30 pF Note 4 | | | 90 | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**4.** C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

**5.** $t_{KCY1} \geq 4/f_{CLK}$ must also be satisfied.

**Caution Select the normal input buffer for the SIp pin and normal output mode and special slew rate for the SOp pin and $\overline{SCKp}$ pin.**

**Remark** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

### (4) During communication at same potential (CSI mode) (slave mode, $\overline{SCKp}$ … external clock input, normal slew rate)

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY2}$ | | | $8/f_{MCK}$ | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH2}$, $t_{KL2}$ | | | $t_{KCY2}/2$ | | | ns |
| SIp setup time (to $\overline{SCKp}$↑)Note 1 | $t_{SIK2}$ | | | $1/f_{MCK}$ + 20 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑)Note 2 | $t_{KSI2}$ | | | $1/f_{MCK}$ + 31 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp outputNote 3 | $t_{KSO2}$ | C = 30 pF Note 4 | 4.0V ≤ $V_{DD}$ = $EV_{DD0}$ = $EV_{DD1}$ ≤ 5.5V | | | $2/f_{MCK}$ + 44 | ns |
| | | | 2.7V ≤ $V_{DD}$ = $EV_{DD0}$ = $EV_{DD1}$ < 4.0V | | | $2/f_{MCK}$ + 57 | ns |
| $\overline{SSIp}$ setup time | $t_{SSIK}$ | DAP = 0 | | 120 | | | ns |
| | | DAP = 1 | | $1/f_{MCK}$ + 120 | | | ns |
| $\overline{SSIp}$ hold time | $t_{KSSI}$ | DAP = 0 | | $1/f_{MCK}$ + 120 | | | ns |
| | | DAP = 1 | | 120 | | | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from $\overline{SCKp}$↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**4.** C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

**Caution Select the normal input buffer for the SIp, $\overline{SCKp}$ and $\overline{SSIp}$ pins and normal output mode for the SOp pin.**

**Remarks 1.** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

**2.** $f_{MCK}$: Serial array unit operation clock frequency

### (5) During communication at same potential (CSI mode) (slave mode, $\overline{SCKp}$ … external clock input, special slew rate)

**($T_A$ = -40 to +125°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY2}$ | 20 MHz < $f_{MCK}$ | 10/$f_{MCK}$ | | | ns |
| | | 10 MHz < $f_{MCK}$ ≤ 20 MHz | 8/$f_{MCK}$ | | | ns |
| | | $f_{MCK}$ ≤ 10 MHz | 6/$f_{MCK}$ | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH2}$, $t_{KL2}$ | | $t_{KCY2}$/2 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑)Note1 | $t_{SIK2}$ | | 1/$f_{MCK}$ + 50 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑)Note 2 | $t_{KSI2}$ | | 1/$f_{MCK}$ + 50 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp outputNote 3 | $t_{KSO2}$ | C = 30 pFNote 4 | | | 2/$f_{MCK}$ + 80 | ns |
| $\overline{SSIp}$ setup time | $t_{SSIK}$ | DAP = 0 | 120 | | | ns |
| | | DAP = 1 | 1/$f_{MCK}$ + 120 | | | ns |
| $\overline{SSIp}$ hold time | $t_{KSSI}$ | DAP = 0 | 1/$f_{MCK}$ + 120 | | | ns |
| | | DAP = 1 | 120 | | | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from $\overline{SCKp}$↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

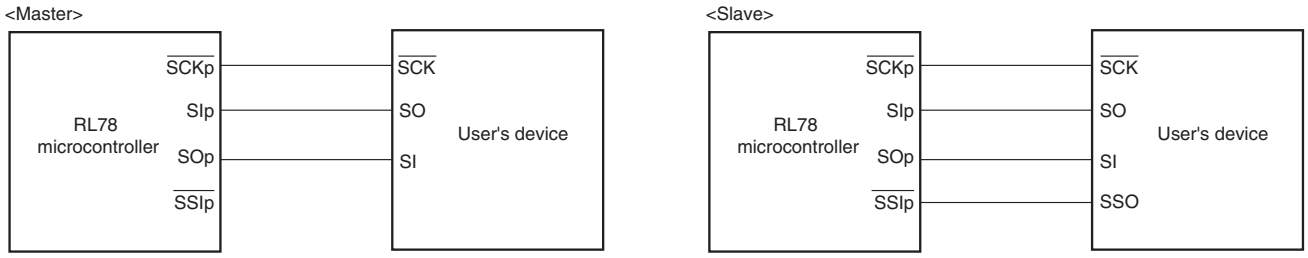
**4.** C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

**Caution Select the normal input buffer for the SIp, $\overline{SCKp}$ and $\overline{SSIp}$ pins and normal output mode and special slew rate for the SOp pin.**

**Remarks 1.** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

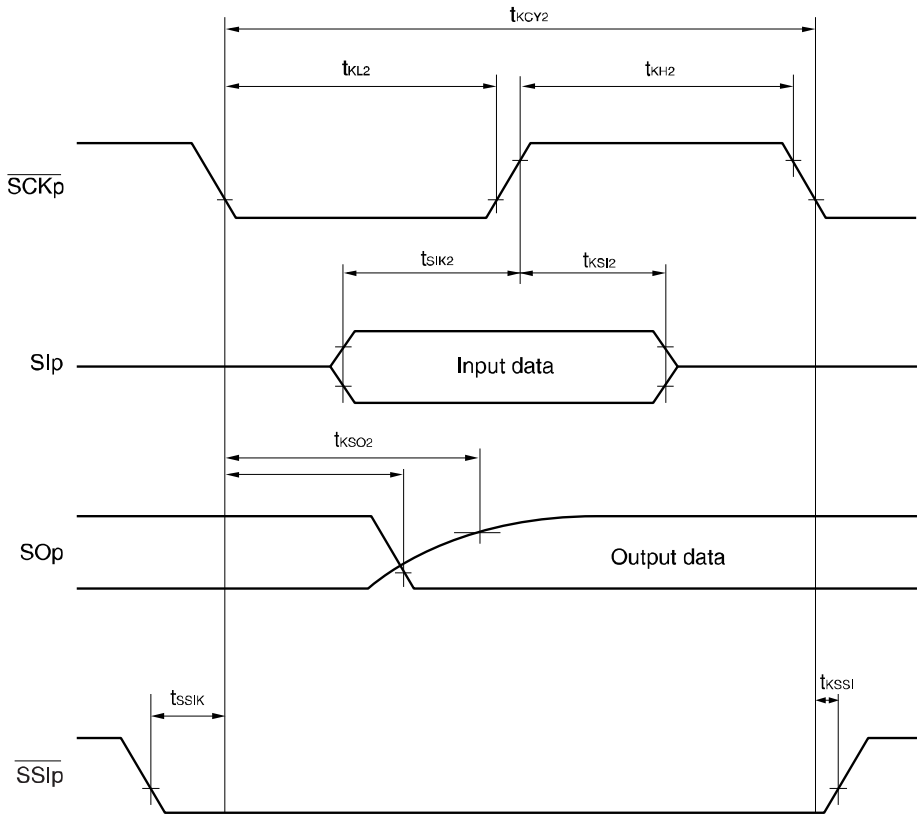
**2.** $f_{MCK}$: Serial array unit operation clock frequency

CSI mode connection diagram (during communication at same potential)



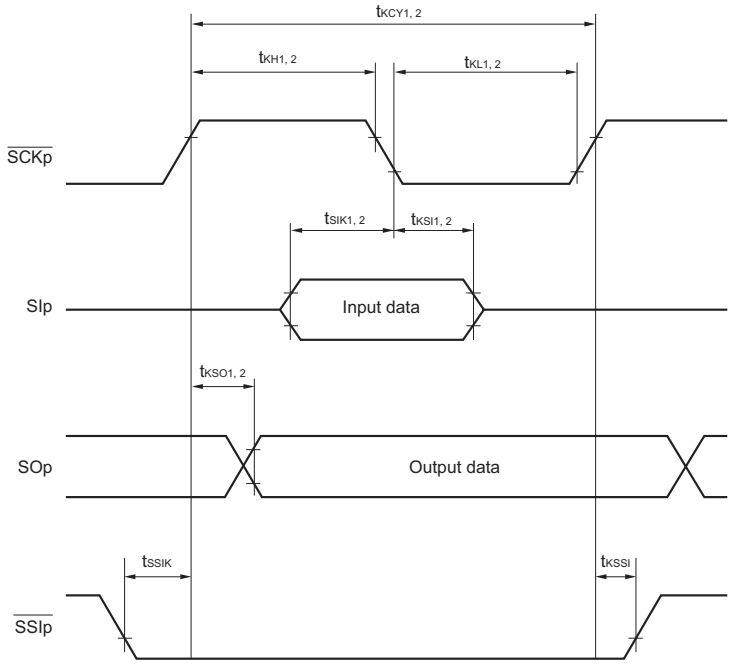

CSI mode serial transfer timing (during communication at same potential)

(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)




**Remark** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

CSI mode serial transfer timing (during communication at same potential)
(When DAPmn= 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)




**Remark** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

### (6) During communication at same potential (simplified I²C mode)

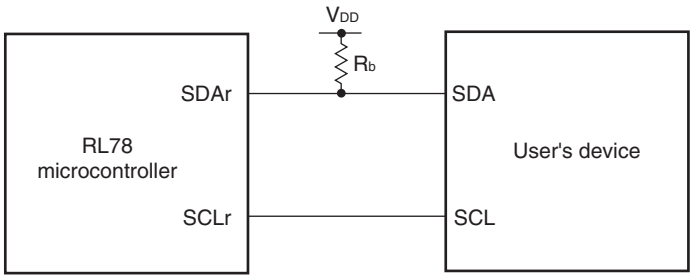
**(SDAr: N-ch open-drain output ($EV_{DD0}$ tolerance) mode, SCLr: normal output mode)**

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

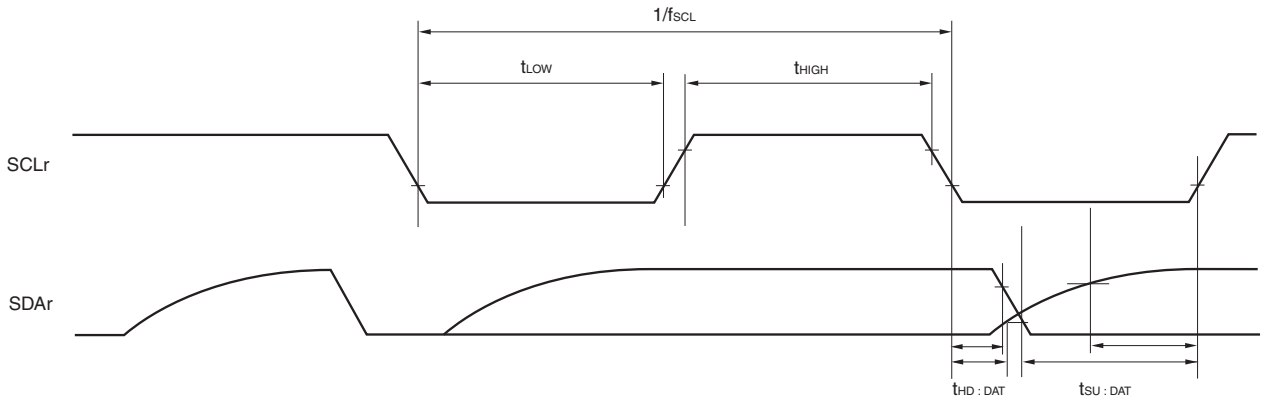
| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| SCLr clock frequency | $f_{SCL}$ | | | | 1000[Note] | kHz |
| Hold time when SCLr = "L" | $t_{LOW}$ | | 475 | | | ns |
| Hold time when SCLr = "H" | $t_{HIGH}$ | | 475 | | | ns |
| Data setup time (reception) | $t_{SU:DAT}$ | | 1/$f_{MCK}$ + 85 | | | ns |
| Data hold time (transmission) | $t_{HD:DAT}$ | $C_b$ = 50 pF, $R_b$ = 2.7 kΩ | 0 | | 305 | ns |

**Note** $f_{CLK} \le f_{MCK}/4$ must also be satisfied.

Simplified I²C mode connection diagram (during communication at same potential)




Simplified I²C mode serial transfer timing (during communication at same potential)




**Caution Select the normal input buffer and N-ch open-drain output mode for the SDAr pin and normal output mode for the SCLr pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (SDAr) pull-up resistance, $C_b$ [F]: Communication line (SCLr, SDAr) load capacitance

**2.** r: IICr (r = 00, 01, 10, 11)

**3.** $f_{MCK}$: Serial array unit operation clock frequency

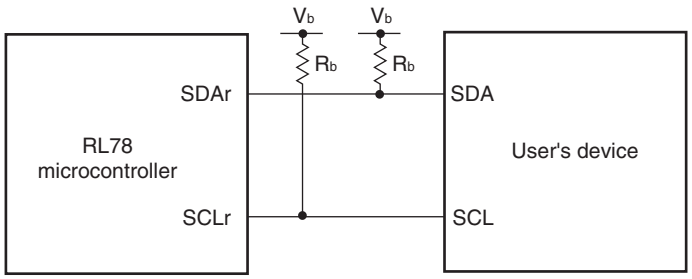
### (7) During communication at same potential (simplified I²C mode) (SDAr and SCLr: N-ch open-drain output (EVDD0 tolerance) mode)

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| SCLr clock frequency | $f_{SCL}$ | | | 400[Note] | kHz |
| Hold time when SCLr = "L" | $t_{LOW}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $C_b$ = 100 pF, $R_b$ = 1.7 kΩ | 1300 | | ns |
| <R> | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $C_b$ = 100 pF, $R_b$ = 2.7 kΩ | | | |
| Hold time when SCLr = "H" | $t_{HIGH}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $C_b$ = 100 pF, $R_b$ = 1.7 kΩ | 600 | | ns |
| <R> | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $C_b$ = 100 pF, $R_b$ = 2.7 kΩ | | | |
| Data setup time (reception) | $t_{SU:DAT}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $C_b$ = 100 pF, $R_b$ = 1.7 kΩ | 1/$f_{MCK}$ + 120 | | ns |
| <R> | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $C_b$ = 100 pF, $R_b$ = 2.7 kΩ | 1/$f_{MCK}$ + 270 | | ns |
| Data hold time (transmission) | $t_{HD:DAT}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $C_b$ = 100 pF, $R_b$ = 1.7 kΩ | 0 | 300 | ns |
| <R> | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $C_b$ = 100 pF, $R_b$ = 2.7 kΩ | | | |

**Note** $f_{CLK}$ ≤ $f_{MCK}$/4 must also be satisfied.

Simplified I²C mode connection diagram (during communication at same potential)



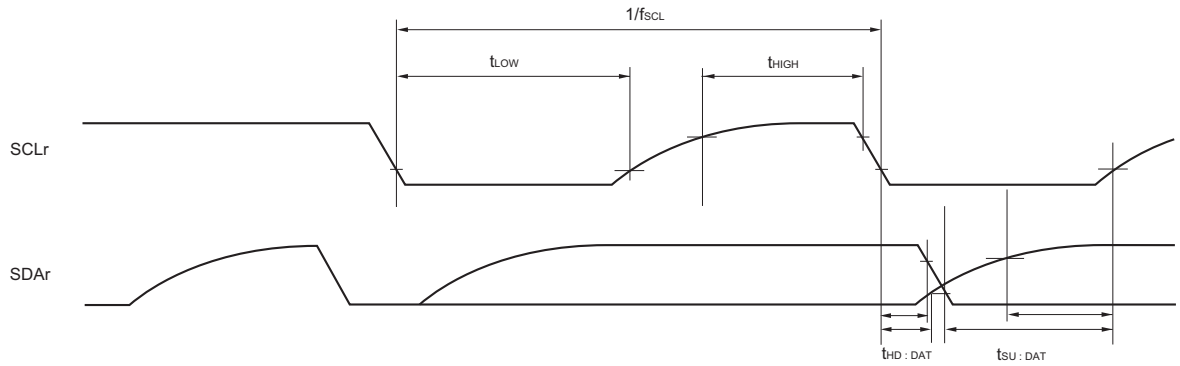

**Caution Select the normal input buffer and N-ch open-drain output mode for the SDAr pin and SCLr pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (SDAr, SCLr) pull-up resistance, $C_b$ [F]: Communication line (SDAr, SCLr) load capacitance, $V_b$[V]: Communication line voltage

**2.** r: IICr (r = 00, 01, 10, 11)

**3.** $f_{MCK}$: Serial array unit operation clock frequency

Simplified $I^2C$ mode serial transfer timing (during communication at same potential)




**Remark** r: IICr (r = 00, 01, 10, 11)

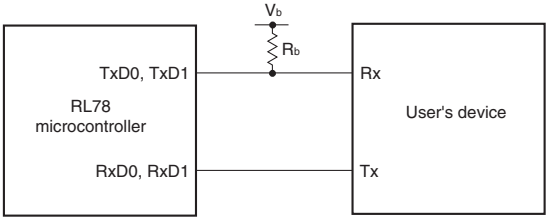
### (8) Communication at different potential (UART mode) (TxD output buffer: N-ch open-drain, RxD input buffer: TTL)

**($T_A$ = -40 to +125°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

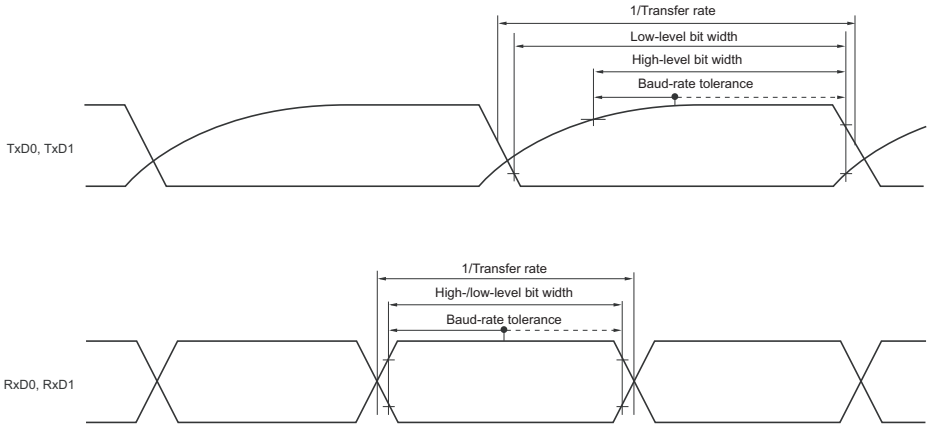
| Parameter | Symbol | Conditions | | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|---|
| Transfer rate | – | Reception | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $V_{IH}$ = 2.2 V, $V_{IL}$ = 0.8 V | | | | $f_{MCK}$/6 | bps |
| | | | | Theoretical value of the maximum transfer rate Note ($C_b$ = 30 pF) | | | 4.0 | Mbps |
| | | Transmission | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $V_{OH}$ = 2.2 V, $V_{OL}$ = 0.8 V | | | | Smaller number of the values given by $f_{MCK}$/6 and expression 1 is applicable. | bps |
| | | | | Theoretical value of the maximum transfer rate Note ($C_b$ = 30 pF) Normal slew rate | | | 4.0 | Mbps |

**Note** Expression 1: Maximum transfer rate = $1 / [\{-C_b \times R_b \times \ln (1 - 2.2/V_b)\} \times 3]$

UART mode connection diagram (during communication at different potential)




UART mode bit width (during communication at different potential) (reference)




**Caution Select the TTL input buffer for the RxD0 pin and RxD1 pin and N-ch open-drain output mode for the TxD0 pin and TxD1 pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (TxD) pull-up resistance, $C_b$ [F]: Communication line (TxD) load capacitance, $V_b$ [V]: Communication line voltage

**2.** $f_{MCK}$: Serial array unit operation clock frequency

### (9) During communication at different potential (3-V supply system) (CSI mode) (master mode, $\overline{SCKp}$ … internal clock output, normal slew rate)

**($T_A$ = -40 to +125°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

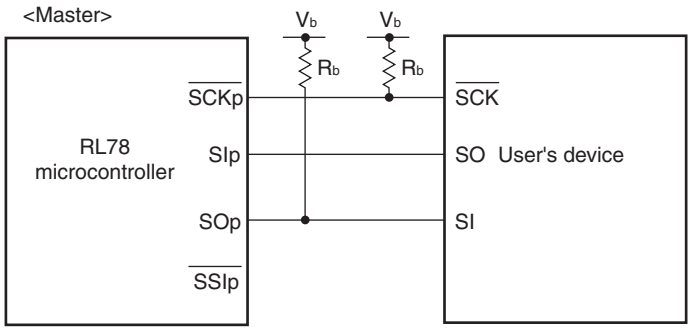
| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 400 Note3 | | | ns |
| $\overline{SCKp}$ high-level width | $t_{KH1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | $t_{KCY1}$/2 – 75 | | | ns |
| $\overline{SCKp}$ low-level width | $t_{KL1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | $t_{KCY1}$/2 – 20 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑) Note 1 | $t_{SIK1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 150 | | | ns |
| SIp setup time (to $\overline{SCKp}$↓) Note 2 | $t_{SIK1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 70 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑) Note 1 | $t_{KSI1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 30 | | | ns |
| SIp hold time (from $\overline{SCKp}$↓) Note 2 | $t_{KSI1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 30 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output Note1 | $t_{KSO1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | | | 120 | ns |
| Delay time from $\overline{SCKp}$↑ to SOp output Note2 | $t_{KSO1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | | | 40 | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

**2.** When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**3.** $t_{KCY1} \geq 4/f_{CLK}$ must also be satisfied.

CSI mode connection diagram (during communication at different potential)




**Caution Select the TTL input buffer for the SIp pin and N-ch open-drain output mode for the SOp pin and $\overline{\text{SCKp}}$ pin.**

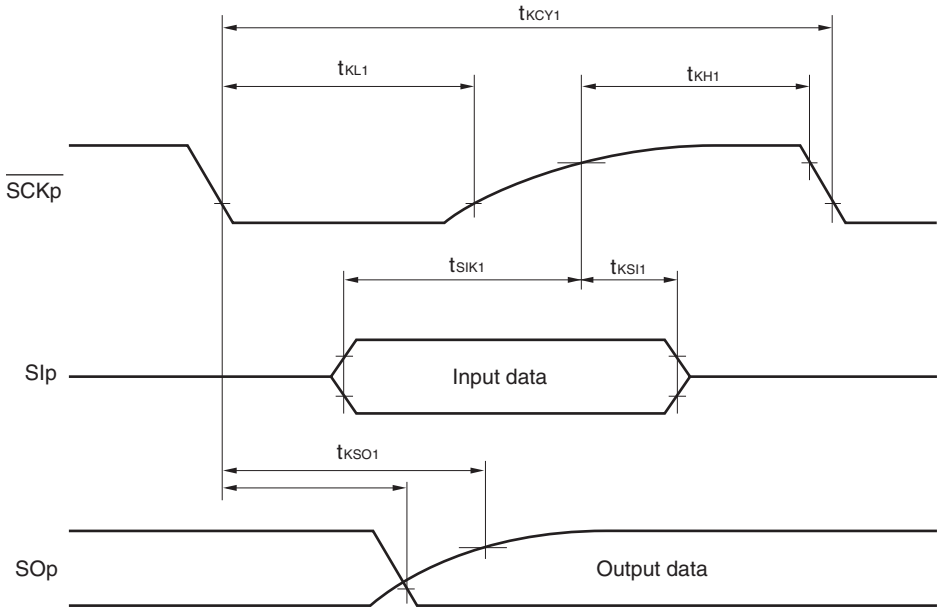
**Remarks 1.** $R_b$ [Ω]: Communication line ($\overline{\text{SCKp}}$, SOp) pull-up resistance, $C_b$ [F]: Communication line (SOp, $\overline{\text{SCKp}}$) load capacitance, $V_b$ [V]: Communication line voltage

**2.** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

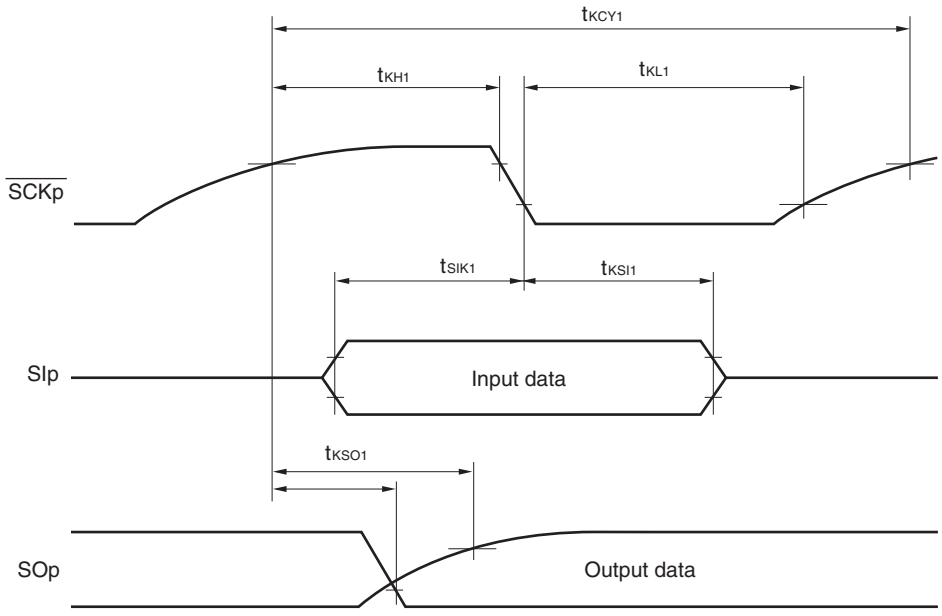
**3.** AC characteristics of the serial array unit during communication at different potential in CSI mode are measured with the $V_{IH}$ and $V_{IL}$ below:

When 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, 2.7 V ≤ $V_b$ ≤ 4.0 V: $V_{IH}$ = 2.2 V, $V_{IL}$ = 0.8 V

CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)




CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn= 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



### (10) During communication at different potential (3-V supply system) (CSI mode) (slave mode, $\overline{SCKp}$ … external clock input, normal slew rate)

**($T_A$ = -40 to +125°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

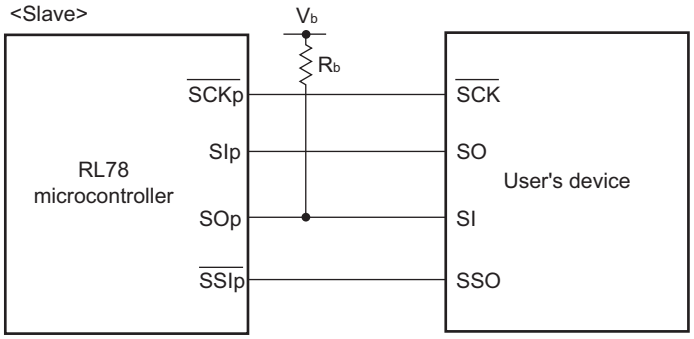
| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY2}$ | 2.7 V ≤ $V_b$ ≤ $V_{DD}$ | 20 MHz < $f_{MCK}$ ≤ 24 MHz | 12/$f_{MCK}$ | | | ns |
| | | | 8 MHz < $f_{MCK}$ ≤ 20 MHz | 10/$f_{MCK}$ | | | ns |
| | | | 4 MHz < $f_{MCK}$ ≤ 8 MHz | 8/$f_{MCK}$ | | | ns |
| | | | $f_{MCK}$ ≤ 4 MHz | 6/$f_{MCK}$ | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH2}$, $t_{KL2}$ | 2.7 V ≤ $V_b$ ≤ $V_{DD}$ | | $t_{KCY2}$/2 – 20 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑)[Note 1] | $t_{SIK2}$ | | | 90 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑)[Note 2] | $t_{KSI2}$ | | | 1/$f_{MCK}$ + 50 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output[Note 3] | $t_{KSO2}$ | 2.7 V ≤ $V_b$ ≤ $V_{DD}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | | | | 2/$f_{MCK}$ + 120 | ns |
| $\overline{SSIp}$ setup time | $t_{SSIK}$ | DAP = 0 | | 120 | | | ns |
| | | DAP = 1 | | 1/$f_{MCK}$ + 120 | | | ns |
| $\overline{SSIp}$ hold time | $t_{KSSI}$ | DAP = 0 | | 1/$f_{MCK}$ + 120 | | | ns |
| | | DAP = 1 | | 120 | | | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from $\overline{SCKp}$↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

CSI mode connection diagram (during communication at different potential)




**Caution Select the TTL input buffer for the SIp, $\overline{\text{SCKp}}$ and $\overline{\text{SSIp}}$ pins and N-ch open-drain output mode for the SOp pin.**

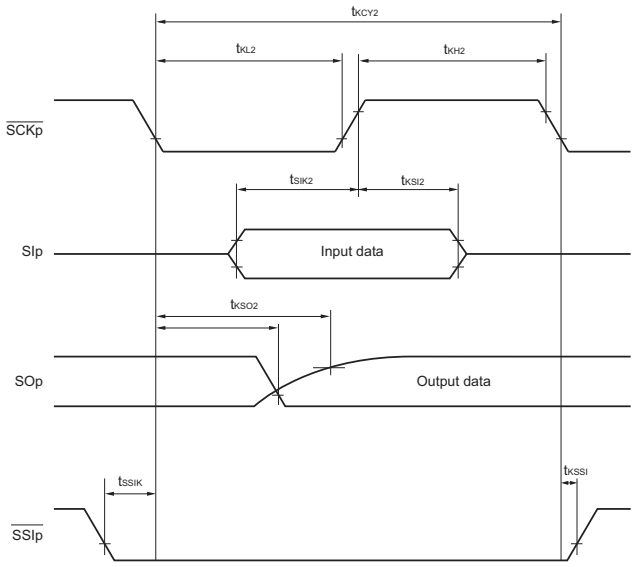
**Remarks 1.** $R_b$ [Ω]: Communication line (SOp) pull-up resistance, $C_b$ [F]: Communication line (SOp) load capacitance, $V_b$ [V]: Communication line voltage

**2.** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

**3.** AC characteristics of the serial array unit during communication at different potential in CSI mode are measured with the $V_{IH}$ and $V_{IL}$ below:
When 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, 2.7 V ≤ $V_b$ ≤ 4.0 V: $V_{IH}$ = 2.2 V, $V_{IL}$ = 0.8 V

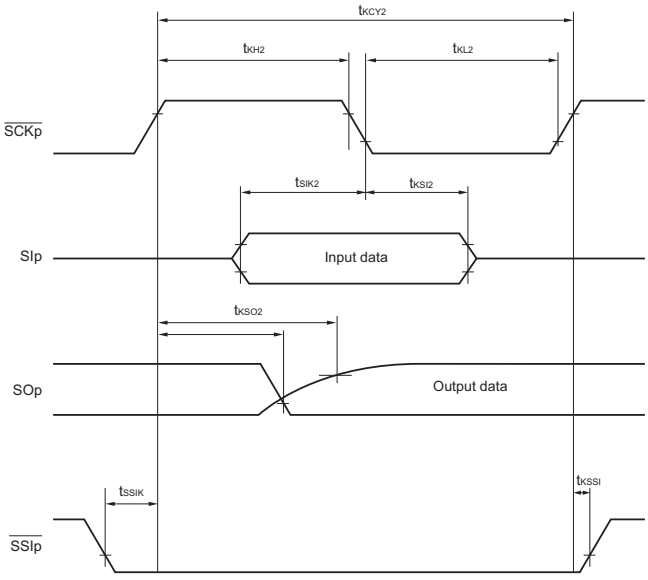
CSI mode serial transfer timing (slave mode) (during communication at different potential)

(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)




CSI mode serial transfer timing (slave mode) (during communication at different potential)

(When DAPmn= 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



**(11) During communication at different potential (3-V supply system) (simplified $I^2C$ mode)**

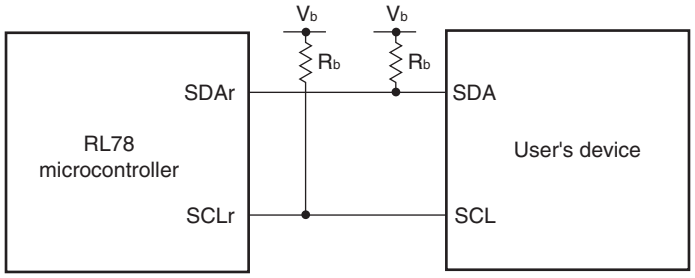
**(SDAr: TTL input buffer mode or N-ch open-drain output ($EV_{DD0}$ tolerance) mode, SCLr: N-ch open-drain output ($EV_{DD0}$ tolerance) mode)**

**($T_A$ = -40 to +125°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

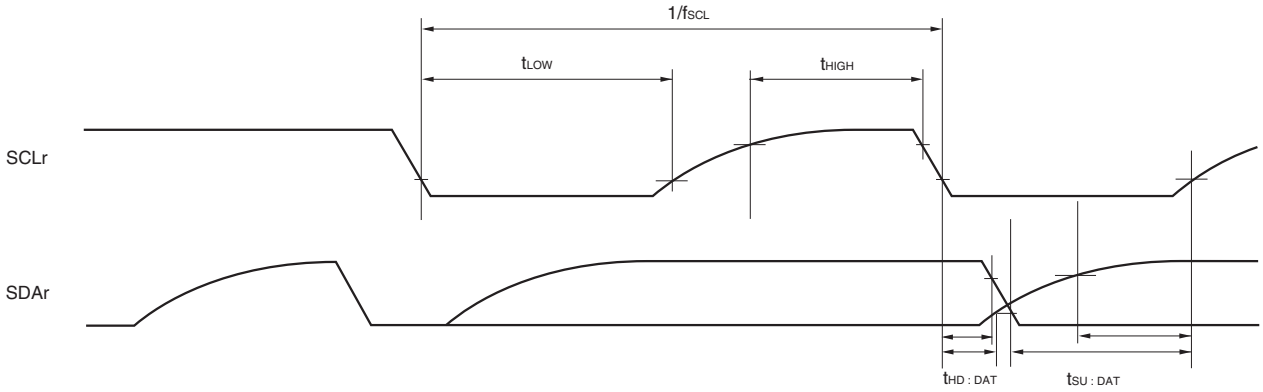
| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| SCLr clock frequency | $f_{SCL}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | | 400[Note] | kHz |
| Hold time when SCLr = "L" | $t_{LOW}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | 1200 | | ns |
| Hold time when SCLr = "H" | $t_{HIGH}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | 600 | | ns |
| Data setup time (reception) | $t_{SU:DAT}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | 135 + 1/$f_{MCK}$ | | ns |
| Data hold time (transmission) | $t_{HD:DAT}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | 0 | 140 | ns |

**Note** $f_{SCL}$ ≤ $f_{MCK}$/4 must also be satisfied.

Simplified I²C mode connection diagram (during communication at different potential)




Simplified I²C mode serial transfer timing (during communication at different potential)




**Caution Select the TTL input buffer and the N-ch open-drain output mode for the SDAr pin and N-ch open-drain output mode for the SCLr pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (SDAr, SCLr) pull-up resistance, $C_b$ [F]: Communication line (SDAr, SCLr) load capacitance, $V_b$ [V]: Communication line voltage

**2.** $f_{MCK}$: Serial array unit operation clock frequency

### 35.5.2 Serial Interface IICA

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | Normal Mode | | Fast Mode | | Fast Mode Plus | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| SCLA0 clock frequency | $f_{SCL}$ | Fast mode plus: 10 MHz ≤ $f_{CLK}$ | | | | | 0 | 1000 | kHz |
| | | Fast mode: 3.5 MHz ≤ $f_{CLK}$ | | | 0 | 400 | | | kHz |
| | | Normal mode: 1 MHz ≤ $f_{CLK}$ | 0 | 100 | | | | | kHz |
| Setup time of restart condition[Note 1] | $t_{SU:STA}$ | | 4.7 | | 0.6 | | 0.26 | | µs |
| Hold time | $t_{HD:STA}$ | | 4.0 | | 0.6 | | 0.26 | | µs |
| Hold time when SCLA0 = "L" | $t_{LOW}$ | | 4.7 | | 1.3 | | 0.5 | | µs |
| Hold time when SCLA0 = "H" | $t_{HIGH}$ | | 4.0 | | 0.6 | | 0.26 | | µs |
| Data setup time (reception) | $t_{SU:DAT}$ | | 250 | | 100 | | 50 | | ns |
| Data hold time (transmission)[Note 2] | $t_{HD:DAT}$ | | 0 | 3.45 | 0 | 0.9 | 0 | | µs |
| Setup time of stop condition | $t_{SU:STO}$ | | 4.0 | | 0.6 | | 0.26 | | µs |
| Bus-free time | $t_{BUF}$ | | 4.7 | | 1.3 | | 0.5 | | µs |

**Notes 1.** The first clock pulse is generated after this period when the start/restart condition is detected.

**2.** The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a wait state is inserted in the $\overline{ACK}$ (acknowledge) timing.

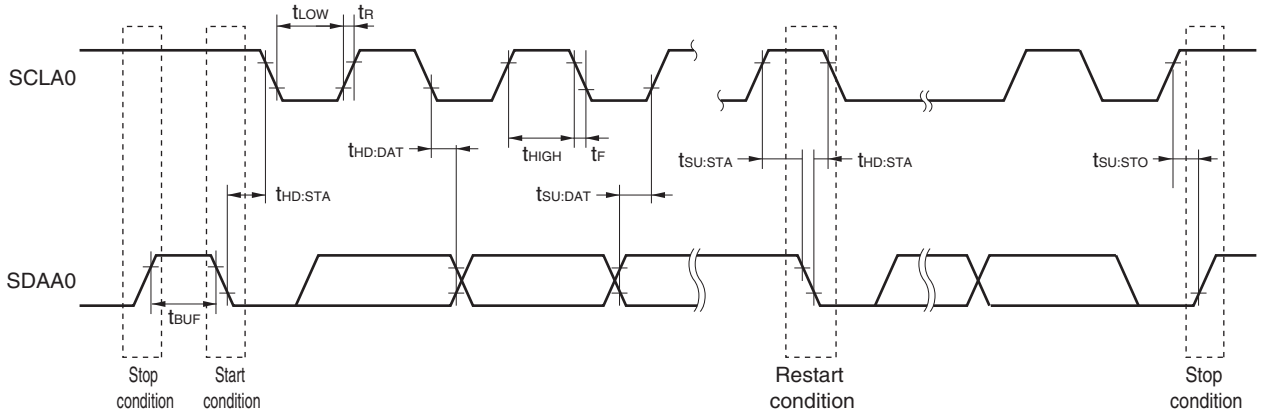
**Remark** The maximum value of $C_b$ (communication line capacitance) and the value of $R_b$ (communication line pull-up resistor) at that time in each mode are as follows.

Standard mode: $C_b$ = 400 pF, $R_b$ = 2.7 kΩ

Fast mode: $C_b$ = 320 pF, $R_b$ = 1.1 kΩ

Fast mode plus: $C_b$ = 120 pF, $R_b$ = 1.1 kΩ

IICA serial transfer timing



### 35.5.3 On-chip Debug (UART)

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Transfer rate | - | | 115.2 k | | 1 M | bps |

### 35.5.4 LIN/UART Module (RLIN3) UART Mode

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Transfer rate | - | Operation mode, HALT mode | LIN communication clock source ($f_{CLK}$ or $f_{MX}$): 4 to 24 MHz | | | 4000 | kbps |
| | | SNOOZE mode | LIN communication clock source ($f_{CLK}$): 1 to 24 MHz FRQSEL4 = 0 in the user option byte (000C2H/020C2H) | | | 4.8 | |
| | | | LIN communication clock source ($f_{CLK}$): 1 to 24 MHz FRQSEL4 = 1 in the user option byte (000C2H/020C2H) | | | 2.4 | |

## 35.6 Analog Characteristics

### 35.6.1 A/D Converter Characteristics

**(1) When $AV_{REF}$ (+) = $AV_{REFP}$/ANI0 (ADREFP1 = 0, ADREFP0 = 1), $AV_{REF}$ (-) = $AV_{REFM}$/ANI1 (ADREFM = 1), target ANI pin: ANI2 to ANI23 (power supply: $V_{DD}$)**

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V, Reference voltage (+) = $AV_{REFP}$, Reference voltage (-) = $AV_{REFM}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | 8 | | 10 | bit |
| Overall error[Note 1] | AINL | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.2 | ±3.0 | LSB |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.2 | ±3.5 | LSB |
| Conversion time | $t_{CONV}$ | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 2.125 | | 39 | μs |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 3.1875 | | 39 | μs |
| Zero-scale error[Notes 1, 2] | EZS | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.25 | %FSR |
| Full-scale error[Notes 1, 2] | EFS | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.25 | %FSR |
| Integral linearity error[Note 1] | ILE | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.5 | LSB |
| Differential linearity error[Note 1] | DLE | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±1.5 | LSB |
| Reference voltage (+) | $AV_{REFP}$ | | | 2.7 | | $V_{DD}$ | V |
| Analog input voltage | $V_{AIN}$ | | | 0 | | $AV_{REFP}$ | V |
| Internal reference voltage (+) | $V_{BGR}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.38 | 1.45 | 1.5 | V |

**Notes 1.** Excludes quantization error (±1/2 LSB).

**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

**(2) When $AV_{REF}$ (+) = $AV_{REFP}$/ANI0 (ADREFP1 = 0, ADREFP0 = 1), $AV_{REF}$ (-) = $AV_{REFM}$/ANI1 (ADREFM = 1), target ANI pin: ANI24 to ANI30 (power supply: $EV_{DD0}$)**

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V, Reference voltage (+) = $AV_{REFP}$, Reference voltage (-) = $AV_{REFM}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | 8 | | 10 | bit |
| Overall error[Note 1] | AINL | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.2 | ±4.5 | LSB |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.2 | ±5.0 | LSB |
| Conversion time | $t_{CONV}$ | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 2.125 | | 39 | μs |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 3.1875 | | 39 | μs |
| Zero-scale error[Notes 1, 2] | EZS | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.35 | %FSR |
| Full-scale error[Notes 1, 2] | EFS | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.35 | %FSR |
| Integral linearity error[Note 1] | ILE | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±3.5 | LSB |
| Differential linearity error[Note 1] | DLE | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.0 | LSB |
| Reference voltage (+) | $AV_{REFP}$ | | | 2.7 | | $V_{DD}$ | V |
| Analog input voltage | $V_{AIN}$ | | | 0 | | $AV_{REFP}$ and $EV_{DD0}$ | V |
| Internal reference voltage (+) | $V_{BGR}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.38 | 1.45 | 1.5 | V |

**Notes 1.** Excludes quantization error (±1/2 LSB).

**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

**(3) When $AV_{REF}$ (+) = $V_{DD}$ (ADREFP1 = 0, ADREFP0 = 0), $AV_{REF}$ (-) = $V_{SS}$ (ADREFM = 0), target ANI pin: ANI0 to ANI23, ANI24 to ANI30**

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V, Reference voltage (+) = $V_{DD}$, Reference voltage (-) = $V_{SS}$)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | 8 | | 10 | bit |
| Overall error[Note 1] | AINL | 10-bit resolution ANI0 to ANI23 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.2 | ±5.0 | LSB |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.2 | ±5.5 | LSB |
| | | 10-bit resolution ANI24 to ANI30 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.2 | ±6.5 | LSB |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.2 | ±7.0 | LSB |
| Conversion time | $t_{CONV}$ | 10-bit resolution | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 2.125 | | 39 | μs |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 3.1875 | | 39 | μs |
| Zero-scale error[Notes 1, 2] | EZS | 10-bit resolution ANI0 to ANI23 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.50 | %FSR |
| | | 10-bit resolution ANI24 to ANI30 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.60 | %FSR |
| Full-scale error[Notes 1, 2] | EFS | 10-bit resolution ANI0 to ANI23 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.50 | %FSR |
| | | 10-bit resolution ANI24 to ANI30 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.60 | %FSR |
| Integral linearity error[Note 1] | ILE | 10-bit resolution ANI0 to ANI23 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±3.5 | LSB |
| | | 10-bit resolution ANI24 to ANI30 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±4.0 | LSB |
| Differential linearity error[Note 1] | DLE | 10-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.0 | LSB |
| Analog input voltage | $V_{AIN}$ | ANI0 to ANI23[Note 3] | | 0 | | $V_{DD}$ | V |
| | | ANI24 to ANI30[Note 3] | | $EV_{SS}$ | | $EV_{DD0}$ | V |
| Internal reference voltage (+) | $V_{BGR}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.38 | 1.45 | 1.5 | V |

**Notes 1.** Excludes quantization error (±1/2 LSB).

**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

**3.** The number of pins depends on the product. For details, refer to **2.1 Pin Function List**.

**(4) When $AV_{REF}$ (+) = internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), $AV_{REF}$ (-) = $AV_{REFM}$/ANI1 (ADREFM = 1), target ANI pin: ANI0 to ANI23, ANI24 to ANI30**

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V, Reference voltage (+) = $V_{BGR}$, Reference voltage (-) = $AV_{REFM}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | | 8 | | bit |
| Conversion time | $t_{CONV}$ | 8-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | 17 | | 39 | μs |
| Zero-scale error[Notes 1, 2] | EZS | 8-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.60 | %FSR |
| Integral linearity error[Note 1] | ILE | 8-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.0 | LSB |
| Differential linearity error[Note 1] | DLE | 8-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±1.0 | LSB |
| Reference voltage (+) | $V_{BGR}$ | | | 1.38 | 1.45 | 1.5 | V |
| Analog input voltage | $V_{AIN}$ | | | 0 | | $V_{BGR}$ | V |

**Notes 1.** Excludes quantization error (±1/2 LSB).

**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

### 35.6.2 Temperatures Sensor Characteristics

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Temperature sensor output voltage | $V_{TMPS25}$ | Setting ADS register = 80H, $T_A$ = +25°C | | 1.1 | | V |
| Reference output voltage | $V_{CONST}$ | Setting ADS register = 81H | 1.38 | 1.45 | 1.5 | V |
| Temperature coefficient | $F_{VTMPS}$ | Temperature sensor that depends on the temperature | | -3.3 | | mV/°C |
| Operation stabilization wait time | $t_{AMP}$ | | 5 | | | μs |

### 35.6.3 D/A Converter Characteristics

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | | | 8 | bit |
| Overall error | AINL | Rload = 4 MΩ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.5 | LSB |
| | | Rload = 8 MΩ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.5 | LSB |
| Settling time | $t_{SET}$ | Cload = 20 pF | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | 3 | μs |

### 35.6.4 Comparator Characteristics

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input offset voltage | $V_{IOCMP}$ | | | ±5 | ±40 | mV |
| Input voltage range | $V_{ICMP}$ | | 0 | | $V_{DD}$ | V |
| Response time | $t_{CR}$, $t_{CF}$ | Input amplitude ±100 mV | | 70 | 200 | ns |
| Stabilization wait time during input channel switching[Note 1] | $t_{WAIT}$ | Input amplitude ±100 mV | 300 | | | ns |
| Operation stabilization wait time[Note 2] | $t_{CMP}$ | 3.3 V ≤ $V_{DD}$ ≤ 5.5 V | 1 | | | μs |
| | | 2.7 V ≤ $V_{DD}$ < 3.3 V | 3 | | | μs |

**Notes 1.** Period of time from when the comparator input channel is switched until the comparator is switched to output

**2.** Period of time from when the comparator operation is enabled (HCMPON bit in CMPCTL is set to 1) until the comparator satisfies the DC/AC characteristics.

### 35.6.5 POR Circuit Characteristics

**($T_A$ = -40 to +125°C, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Detection voltage[Note] | $V_{POR}$ | Power supply rise time | 1.48 | 1.56 | 1.62 | V |
| | $V_{PDR}$ | Power supply fall time | 1.47 | 1.55 | 1.61 | V |
| Minimum pulse width | $T_{PW}$ | | 300 | | | μs |
| Detection delay time | $T_{PD}$ | | | | 350 | μs |

**Note** This indicates the POR circuit characteristics, and normal operation is not guaranteed under the condition of less than lower limit operation voltage (2.7 V).

### 35.6.6 LVD Circuit Characteristics

#### (1) LVD detection voltage of interrupt mode or reset mode

**($T_A$ = -40 to +125°C, $V_{PDR} \le EV_{DD0} = EV_{DD1} = V_{DD} \le 5.5$ V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)**

| Parameter | | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Detection voltage | Supply voltage level | $V_{LVD0}$ | Power supply rise time | 4.62 | 4.74 | 4.94 | V |
| | | | Power supply fall time | 4.52 | 4.64 | 4.84 | V |
| | | $V_{LVD1}$ | Power supply rise time | 4.50 | 4.62 | 4.82 | V |
| | | | Power supply fall time | 4.40 | 4.52 | 4.71 | V |
| | | $V_{LVD2}$ | Power supply rise time | 4.30 | 4.42 | 4.61 | V |
| | | | Power supply fall time | 4.21 | 4.32 | 4.51 | V |
| | | $V_{LVD3}$ | Power supply rise time | 3.13 | 3.22 | 3.39 | V |
| | | | Power supply fall time | 3.07 | 3.15 | 3.31 | V |
| | | $V_{LVD4}$ | Power supply rise time | 2.95 | 3.02 | 3.17 | V |
| | | | Power supply fall time | 2.89 | 2.96 | 3.09 | V |
| | | $V_{LVD5}$ | Power supply rise time | 2.74 | 2.81 | 2.95 | V |
| | | | Power supply fall time | 2.68 Note | 2.75 | 2.88 | V |
| Minimum pulse width | | $t_{LW}$ | | 300 | | | μs |
| Detection delay time | | $t_{LD}$ | | | | 300 | μs |

**Note** The minimum value exceeds below the lower limit operation voltage (2.7 V), however, in reset mode, normal operation (same behavior when $V_{DD}$ = 2.7 V) is possible until a reset is effected at the power supply falling time.

#### (2) LVD detection voltage of interrupt & reset mode

**($T_A$ = -40 to +125°C, $V_{PDR} \le EV_{DD0} = EV_{DD1} = V_{DD} \le 5.5$ V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Interrupt and reset mode | $V_{LVD5}$ | VPOC2, VPOC1, VPOC0 = 0, 0, 1 Note 1, falling reset voltage: 2.75 V | | 2.68 Note 2 | 2.75 | 2.88 | V |
| | $V_{LVD2}$ | LVIS1, LVIS0 = 1, 0 | Rising release reset voltage | 4.30 | 4.42 | 4.61 | V |
| | | | Falling interrupt voltage | 4.21 | 4.32 | 4.51 | V |
| | $V_{LVD5}$ | VPOC2, VPOC1, VPOC0 = 0, 1, 0 Note 1, falling reset voltage: 2.75 V | | 2.68 Note 2 | 2.75 | 2.88 | V |
| | $V_{LVD1}$ | LVIS1, LVIS0 = 0, 0 | Rising release reset voltage | 4.50 | 4.62 | 4.82 | V |
| | | | Falling interrupt voltage | 4.40 | 4.52 | 4.71 | V |
| | $V_{LVD5}$ | VPOC2, VPOC1, VPOC0 = 0, 1, 1 Note 1, falling reset voltage: 2.75 V | | 2.68 Note 2 | 2.75 | 2.88 | V |
| | $V_{LVD3}$ | LVIS1, LVIS0 = 0, 1 | Rising release reset voltage | 3.13 | 3.22 | 3.39 | V |
| | | | Falling interrupt voltage | 3.07 | 3.15 | 3.31 | V |
| | $V_{LVD0}$ | LVIS1, LVIS0 = 0, 0 | Rising release reset voltage | 4.62 | 4.74 | 4.94 | V |
| | | | Falling interrupt voltage | 4.52 | 4.64 | 4.84 | V |

**Notes 1.** These values indicate setting values of option bytes.

**2.** The minimum value exceeds below the lower limit operation voltage (2.7 V), however, in reset mode, normal operation (same behavior when $V_{DD}$ = 2.7 V) is possible until a reset is effected at the power supply falling time.

## 35.7 Power Supply Voltage Rising Time

**($T_A$ = -40 to +125°C, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Maximum power supply voltage rising slope | $S_{vrmax}$ | 0 V → $V_{DD}$ (VPOC2 = 0 or 1[Note 2]) | | | 50[Note 3] | V/ms |
| Minimum power supply voltage rising slope [Note 1] | $S_{vrmin}$ | 0 V → 2.7 V | 6.5 | | | V/ms |

**Notes 1.** The minimum power supply voltage rising slope is applied only under the following condition.
When the voltage detection (LVD) circuit is not used (VPOC2 = 1) and an external reset circuit is not used or when a reset is not effected until $V_{DD}$ = 2.7 V.

**2.** These values indicate setting values of option bytes.

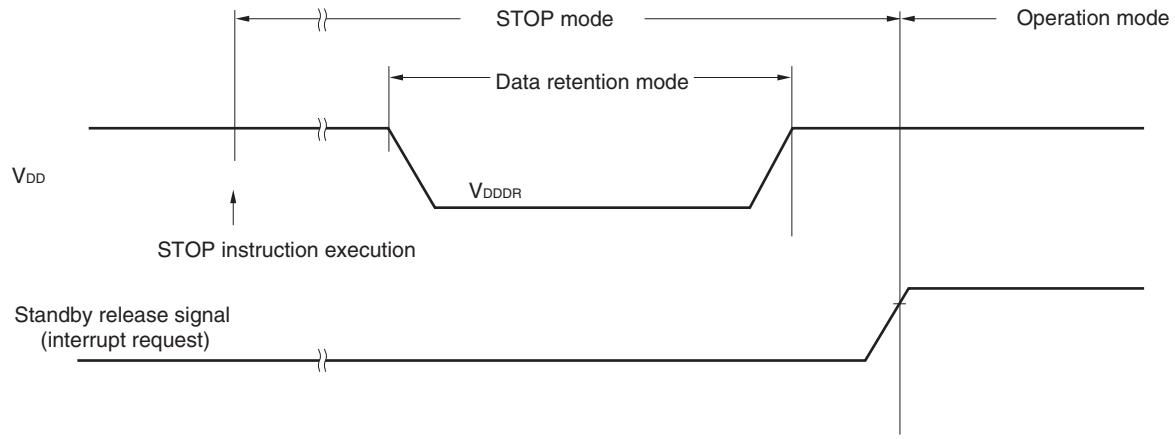
**3.** If the power supply drops below $V_{PDR}$ and a POR reset is effected, this specification is also applied when the power supply is recovered without dropping to 0 V.

## 35.8 STOP Mode Memory Retention Characteristics

**($T_A$ = -40 to +125°C, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Data retention supply voltage | $V_{DDDR}$ | | 1.47[Note] | | 5.5 | V |

**Note** The value depends on the POR detection voltage. When the voltage drops, the data is retained before a POR reset is effected, but data is not retained when a POR reset is effected.



## 35.9 Flash Memory Programming Characteristics

**($T_A$ = -40 to +125°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| System clock frequency | $f_{CLK}$ | | 1 | | 24 | MHz |
| Number of code flash rewrites Notes 1, 2, 3 | $C_{erwr}$ | Retained for 20 years (after rewrite) $T_A$ = +85°C Note 4 | 1,000 | | | Times |
| Number of data flash rewrites Notes 1, 2, 3 | | Retained for 20 years (after rewrite) $T_A$ = +85°C Note 4 | 10,000 | | | |
| | | Retained for 5 years (after rewrite) $T_A$ = +85°C Note 4 | 100,000 | | | |
| <R> Erase time | $T_{erasa}$ | Block erase | 5 | | | ms |
| Write time | $T_{wrwa}$ | 1 word write | 10 | | | μs |

**Notes 1.** 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

**2.** When using flash memory programmer and Renesas Electronics self programming library

**3.** These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

**4.** The specified data retention time is given under the condition that the average temperature ($T_A$) is 85°C or below.

# CHAPTER 36 ELECTRICAL SPECIFICATIONS (GRADE Y)

**Cautions 1. RL78/F13 and RL78/F14 have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.**

**2. With products not provided with an $EV_{DD0}$, $EV_{DD1}$, $EV_{SS0}$, or $EV_{SS1}$ pin, replace $EV_{DD0}$ and $EV_{DD1}$ with $V_{DD}$, or replace $EV_{SS0}$ and $EV_{SS1}$ with $V_{SS}$.**

**3. The pins mounted depending on the product. For details, refer to 1.5 Pin Configurations and 2.1 Pin Function List.**

**4. The products are classified into the following five groups according to the product type, pin count, and code flash memory size. In this chapter, the products are referred to by group names depending on the content. In this case, refer to the following classification.**

**Group A: RL78/F13 (LIN incorporated) products with 20, 30, 32, 48, or 64 pins and 16 Kbytes to 64 Kbytes of code flash memory**

**Group B: RL78/F13 (LIN incorporated) products with 48 or 64 pins and 96 Kbytes to 128 Kbytes of code flash memory or with 80 pins and 64 Kbytes to 128 Kbytes of code flash memory**

**Group C: RL78/F13 (CAN and LIN incorporated) products with 30, 32, 48, 64, or 80 pins and 32 Kbytes to 128 Kbytes of code flash memory**

**Group D: RL78/F14 products with 30, 32, 48, 64, or 80 pins and 48 Kbytes to 96 Kbytes of code flash memory**

**Group E: RL78/F14 products with 48, 64, or 80 pins and 128 Kbytes to 256 Kbytes of code flash memory or with 100 pins and 64 Kbytes to 256 Kbytes of code flash memory**

## 36.1 Absolute Maximum Ratings

(1/2)

| Parameter | Symbol | Conditions | Ratings | Unit |
|---|---|---|---|---|
| Supply voltage | $V_{DD}$ | | -0.5 to +6.5 | V |
| | $EV_{DD0}$, $EV_{DD1}$ | $EV_{DD0} = EV_{DD1}$ | -0.5 to +6.5 | V |
| | $V_{SS}$ | | -0.5 to +0.3 | V |
| | $EV_{SS0}$, $EV_{SS1}$ | $EV_{SS0} = EV_{SS1}$ | -0.5 to +0.3 | V |
| REGC pin input voltage | $V_{IREGC}$ | REGC | -0.3 to +2.8 and -0.3 to $V_{DD}$+0.3 Note 1 | V |
| Input voltage | $V_{I1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97 Note 4, P106, P107, P120, P125 to P127, P140, P150 to P157 | -0.3 to $EV_{DD0}$+0.3 and -0.3 to $V_{DD}$+0.3 Note 2 | V |
| | $V_{I2}$ | P33, P34, P80 to P87, P90 to P97 Note 4, P100 to P105, P121 to P124, P137, $\overline{\text{RESET}}$ | -0.3 to $V_{DD}$+0.3 Note 2 | V |
| Output voltage | $V_{O1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97 Note 4, P106, P107, P120, P125 to P127, P130, P140, P150 to P157 | -0.3 to $EV_{DD0}$+0.3 and -0.3 to $V_{DD}$+0.3 Note 2 | V |
| | $V_{O2}$ | P33, P34, P80 to P87, P90 to P97 Note 4, P100 to P105 | -0.3 to $V_{DD}$+0.3 | V |
| Analog input voltage | $V_{AI1}$ | ANI24 to ANI30 | -0.3 to $EV_{DD0}$+0.3 and -0.3 to $AV_{REF(+)}$+0.3 Notes 2, 3 | V |
| | $V_{AI2}$ | ANI0 to ANI23 | -0.3 to $V_{DD}$+0.3 and -0.3 to $AV_{REF(+)}$+0.3 Notes 2, 3 | V |

**Notes 1.** Connect the REGC pin to $V_{SS}$ via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

**2.** Must be 6.5 V or lower.

**3.** For pins to be used in A/D conversion, the voltage should not exceed the value $AV_{REF(+)}$ + 0.3.

**4.** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(2/2)

| Parameter | Symbol | Conditions | | Ratings | Unit |
|---|---|---|---|---|---|
| Output current, high | $I_{OH1}$ | Per pin | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note 1], P106, P107, P120, P125 to P127, P130, P140, P150 to P157 | -40 | mA |
| | | Total of all pins -170 mA | P01, P02, P40 to P47, P92 to P97[Note 1], P120, P125 to P127, P150 to P153 | -70 | mA |
| | | | P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P106, P107, P130, P140, P154 to P157 | -100 | mA |
| | $I_{OH2}$ | Per pin | P33, P34, P80 to P87, P90 to P97[Note 1], P100 to P105 | -0.5 | mA |
| | | Total of all pins | | -2 | mA |
| Output current, low | $I_{OL1}$ | Per pin | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note 1], P106, P107, P120, P125 to P127, P130, P140, P150 to P157 | 40 | mA |
| <R> | | Total of all pins 170 mA | P01, P02, P40 to P47, P92 to P97[Note 1], P120, P125 to P127, P150 to P153 | 70 | mA |
| <R> | | | P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P106, P107, P130, P140, P154 to P157 | 100 | mA |
| | $I_{OL2}$ | Per pin | P33, P34, P80 to P87, P90 to P97[Note 1], P100 to P105 | 1 | mA |
| | | Total of all pins | | 5 | mA |
| Operating ambient temperature | $T_A$ | In normal operation mode | | -40 to +150 | °C |
| | | In flash memory programming mode | | | |
| Storage temperature | $T_{stg}$ | | | -65 to +150 | °C |

**Note 1.** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

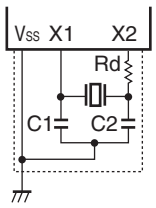
**Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

## 36.2 Oscillator Characteristics

### 36.2.1 Main System Clock Oscillator Characteristics

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Resonator | Recommended Circuit | Parameter | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Ceramic resonator/ Crystal resonator |  | X1 clock oscillation frequency (fx) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | 1.0 | | 20.0 | MHz |

**Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.**

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as $V_{SS}$.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

**2. Customers are requested to consult the resonator manufacturer to select an appropriate resonator and to determine the proper oscillation constant. Customers are also requested to adequately evaluate the oscillation on their system. Determine the X1 clock oscillation stabilization time using the oscillation stabilization time of the oscillation stabilization time counter status register (OSTC) and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.**

### 36.2.2 On-chip Oscillator Characteristics

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Oscillators | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| High-speed on-chip oscillator clock frequency[Note] | $f_{IH}$ | | 1 | | 48 | MHz |
| High-speed on-chip oscillator clock frequency accuracy | - | | -5 | | +5 | % |
| Low-speed on-chip oscillator clock frequency | $f_{IL}$, $f_{WDT}$ | | | 15 | | kHz |
| Low-speed on-chip oscillator clock frequency accuracy | - | | -15 | | +15 | % |

**Note** High-speed on-chip oscillator frequency is selected with bits 0 to 4 of the option byte (000C2H/020C2H) and bits 0 to 2 of the HOCODIV register.

### 36.2.3 Subsystem Clock Oscillator Characteristics

Do not use the XT1 oscillator.

### 36.2.4 PLL Circuit Characteristics

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Resonator | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| PLL input enable clock frequency[Note 1] | $f_{PLLI}$ | PLLMUL = 0 | PLLDIV0 = 0 | 3.92 | 4.0 | 4.08 | MHz |
| | | | PLLDIV0 = 1 | 7.84 | 8.0 | 8.16 | MHz |
| | | PLLMUL = 1 | PLLDIV0 = 0 | 3.92 | 4.0 | 4.08 | MHz |
| | | | PLLDIV0 = 1 | 7.84 | 8.0 | 8.16 | MHz |
| PLL output frequency (center value) | $f_{PLL}$ | PLLMUL = 0 | PLLDIV0 = 0 | $f_{PLLI}$ × 12/2 | | | MHz |
| | | | PLLDIV0 = 1 | $f_{PLLI}$ × 12/4 | | | MHz |
| | | PLLMUL = 1 [Note 4] | PLLDIV0 = 0 [Note 4] | $f_{PLLI}$ × 16/2 | | | MHz |
| | | | PLLDIV0 = 1 | $f_{PLLI}$ × 16/4 | | | MHz |
| Long-term jitter[Notes 2, 3] | $t_{LJ}$ | $f_{PLL}$ = 24 MHz (480 counts) | | -2 | | +2 | ns |
| | | $f_{PLL}$ = 32 MHz (640 counts) | | -2 | | +2 | ns |
| | | $f_{PLL}$ = 48 MHz (960 counts) | | -2 | | +2 | ns |

**Notes 1.** If the high-speed on-chip oscillator clock is to be selected as the PLL input clock, the minimum and maximum values will reflect the range of accuracy of the oscillation frequency by the high-speed on-chip oscillator clock.

**2.** Guaranteed by design, but not tested before shipment.

**3.** Indicates 20 μs.

**4.** Setting of PLLMUL = 1 and PLLDIV0 = 0 is prohibited when $f_{PLLI}$ > 6 MHz.

## 36.3 DC Characteristics

### 36.3.1 Pin Characteristics

For the relationship between the port pins shown in the following tables and the products, refer to **CHAPTER 4 PORT FUNCTIONS**.

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (1/4)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Output current, high[Note 1] | $I_{OH1}$ | Per pin for P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note 3], P106, P107, P120, P125 to P127, P130, P140, P150 to 157 | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -5.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -3.0 | mA |
| | | Per pin for P10, P12, P14, P30,P120,P140 (special slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -0.6 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -0.2 | mA |
| | | Total of P01, P02, P40 to P47, P92 to P97[Note 3], P120, P125 to P127, P150 to P153 (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -20.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -10.0 | mA |
| | | Total of P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P106, P107, P130, P140, P154 to P157 (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -30.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -19.0 | mA |
| | | Total of all pins (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | -32.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | -29.0 | mA |
| | $I_{OH2}$ | Per pin for P33, P34, P80 to P87, P90 to P97[Note 3], P100 to P105 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | -0.1 | mA |
| | | Total of all pins (for duty factors ≤ 70%[Note 2]) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | -2.0 | mA |

<R> **Notes 1.** Value of current at which the device operation is guaranteed even if the current flows from pins $EV_{DD0}$, $EV_{DD1}$ and $V_{DD}$ to an output pin.

**2.** These output current values are obtained under the condition that the duty factor is no greater than 70%.
The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins ($I_{OH}$ × 0.7)/(n × 0.01)

<Example> Where n = 80% and $I_{OH}$ = -10.0 mA

Total output current of pins = (-10.0 × 0.7)/(80 × 0.01) ≈ -8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

**3.** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Caution P10 to P17, P60 to P63, P70 to P72, and P120 do not output high level in N-ch open-drain mode. P10 to P12 and P70 to P72 of the Group A products.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (2/4)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Output current, low[Note 1] | $I_{OL1}$ | Per pin for P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note 3], P106, P107, P120, P125 to P127, P130, P140, P150 to 157 | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 8.5 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 4.0 | mA |
| | | Per pin for P10, P12, P14, P30, P120, P140 (special slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 0.59 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 0.07 | mA |
| | | Total of P01, P02, P40 to P47, P92 to P97[Note 3], P120, P125 to P127, P150 to P153 (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 20.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 15.0 | mA |
| | | Total of P00, P03, P10 to P17, P30 to P32, P50 to P57, P60 to P67, P70 to P77, P106, P107, P130, P140, P154 to P157 (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 35.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 30.0 | mA |
| | | Total of all pins (for duty factors ≤ 70%[Note 2]) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 55.0 | mA |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 45.0 | mA |
| | $I_{OL2}$ | Per pin for P33, P34, P80 to P87, P90 to P97[Note 3], P100 to P105 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | 0.4 | mA |
| | | Total of all pins (for duty factors ≤ 70%[Note 2]) | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | 5.0 | mA |

<R> **Notes 1.** Value of current at which the device operation is guaranteed even if the current flows to the $EV_{SS0}$, $EV_{SS1}$ and $V_{SS}$ pins from an output pin.

**2.** These output current values are obtained under the condition that the duty factor is no greater than 70%. The output current values when the duty factor is changed to a value greater than 70% can be calculated from the following expression (when the duty factor is changed to n%).

- Total output current of pins ($I_{OL}$ × 0.7)/(n × 0.01)

<Example> Where n = 80% and $I_{OL}$ = 10.0 mA

Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≈ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

**3.** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (3/4)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Input voltage, high | $V_{IH1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P106, P107, P120, P125 to P127, P140, P150 to P157 (Schmitt 1 mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0.65 $EV_{DD0}$ | | $EV_{DD0}$ Note 1 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0.7 $EV_{DD0}$ | | $EV_{DD0}$ Note 1 | V |
| | $V_{IH2}$ | P10, P11, P13, P14, P16, P17, P30, P43, P50, P52 to P54, P60 to P63, P70, P71, P73, P75 to P77, P107, P125, P150, P152, P153 (Schmitt 3 mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0.8 $EV_{DD0}$ | | $EV_{DD0}$ Note 1 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0.85 $EV_{DD0}$ | | $EV_{DD0}$ Note 1 | V |
| | $V_{IH3}$ | P10, P11, P13, P14, P16, P17, P30, P54, P62, P63, P70, P71, P73, P125 (TTL mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 2.2 | | $EV_{DD0}$ Note 1 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 2.0 | | $EV_{DD0}$ Note 1 | V |
| | $V_{IH4}$ Note 2 | P33, P34, P80 to P87, P90 to P97, P100 to P105, P137 (fixed to Schmitt 3 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0.8 $V_{DD}$ | | $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0.85 $V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH5}$ | $\overline{\text{RESET}}$ (fixed to Schmitt 1 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0.65 $V_{DD}$ | | $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0.7 $V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH6}$ | P121 to P124, EXCLK, EXCLKS (fixed to Schmitt 2 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0.8 $V_{DD}$ | | $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0.8 $V_{DD}$ | | $V_{DD}$ | V |

**Notes 1.** The maximum value of $V_{IH}$ of the pins P10 to P17, P60 to P63, P70 to P72, and P120 is $EV_{DD0}$, even in N-ch open-drain mode.

**2.** P92 to P96 of the Group A products are fixed to Schmitt 1 mode.
P96 and P97 of the Group B, C, and D products are fixed to Schmitt 1 mode.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (4/4)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Input voltage, low | $V_{IL1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P106, P107, P120, P125 to P127, P140, P150 to P157 (Schmitt 1 mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0 | | 0.35 $EV_{DD0}$ | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0 | | 0.3 $EV_{DD0}$ | V |
| | $V_{IL2}$ | P10, P11, P13, P14, P16, P17, P30, P43, P50, P52 to P54, P60 to P63, P70, P71, P73, P75 to P77, P107, P125, P150, P152, P153 (Schmitt 3 mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0 | | 0.5 $EV_{DD0}$ | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0 | | 0.4 $EV_{DD0}$ | V |
| | $V_{IL3}$ | P10, P11, P13, P14, P16, P17, P30, P54, P62, P63, P70, P71, P73, P125 (TTL mode) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 0 | | 0.8 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 0 | | 0.5 | V |
| | $V_{IL4}$ Note | P33, P34, P80 to P87, P90 to P97, P100 to P105, P137 (fixed to Schmitt 3 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0 | | 0.5 $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0 | | 0.4 $V_{DD}$ | V |
| | $V_{IL5}$ | $\overline{\text{RESET}}$ (fixed to Schmitt 1 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0 | | 0.35 $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0 | | 0.3 $V_{DD}$ | V |
| | $V_{IL6}$ | P121 to P124, EXCLK, EXCLKS (fixed to Schmitt 2 mode) | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 0 | | 0.2 $V_{DD}$ | V |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 0 | | 0.2 $V_{DD}$ | V |

**Note** P92 to P96 of the Group A products are fixed to Schmitt 1 mode.
P96 and P97 of the Group B, C, and D products are fixed to Schmitt 1 mode.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (1/2)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Output voltage, high | $V_{OH1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P130, P140, P150 to P157 (normal slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH1}$ = -5.0 mA | $EV_{DD0}$-0.9 | | | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH1}$ = -3.0 mA | $EV_{DD0}$-0.7 | | | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH1}$ = -1.0 mA | $EV_{DD0}$-0.5 | | | V |
| | $V_{OH2}$ | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OH2}$ = -100 μA | $V_{DD}$-0.5 | | | V |
| | $V_{OH3}$ | P10, P12, P14, P30, P120, P140 (special slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH3}$ = -0.6 mA | $EV_{DD0}$-0.8 | | | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OH3}$ = -0.2 mA | $EV_{DD0}$-0.5 | | | V |
| Output voltage, low | $V_{OL1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P130, P140, P150 to P157 (normal slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL1}$ = 8.5 mA | | | 0.7 | V |
| | | | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL1}$ = 4.0 mA | | | 0.4 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL1}$ = 4.0 mA | | | 0.7 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL1}$ = 1.5 mA | | | 0.4 | V |
| | $V_{OL2}$ | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V, $I_{OL2}$ = 400 μA | | | 0.4 | V |
| | $V_{OL3}$ | P10, P12, P14, P30, P120, P140 (special slew rate) | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL3}$ = 0.6 mA | | | 0.8 | V |
| | | | 2.7 V ≤ $EV_{DD0}$ ≤ 5.5 V, $I_{OL3}$ = 0.07 mA | | | 0.5 | V |

**Note** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Caution P10 to P17, P60 to P63, P70 to P72, and P120 do not output high level in N-ch open-drain mode.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (2/2)**

| Items | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Input leakage current, high | $I_{LIH1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P140, P150 to P157 | $V_I = EV_{DD0}$ | | | 1 | μA |
| | $I_{LIH2}$ | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105, P137, $\overline{RESET}$ | $V_I = V_{DD}$ | | | 1 | μA |
| | $I_{LIH3}$ | P121 to P124 (X1, X2, EXCLK, EXCLKS) | $V_I = V_{DD}$ In input port or external clock input | | | 1 | μA |
| | | | In resonator connection | | | 10 | μA |
| Input leakage current, low | $I_{LIL1}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97[Note], P106, P107, P120, P125 to P127, P140, P150 to P157 | $V_I = EV_{SS0}$ | | | -1 | μA |
| | $I_{LIL2}$ | P33, P34, P80 to P87, P90 to P97[Note], P100 to P105, P137, $\overline{RESET}$ | $V_I = V_{SS}$ | | | -1 | μA |
| | $I_{LIL3}$ | P121 to P124 (X1, X2, EXCLK, EXCLKS) | $V_I = V_{SS}$ In input port or external clock input | | | -1 | μA |
| | | | In resonator connection | | | -10 | μA |
| On-chip pull-up resistance | $R_U$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P92 to P97, P100 to P107, P120, P125 to P127, P140, P150 to P157 | $V_I = EV_{SS0}$, in input port | 10 | 20 | 100 | kΩ |

**Note** For pin I/O buffer power supplies, refer to **Table 4-1 Pin I/O Buffer Power Supplies**.

**Caution P10 to P17, P60 to P63, P70 to P72, and P120 do not output high level in N-ch open-drain mode.**

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

### 36.3.2 Supply Current Characteristics

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (1/3)**

| Items | Symbol | Conditions | | | | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| Supply current Note 1 | $I_{DD1}$ | Operating mode | Normal operation Note 2 | High-speed on-chip oscillator clock operation | $f_{IH}$ = 48 MHz | $f_{CLK}$ = 24 MHz Notes 3, 4 | | 5.1 | 12.5 | mA |
| | | | | | $f_{IH}$ = 24 MHz | $f_{CLK}$ = $f_{IH}$ Notes 3, 4 | | 4.8 | 11.5 | mA |
| | | | | | $f_{IH}$ = 1 MHz | $f_{CLK}$ = $f_{IH}$ Notes 3, 4 | | 1.0 | 2.6 | mA |
| | | | | Resonator operation | $f_{MX}$ = 20 MHz | $f_{CLK}$ = $f_{MX}$ Notes 3, 5 | | 4.2 | 9.5 | mA |
| | | | | | $f_{MX}$ = 1 MHz | $f_{CLK}$ = $f_{MX}$ Notes 3, 5 | | 0.9 | 2.6 | mA |
| | | | | Resonator operation (PLL operation) (PLL input clock = $f_{MX}$) | $f_{PLL}$ = 48 MHz, $f_{MX}$ = 8 MHz | $f_{CLK}$ = 24 MHz Notes 3, 6 | | 5.0 | 12.5 | mA |
| | | | | | $f_{PLL}$ = 24 MHz, $f_{MX}$ = 8 MHz | $f_{CLK}$ = 24 MHz Notes 3, 6 | | 4.9 | 11.5 | mA |
| | | | | | $f_{PLL}$ = 24 MHz, $f_{MX}$ = 4 MHz | $f_{CLK}$ = 24 MHz Notes 3, 6 | | 4.7 | 11.5 | mA |
| | | | | Subsystem clock operation ($f_{SUB}$ = $f_{EXS}$) | $f_{SUB}$ = 32.768 kHz | $f_{CLK}$ = $f_{SUB}$ Note 7 | | | | |
| | | | | | | Groups A to D | | 6.0 | 170.0 | μA |
| | | | | | | Group E | | 6.0 | 270.0 | μA |
| | | | | Low-speed on-chip oscillator clock operation | $f_{IL}$ = 15 kHz | $f_{CLK}$ = $f_{IL}$ Note 8 | | | | |
| | | | | | | Groups A to D | | 3.0 | 160.0 | μA |
| | | | | | | Group E | | 3.0 | 260.0 | μA |

**Notes 1.** Total current flowing into $V_{DD}$ and $EV_{DD0}$, including the input leakage current flowing when the level of the input pin is fixed to $V_{DD}$, $EV_{DD0}$, $V_{SS}$, or $EV_{SS0}$. However, not including the current flowing into the I/O buffer and on-chip pull-up/pull-down resistors.

**2.** Current drawn when all the CPU instructions are executed.

**3.** The values below the MAX. column include the peripheral operation current (except for background operation (BGO)). However, the LVD circuit, A/D converter, D/A converter, and comparator are stopped.

**4.** When high-speed system clock, subsystem clock, PLL clock, and low-speed on-chip oscillator clock are stopped.

**5.** When subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

**6.** When subsystem clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

**7.** When high-speed system clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator are stopped.

**8.** When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped.

**Remarks 1.** $f_{MX}$: High-speed system clock frequency

**2.** $f_{SUB}$: Subsystem clock frequency

**3.** $f_{EXS}$: External subsystem clock frequency

**4.** $f_{PLL}$: PLL clock frequency

**5.** $f_{IH}$: High-speed on-chip oscillator clock frequency

**6.** $f_{IL}$: Low-speed on-chip oscillator clock frequency

**7.** $f_{CLK}$: CPU/peripheral hardware clock frequency

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (2/3)**

| Items | Symbol | Conditions | | | | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| Supply current Notes 1, 3 | $I_{DD2}$ | HALT mode Note 2 | Normal operation Note 4 | High-speed on-chip oscillator clock operation | $f_{IH}$ = 48 MHz | $f_{CLK}$ = 24 MHz Note 6 | | 0.9 | 8.5 | mA |
| | | | | | $f_{IH}$ = 24 MHz | $f_{CLK}$ = $f_{IH}$ Note 6 | | 0.7 | 7.5 | mA |
| | | | | | $f_{IH}$ = 1 MHz | $f_{CLK}$ = $f_{IH}$ Note 6 | | 0.3 | 1.6 | mA |
| | | | | Resonator operation | $f_{MX}$ = 20 MHz | $f_{CLK}$ = $f_{MX}$ Note 7 | | 0.6 | 6.5 | mA |
| | | | | | $f_{MX}$ = 1 MHz | $f_{CLK}$ = $f_{MX}$ Note 7 | | 0.2 | 1.6 | mA |
| | | | | Resonator operation (PLL operation) (PLL input clock = $f_{MX}$) | $f_{PLL}$ = 48 MHz, $f_{MX}$ = 8 MHz | $f_{CLK}$ = 24 MHz Note 8 | | 0.9 | 8.5 | mA |
| | | | | | $f_{PLL}$ = 24 MHz, $f_{MX}$ = 8 MHz | $f_{CLK}$ = 24 MHz Note 8 | | 0.8 | 7.5 | mA |
| | | | | | $f_{PLL}$ = 24 MHz, $f_{MX}$ = 4 MHz | $f_{CLK}$ = 24 MHz Note 8 | | 0.6 | 7.5 | mA |
| | | | | Subsystem clock operation ($f_{SUB}$ = $f_{EXS}$) | $f_{SUB}$ = 32.768 kHz | $f_{CLK}$ = $f_{SUB}$ Note 9 | | | | |
| | | | | | Groups A to D | | | 0.7 | 165.0 | μA |
| | | | | | Group E | | | 0.7 | 265.0 | |
| | | | | Low-speed on-chip oscillator clock operation | $f_{IL}$ = 15 kHz | $f_{CLK}$ = $f_{IL}$ Note 10 | | | | |
| | | | | | Groups A to D | | | 0.7 | 155.0 | μA |
| | | | | | Group E | | | 0.7 | 255.0 | |
| | $I_{DD3}$ | STOP mode Note 5 | | $T_A$ = +25°C | Groups A to D | | | 0.5 | | μA |
| | | | | | Group E | | | 0.5 | | |
| | | | | $T_A$ = +50°C | Groups A to D | | | | 2.5 | |
| | | | | | Group E | | | | 4.5 | |
| | | | | $T_A$ = +70°C | Groups A to D | | | | 4.5 | |
| | | | | | Group E | | | | 8.0 | |
| | | | | $T_A$ = +105°C | Groups A to D | | | | 30.0 | |
| | | | | | Group E | | | | 50.0 | |
| | | | | $T_A$ = +125°C | Groups A to D | | | | 60.0 | |
| | | | | | Group E | | | | 100.0 | |
| | | | | $T_A$ = +150°C | Groups A to D | | | | 150.0 | |
| | | | | | Group E | | | | 250.0 | |

**Notes 1.** Total current flowing into $V_{DD}$ and $EV_{DD0}$, including the input leakage current flowing when the level of the input pin is fixed to $V_{DD}$, $EV_{DD0}$, $V_{SS}$, or $EV_{SS0}$. However, not including the current flowing into the I/O buffer and on-chip pull-up/pull-down resistors.

**2.** When HALT mode is entered during fetch from the flash memory.

**3.** The values below the MAX. column include the peripheral operation current and STOP leakage current. However, the watchdog timer, LVD circuit, A/D converter, D/A converter, and comparator are stopped

**4.** Current flowing when all the instructions are executed by the CPU.

**5.** When high-speed system clock, subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

**6.** When high-speed system clock, subsystem clock, PLL clock, and low-speed on-chip oscillator clock are stopped.

7. When subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.
8. When subsystem clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.
9. When high-speed system clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.
10. When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped.

**Remarks** 1. $f_{MX}$: High-speed system clock frequency
2. $f_{SUB}$: Subsystem clock frequency
3. $f_{EXS}$: External subsystem clock frequency
4. $f_{PLL}$: PLL clock frequency
5. $f_{IH}$: High-speed on-chip oscillator clock frequency
6. $f_{IL}$: Low-speed on-chip oscillator clock frequency
7. $f_{CLK}$: CPU/peripheral hardware clock frequency

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (3/3)**

| Items | Symbol | Conditions | | | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|---|---|
| Supply current[Notes 1, 2] | $I_{SNOZ}$ | SNOOZE mode | A/D comparator operation | During mode transition | | | 1.0 | 1.3 | mA |
| | | | | During conversion | Low-voltage mode $AV_{REFP}$ = $V_{DD}$ = 5.0 V | | 2.1 | 2.6 | mA |
| | | | DTC operation | | | | 4.5 | | mA |

**Notes 1.** Total current flowing into $V_{DD}$ and $EV_{DD0}$, including the input leakage current flowing when the level of the input pin is fixed to $V_{DD}$, $EV_{DD0}$, $V_{SS}$, or $EV_{SS0}$. However, not including the current flowing into the I/O buffer and on-chip pull-up/pull-down resistors.

**2.** The values below the MAX. column include the STOP leakage current.

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Window watchdog timer operating current | $I_{WDT}$ Notes 1, 2 | $f_{IL}$ = 15 kHz | | | 0.22 | | μA |
| A/D converter operating current | $I_{ADC}$ Note 3 | When conversion at maximum speed | Normal mode, $AV_{REFP}$ = $V_{DD}$ = 5.0 V | | 1.3 | 1.7 | mA |
| | | When internal reference voltage is selected Note 5 | | | 75.0 | | μA |
| LVD operating current | $I_{LVD}$ Note 4 | | | | 0.08 | | μA |
| Temperature sensor operating current | $I_{TMPS}$ | | | | 75.0 | | μA |
| D/A converter operating current | $I_{DAC}$ | Per channel | | | 0.8 | 1.5 | mA |
| Comparator operating current | $I_{CMP}$ | | | | 50.0 | | μA |
| BGO operating current | $I_{BGO}$ Note 6 | | | | 2.50 | 12.20 | mA |

**Notes 1.** When the high-speed on-chip oscillator clock and high-speed system clock are stopped.

**2.** Current flowing only to the watchdog timer (including the operation current of the 1.5 kHz on-chip oscillator). The current value is the sum of $I_{DD1}$, $I_{DD2}$, or $I_{DD3}$ and $I_{WDT}$ when the watchdog timer operates in STOP mode.

**3.** Current flowing only to the A/D converter. The current value is the sum of $I_{DD1}$ or $I_{DD2}$ and $I_{ADC}$ when the A/D converter operates in operation mode or HALT mode.

**4.** Current flowing only to the LVD circuit. The current value is the sum of $I_{DD1}$, $I_{DD2}$, or $I_{DD3}$ and $I_{LVD}$ when the LVD circuit operates in operation mode, HALT mode, or STOP mode.

**5.** Operating current that increases when the internal reference voltage is selected. This current flows even when conversion is stopped.

**6.** Current increased by the BGO operation. The current value is the sum of $I_{DD1}$ or $I_{DD2}$ and $I_{BGO}$ when the BGO operates in operation mode or HALT mode.

## 36.4 AC Characteristics

### 36.4.1 Basic Operation

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (1/2)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Instruction cycle (minimum instruction execution time) | $T_{CY}$ | High-speed on-chip oscillator clock operation | | 0.04166 | | 1 | μs |
| | | High-speed system clock operation | | 0.05 | | 1 | μs |
| | | PLL clock operation | | 0.04166 | | 1 | μs |
| | | Subsystem clock operation | | 28.5 | 30.5 | 34.5 | μs |
| | | Low-speed on-chip oscillator clock operation | | | 66.6 | | μs |
| | | In self programming mode | | 0.04166 | | 1 | μs |
| CPU/peripheral hardware clock frequency | $f_{CLK}$ | | | 0.04166 | | 66.6 | μs |
| External system clock frequency | $f_{EX}$ | | | 1.0 | | 20.0 | MHz |
| | $f_{EXS}$ | | | 29 | | 35 | kHz |
| External system clock input high-level width, low-level width | $t_{EXH}$, $t_{EXL}$ | | | 24 | | | ns |
| | $t_{EXHS}$, $t_{EXLS}$ | | | 13.7 | | | μs |
| TI00 to TI07, TI10 to TI17 input high-level width, low-level width | $t_{TIH}$, $t_{TIL}$ | | | $1/f_{MCK}$+10 | | | ns |
| TO00 to TO07, TO10 to TO17 output frequency | $f_{TO}$ | All TO pins, Normal slew rate, C = 30 pF | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 12 | MHz |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 6 | MHz |
| | | TO01, TO06, TO07, TO11, TO13 only, Special slew rate, C = 30 pF | | | | 2 | MHz |
| PCLBUZ0 output frequency | $f_{PCL}$ | Normal slew rate C = 30 pF | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 12 | MHz |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 6 | MHz |
| | | Special slew rate C = 30 pF | | | | 2 | MHz |
| Timer RJ input cycle | $t_C$ | TRJIO0 | | 100 | | | ns |
| Timer RJ input high-level width, low-level width | $t_{WH}$, $t_{WL}$ | TRJIO0 | | 40 | | | ns |
| Interrupt input high-level width, low-level width | $t_{INTH}$, $t_{INTL}$ | INTP1 to INTP13 Note | | 1 | | | μs |
| KR0 to KR7 key interrupt tinput low-level width | $t_{KR}$ | | | 250 | | | ns |
| $\overline{\text{RESET}}$ low-level width | $t_{RSL}$ | | | 10 | | | μs |

<R> **Note** Pins $\overline{\text{RESET}}$, INTP0 to INTP3, INTP12, and INTP13 have noise filters for transient levels lasting less than 100 ns.

**Caution Excluding the error in oscillation frequency accuracy**.

**Remark** $f_{MCK}$: Timer array unit operation clock frequency

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V) (2/2)**

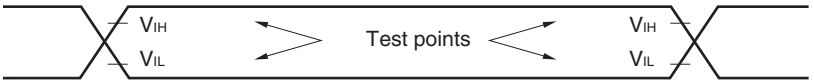
| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Port output rise time, port output fall time | $t_{RO}$, $t_{FO}$ | P00 to P03, P10 to P17, P30 to P32, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P96, P97, P106, P107, P120, P125 to P127, P130, P140, P150 to 157 (normal slew rate) C = 30 pF | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | | 25 | ns |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 55 | ns |
| | | P10, P12, P14, P30, P120, P140 (special slew rate) C = 30 pF | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | | 25 Note | 60 | ns |
| | | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | | | 100 | ns |

**Note** $T_A$ = +25°C, $EV_{DD0}$ = 5.0 V

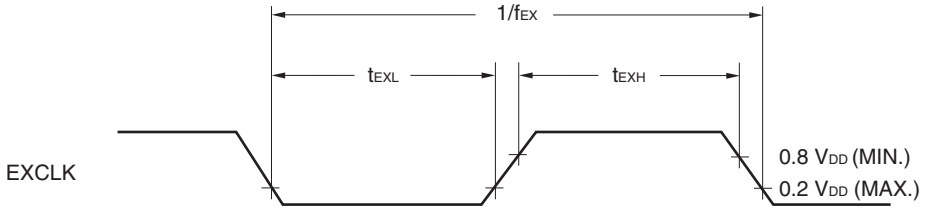
**Caution Excluding the error in oscillation frequency accuracy**.

**Remark** $f_{MCK}$: Timer array unit operation clock frequency

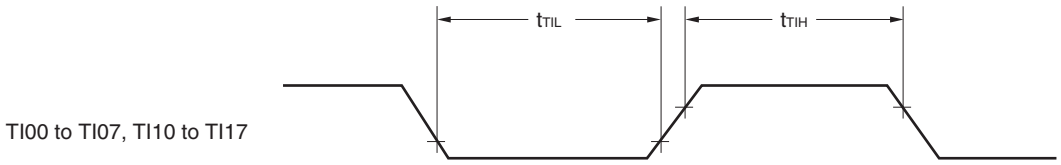
AC Timing Test Points

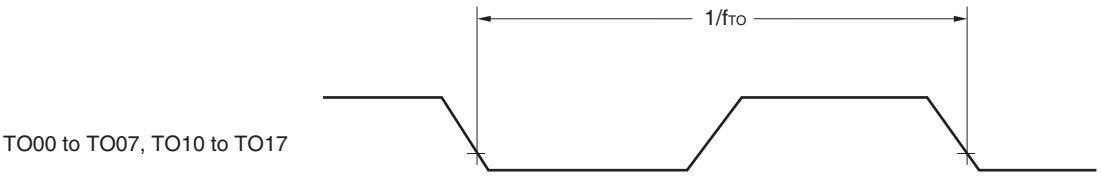



External System Clock Timing

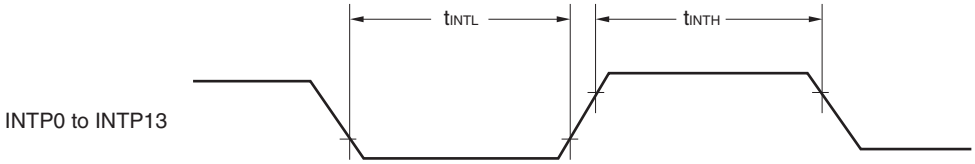


TI/TO Timing

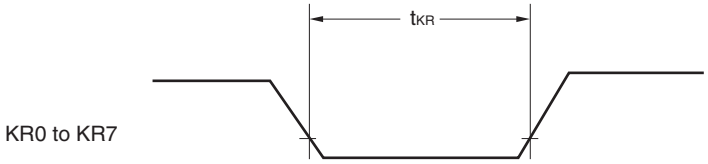


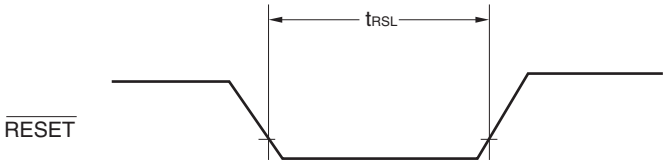

Interrupt Request Input Timing



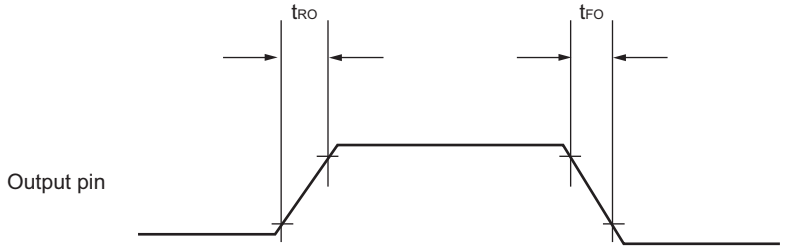

Key Interrupt Input Timing




$\overline{\text{RESET}}$ Input Timing




Output Rising and Falling Timing



## 36.5 Peripheral Functions Characteristics

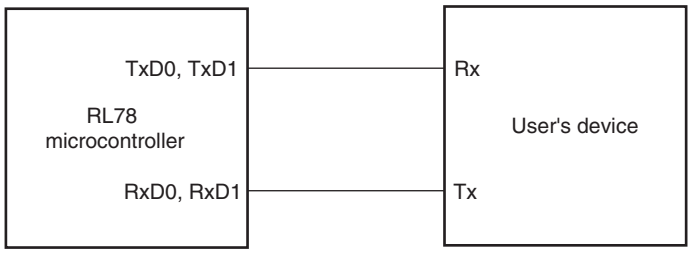
### 36.5.1 Serial Array Unit

**(1) During communication at same potential (UART mode) (dedicated baud rate generator output)**

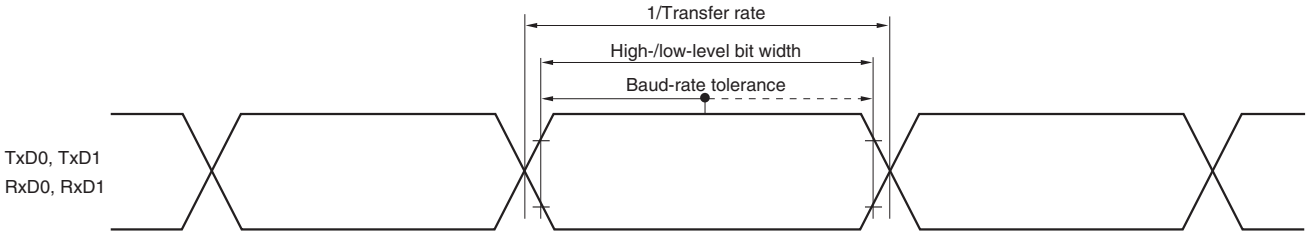
**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Transfer rate | - | | | | | $f_{MCK}$/6 | bps |
| | | $f_{CLK}$ = 24 MHz, $f_{MCK}$ = $f_{CLK}$ | Normal slew rate | | | 4 | Mbps |
| | | | Special slew rate | | | 2 | Mbps |

UART mode connection diagram (during communication at same potential)




UART mode bit width (during communication at same potential) (reference)




**Caution Select the normal input buffer for the RxD0 pin and RxD1 pin and normal output mode for the TxD0 pin and TxD1 pin.**

**Remark** $f_{MCK}$: Serial array unit operation clock frequency

**(2) During communication at same potential (CSI mode) (master mode, $\overline{SCKp}$ … internal clock output, normal slew rate)**

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY1}$ | | 166.6 Note 5 | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH1}$, $t_{KL1}$ | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | $t_{KCY1}$/2 – 12 | | | ns |
| | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | $t_{KCY1}$/2 – 18 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑) Note 1 | $t_{SIK1}$ | 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V | 55 | | | ns |
| | | 2.7 V ≤ $EV_{DD0}$ < 4.0 V | 66 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑) Note 2 | $t_{KSI1}$ | | 30 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output Note 3 | $t_{KSO1}$ | C = 30 pF Note 4 | | | 40 | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**4.** C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

**5.** $t_{KCY1} \geq 4/f_{CLK}$ must also be satisfied.

**Caution Select the normal input buffer for the SIp pin and normal output mode for the SOp pin and $\overline{SCKp}$ pin.**

**Remark** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

### (3) During communication at same potential (CSI mode) (master mode, $\overline{SCKp}$ … internal clock output, special slew rate)

**($T_A$ = -40 to +150°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY1}$ | | 500 Note 5 | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH1}$, $t_{KL1}$ | | $t_{KCY1}$/2 – 60 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑) Note 1 | $t_{SIK1}$ | | 120 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑) Note 2 | $t_{KSI1}$ | | 80 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output Note 3 | $t_{KSO1}$ | C = 30 pF Note 4 | | | 90 | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**4.** C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

**5.** $t_{KCY1} \geq 4/f_{CLK}$ must also be satisfied.

**Caution Select the normal input buffer for the SIp pin and normal output mode and special slew rate for the SOp pin and $\overline{SCKp}$ pin.**

**Remark** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

### (4) During communication at same potential (CSI mode) (slave mode, $\overline{SCKp}$ … external clock input, normal slew rate)

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY2}$ | | | $8/f_{MCK}$ | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH2}$, $t_{KL2}$ | | | $t_{KCY2}/2$ | | | ns |
| SIp setup time (to $\overline{SCKp}$↑)[Note 1] | $t_{SIK2}$ | | | $1/f_{MCK}$ + 20 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑)[Note 2] | $t_{KSI2}$ | | | $1/f_{MCK}$ + 31 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output[Note 3] | $t_{KSO2}$ | C = 30 pF[Note 4] | 4.0V ≤ $V_{DD}$ = $EV_{DD0}$ = $EV_{DD1}$ ≤ 5.5V | | | $2/f_{MCK}$ + 44 | ns |
| | | | 2.7V ≤ $V_{DD}$ = $EV_{DD0}$ = $EV_{DD1}$ < 4.0V | | | $2/f_{MCK}$ + 60 | ns |
| $\overline{SSIp}$ setup time | $t_{SSIK}$ | DAP = 0 | | 120 | | | ns |
| | | DAP = 1 | | $1/f_{MCK}$ + 120 | | | ns |
| $\overline{SSIp}$ hold time | $t_{KSSI}$ | DAP = 0 | | $1/f_{MCK}$ + 120 | | | ns |
| | | DAP = 1 | | 120 | | | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from $\overline{SCKp}$↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**4.** C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

**Caution Select the normal input buffer for the SIp, $\overline{SCKp}$ and $\overline{SSIp}$ pins and normal output mode for the SOp pin.**

**Remarks 1.** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

**2.** $f_{MCK}$: Serial array unit operation clock frequency

### (5) During communication at same potential (CSI mode) (slave mode, $\overline{SCKp}$ … external clock input, special slew rate)

**($T_A$ = -40 to +150°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY2}$ | 20 MHz < $f_{MCK}$ | 10/$f_{MCK}$ | | | ns |
| | | 10 MHz < $f_{MCK}$ ≤ 20 MHz | 8/$f_{MCK}$ | | | ns |
| | | $f_{MCK}$ ≤ 10 MHz | 6/$f_{MCK}$ | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH2}$, $t_{KL2}$ | | $t_{KCY2}$/2 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑)Note1 | $t_{SIK2}$ | | 80 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑)Note 2 | $t_{KSI2}$ | | 1/$f_{MCK}$ + 50 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp outputNote 3 | $t_{KSO2}$ | C = 30 pFNote 4 | | | 2/$f_{MCK}$ + 80 | ns |
| $\overline{SSIp}$ setup time | $t_{SSIK}$ | DAP = 0 | 120 | | | ns |
| | | DAP = 1 | 1/$f_{MCK}$ + 120 | | | ns |
| $\overline{SSIp}$ hold time | $t_{KSSI}$ | DAP = 0 | 1/$f_{MCK}$ + 120 | | | ns |
| | | DAP = 1 | 120 | | | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from $\overline{SCKp}$↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

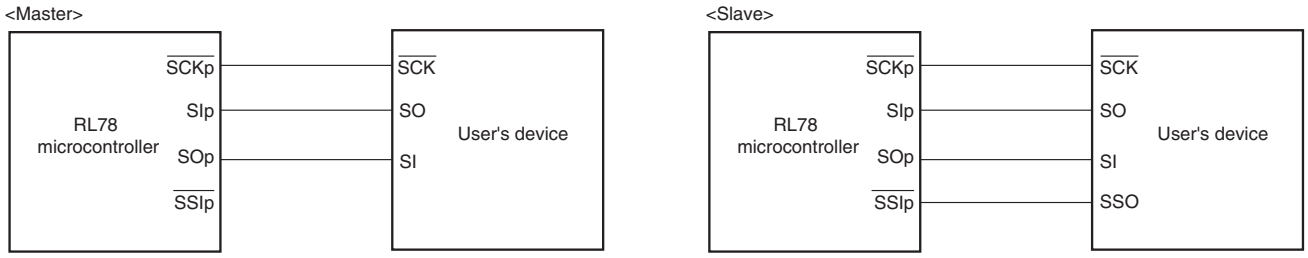
**4.** C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

**Caution Select the normal input buffer for the SIp, $\overline{SCKp}$ and $\overline{SSIp}$ pins and normal output mode and special slew rate for the SOp pin.**

**Remarks 1.** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

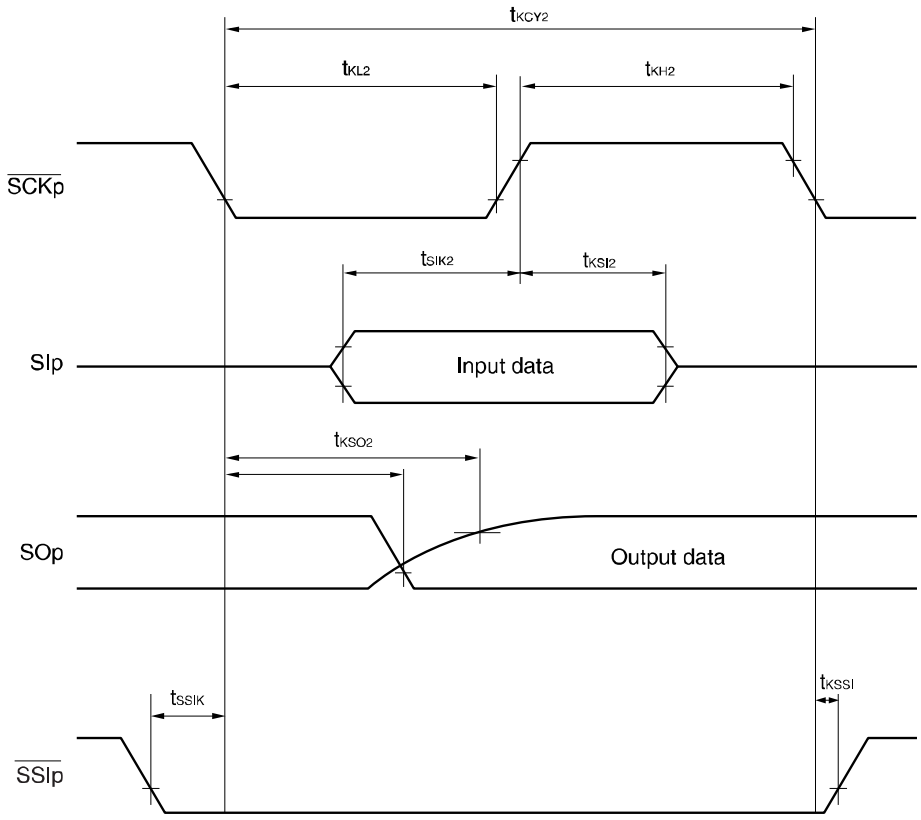
**2.** $f_{MCK}$: Serial array unit operation clock frequency

CSI mode connection diagram (during communication at same potential)



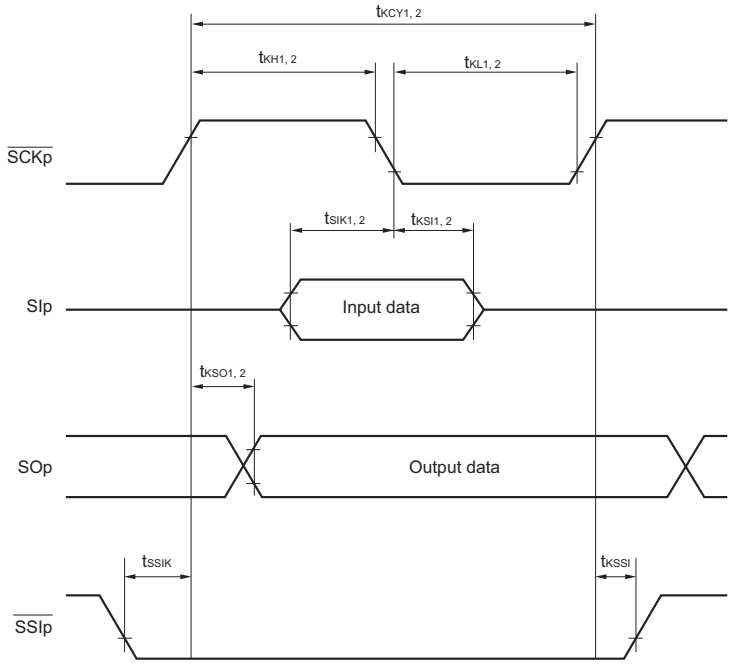

CSI mode serial transfer timing (during communication at same potential)

(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)




**Remark** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

CSI mode serial transfer timing (during communication at same potential)
(When DAPmn= 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)




**Remark** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

### (6) During communication at same potential (simplified I²C mode)

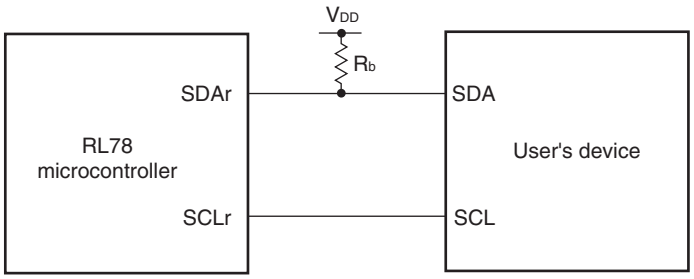
**(SDAr: N-ch open-drain output ($EV_{DD0}$ tolerance) mode, SCLr: normal output mode)**

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

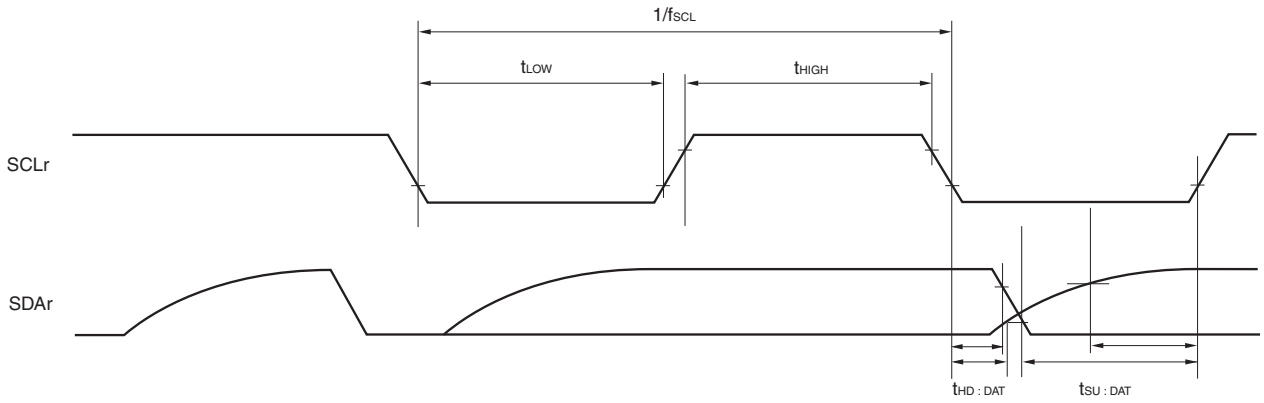
| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| SCLr clock frequency | $f_{SCL}$ | | | | 1000[Note] | kHz |
| Hold time when SCLr = "L" | $t_{LOW}$ | | 475 | | | ns |
| Hold time when SCLr = "H" | $t_{HIGH}$ | | 475 | | | ns |
| Data setup time (reception) | $t_{SU:DAT}$ | | 1/$f_{MCK}$ + 85 | | | ns |
| Data hold time (transmission) | $t_{HD:DAT}$ | $C_b$ = 50 pF, $R_b$ = 2.7 kΩ | 0 | | 305 | ns |

**Note** $f_{CLK} \le f_{MCK}/4$ must also be satisfied.

Simplified I²C mode connection diagram (during communication at same potential)




Simplified I²C mode serial transfer timing (during communication at same potential)




**Caution Select the normal input buffer and N-ch open-drain output mode for the SDAr pin and normal output mode for the SCLr pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (SDAr) pull-up resistance, $C_b$ [F]: Communication line (SCLr, SDAr) load capacitance

**2.** r: IICr (r = 00, 01, 10, 11)

**3.** $f_{MCK}$: Serial array unit operation clock frequency

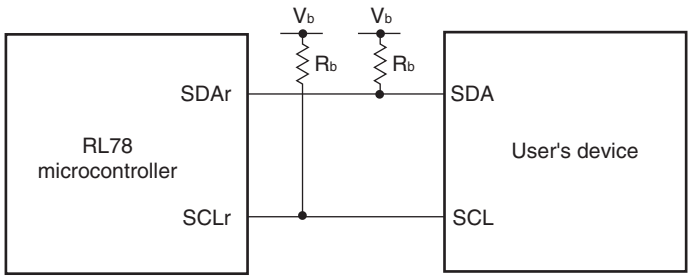
### (7) During communication at same potential (simplified $I^2C$ mode) (SDAr and SCLr: N-ch open-drain output ($EV_{DD0}$ tolerance) mode)

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| SCLr clock frequency | $f_{SCL}$ | | | 400[Note] | kHz |
| Hold time when SCLr = "L" | $t_{LOW}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $C_b$ = 100 pF, $R_b$ = 1.7 kΩ | 1300 | | ns |
| <R> | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $C_b$ = 100 pF, $R_b$ = 2.7 kΩ | | | |
| Hold time when SCLr = "H" | $t_{HIGH}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $C_b$ = 100 pF, $R_b$ = 1.7 kΩ | 600 | | ns |
| <R> | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $C_b$ = 100 pF, $R_b$ = 2.7 kΩ | | | |
| Data setup time (reception) | $t_{SU:DAT}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $C_b$ = 100 pF, $R_b$ = 1.7 kΩ | 1/$f_{MCK}$ + 120 | | ns |
| <R> | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $C_b$ = 100 pF, $R_b$ = 2.7 kΩ | 1/$f_{MCK}$ + 270 | | ns |
| Data hold time (transmission) | $t_{HD:DAT}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V, $C_b$ = 100 pF, $R_b$ = 1.7 kΩ | 0 | 300 | ns |
| <R> | | 2.7 V ≤ $V_{DD}$ < 4.0 V, $C_b$ = 100 pF, $R_b$ = 2.7 kΩ | | | |

**Note** $f_{CLK} \leq f_{MCK}/4$ must also be satisfied.

Simplified $I^2C$ mode connection diagram (during communication at same potential)



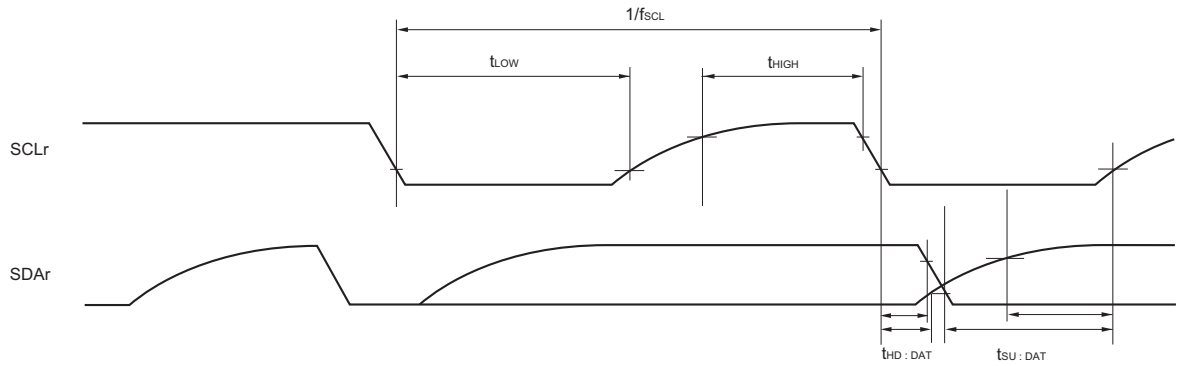

**Caution Select the normal input buffer and N-ch open-drain output mode for the SDAr pin and SCLr pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (SDAr, SCLr) pull-up resistance, $C_b$ [F]: Communication line (SDAr, SCLr) load capacitance, $V_b$[V]: Communication line voltage

**2.** r: IICr (r = 00, 01, 10, 11)

**3.** $f_{MCK}$: Serial array unit operation clock frequency

Simplified $I^2C$ mode serial transfer timing (during communication at same potential)




**Remark** r: IICr (r = 00, 01, 10, 11)

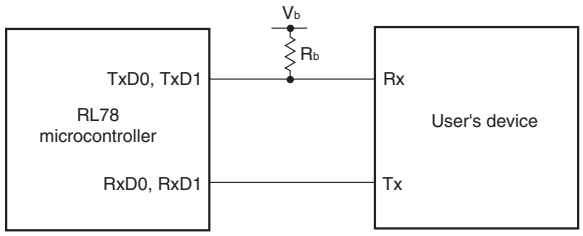
### (8) Communication at different potential (UART mode) (TxD output buffer: N-ch open-drain, RxD input buffer: TTL)

**($T_A$ = -40 to +150°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

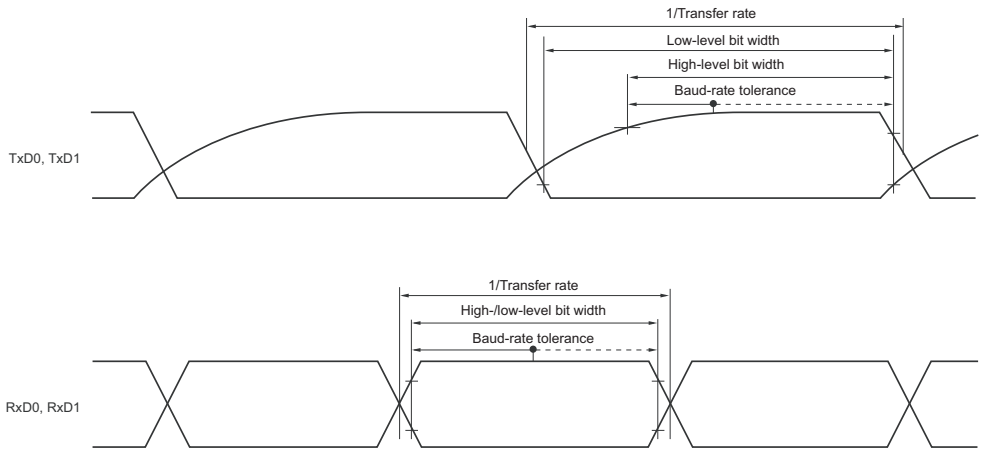
| Parameter | Symbol | Conditions | | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|---|
| Transfer rate | – | Reception | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $V_{IH}$ = 2.2 V, $V_{IL}$ = 0.8 V | | | | $f_{MCK}$/6 | bps |
| | | | | Theoretical value of the maximum transfer rate Note ($C_b$ = 30 pF) | | | 4.0 | Mbps |
| | | Transmission | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $V_{OH}$ = 2.2 V, $V_{OL}$ = 0.8 V | | | | Smaller number of the values given by $f_{MCK}$/6 and expression 1 is applicable. | bps |
| | | | | Theoretical value of the maximum transfer rate Note ($C_b$ = 30 pF) Normal slew rate | | | 4.0 | Mbps |

**Note** Expression 1: Maximum transfer rate = $1 / [\{-C_b \times R_b \times \ln (1 - 2.2/V_b)\} \times 3]$

UART mode connection diagram (during communication at different potential)




UART mode bit width (during communication at different potential) (reference)




**Caution Select the TTL input buffer for the RxD0 pin and RxD1 pin and N-ch open-drain output mode for the TxD0 pin and TxD1 pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (TxD) pull-up resistance, $C_b$ [F]: Communication line (TxD) load capacitance, $V_b$ [V]: Communication line voltage

**2.** $f_{MCK}$: Serial array unit operation clock frequency

### (9) During communication at different potential (3-V supply system) (CSI mode) (master mode, $\overline{SCKp}$ … internal clock output, normal slew rate)

**($T_A$ = -40 to +150°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

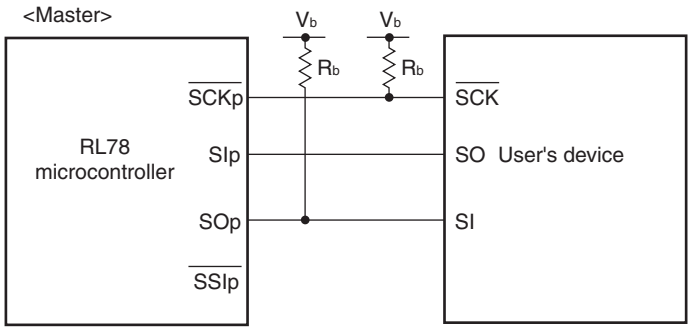
| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 400 Note3 | | | ns |
| $\overline{SCKp}$ high-level width | $t_{KH1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | $t_{KCY1}$/2 – 75 | | | ns |
| $\overline{SCKp}$ low-level width | $t_{KL1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | $t_{KCY1}$/2 – 20 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑) Note 1 | $t_{SIK1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 150 | | | ns |
| SIp setup time (to $\overline{SCKp}$↓) Note 2 | $t_{SIK1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 70 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑) Note 1 | $t_{KSI1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 30 | | | ns |
| SIp hold time (from $\overline{SCKp}$↓) Note 2 | $t_{KSI1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | 30 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output Note1 | $t_{KSO1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | | | 120 | ns |
| Delay time from $\overline{SCKp}$↑ to SOp output Note2 | $t_{KSO1}$ | 2.7 V ≤ $V_b$ ≤ $EV_{DD0}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | | | 40 | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

**2.** When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

**3.** $t_{KCY1} \geq 4/f_{CLK}$ must also be satisfied.

CSI mode connection diagram (during communication at different potential)




**Caution Select the TTL input buffer for the SIp pin and N-ch open-drain output mode for the SOp pin and $\overline{\text{SCKp}}$ pin.**

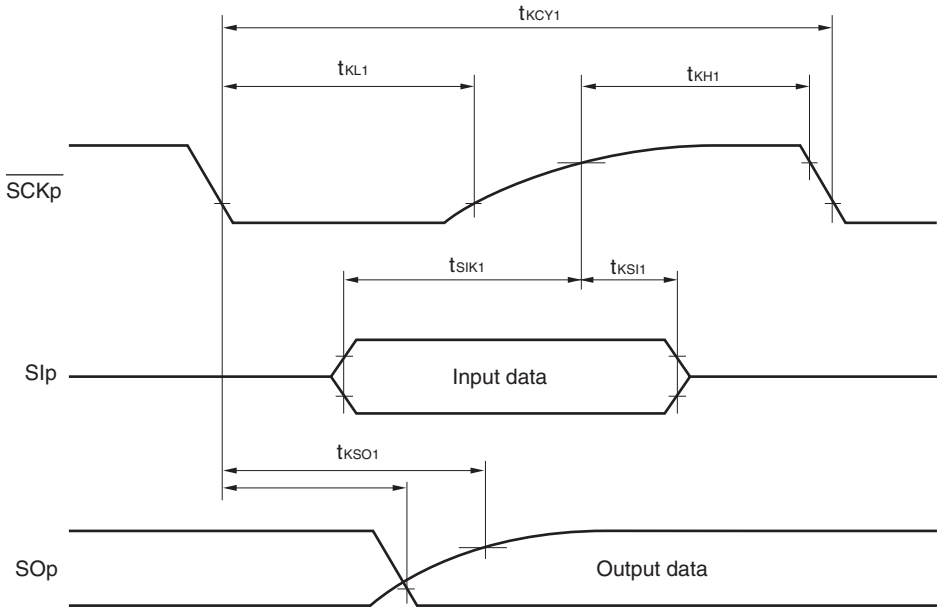
**Remarks 1.** $R_b$ [Ω]: Communication line ($\overline{\text{SCKp}}$, SOp) pull-up resistance, $C_b$ [F]: Communication line (SOp, $\overline{\text{SCKp}}$) load capacitance, $V_b$ [V]: Communication line voltage

**2.** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

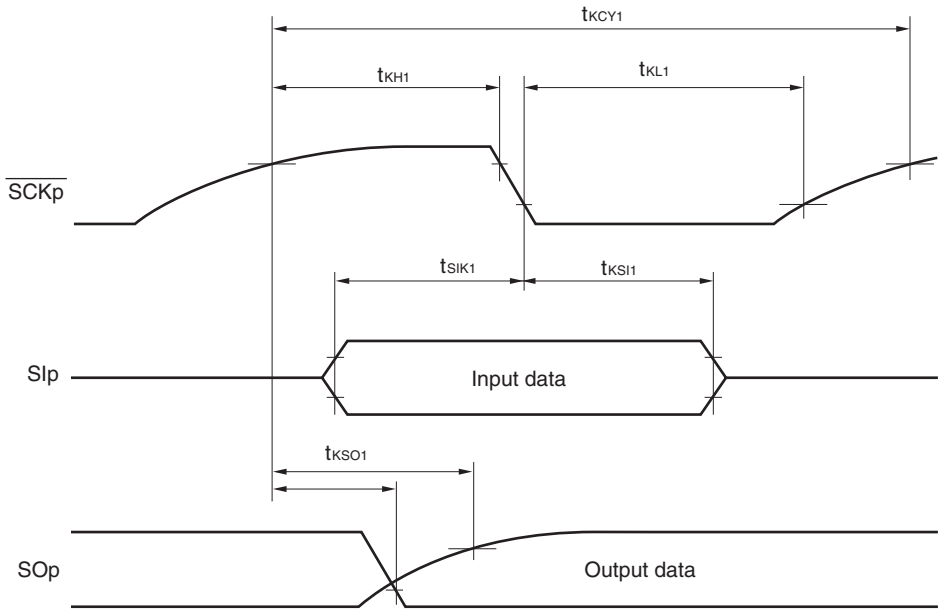
**3.** AC characteristics of the serial array unit during communication at different potential in CSI mode are measured with the $V_{IH}$ and $V_{IL}$ below:

When 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, 2.7 V ≤ $V_b$ ≤ 4.0 V: $V_{IH}$ = 2.2 V, $V_{IL}$ = 0.8 V

CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)




CSI mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn= 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



### (10) During communication at different potential (3-V supply system) (CSI mode) (slave mode, $\overline{SCKp}$ … external clock input, normal slew rate)

**($T_A$ = -40 to +150°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

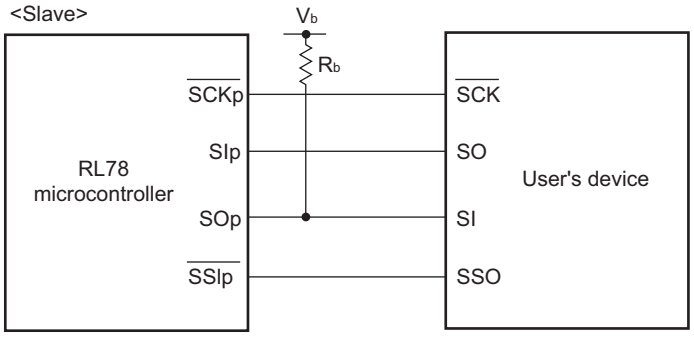
| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| $\overline{SCKp}$ cycle time | $t_{KCY2}$ | 2.7 V ≤ $V_b$ ≤ $V_{DD}$ | 20 MHz < $f_{MCK}$ ≤ 24 MHz | 12/$f_{MCK}$ | | | ns |
| | | | 8 MHz < $f_{MCK}$ ≤ 20 MHz | 10/$f_{MCK}$ | | | ns |
| | | | 4 MHz < $f_{MCK}$ ≤ 8 MHz | 8/$f_{MCK}$ | | | ns |
| | | | $f_{MCK}$ ≤ 4 MHz | 6/$f_{MCK}$ | | | ns |
| $\overline{SCKp}$ high-level width, low-level width | $t_{KH2}$, $t_{KL2}$ | 2.7 V ≤ $V_b$ ≤ $V_{DD}$ | | $t_{KCY2}$/2 – 20 | | | ns |
| SIp setup time (to $\overline{SCKp}$↑)[Note 1] | $t_{SIK2}$ | | | 90 | | | ns |
| SIp hold time (from $\overline{SCKp}$↑)[Note 2] | $t_{KSI2}$ | | | 1/$f_{MCK}$ + 50 | | | ns |
| Delay time from $\overline{SCKp}$↓ to SOp output[Note 3] | $t_{KSO2}$ | 2.7 V ≤ $V_b$ ≤ $V_{DD}$, $C_b$ = 30 pF, $R_b$ = 1.4 kΩ | | | | 2/$f_{MCK}$ + 120 | ns |
| $\overline{SSIp}$ setup time | $t_{SSIK}$ | DAP = 0 | | 120 | | | ns |
| | | DAP = 1 | | 1/$f_{MCK}$ + 120 | | | ns |
| $\overline{SSIp}$ hold time | $t_{KSSI}$ | DAP = 0 | | 1/$f_{MCK}$ + 120 | | | ns |
| | | DAP = 1 | | 120 | | | ns |

**Notes 1.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**2.** When DAPmn = 0 and CKPmn = 0 or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1 or DAPmn = 1 and CKPmn = 0.

**3.** When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

CSI mode connection diagram (during communication at different potential)




**Caution Select the TTL input buffer for the SIp, $\overline{\text{SCKp}}$ and $\overline{\text{SSIp}}$ pins and N-ch open-drain output mode for the SOp pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (SOp) pull-up resistance, $C_b$ [F]: Communication line (SOp) load capacitance, $V_b$ [V]: Communication line voltage

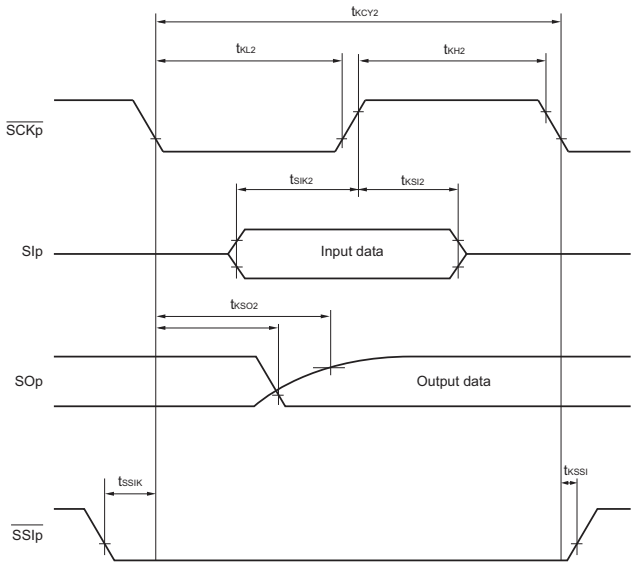
**2.** p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

**3.** AC characteristics of the serial array unit during communication at different potential in CSI mode are measured with the $V_{IH}$ and $V_{IL}$ below:

When 4.0 V ≤ $EV_{DD0}$ ≤ 5.5 V, 2.7 V ≤ $V_b$ ≤ 4.0 V: $V_{IH}$ = 2.2 V, $V_{IL}$ = 0.8 V

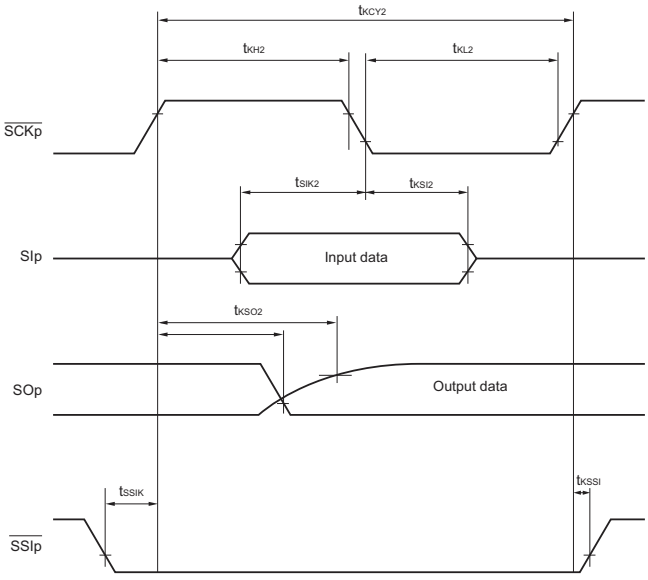
CSI mode serial transfer timing (slave mode) (during communication at different potential)

(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)




CSI mode serial transfer timing (slave mode) (during communication at different potential)

(When DAPmn= 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



**(11) During communication at different potential (3-V supply system) (simplified $I^2C$ mode)**

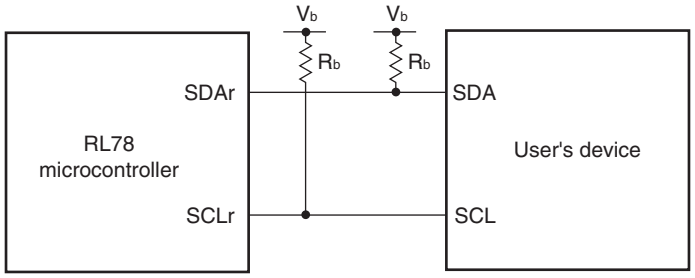
**(SDAr: TTL input buffer mode or N-ch open-drain output ($EV_{DD0}$ tolerance) mode, SCLr: N-ch open-drain output ($EV_{DD0}$ tolerance) mode)**

**($T_A$ = -40 to +150°C, 4.0 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

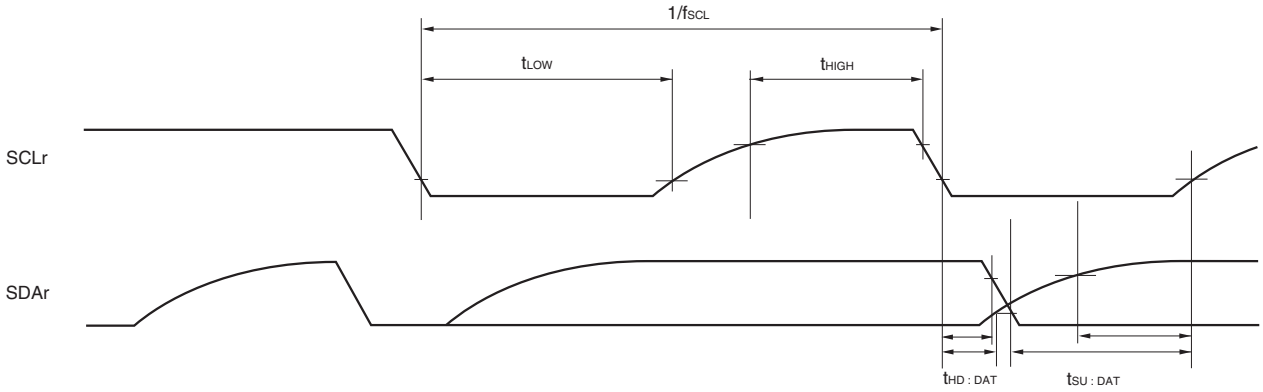
| Parameter | Symbol | Conditions | MIN. | MAX. | Unit |
|---|---|---|---|---|---|
| SCLr clock frequency | $f_{SCL}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | | 400[Note] | kHz |
| Hold time when SCLr = "L" | $t_{LOW}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | 1200 | | ns |
| Hold time when SCLr = "H" | $t_{HIGH}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | 600 | | ns |
| Data setup time (reception) | $t_{SU:DAT}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | 135 + 1/$f_{MCK}$ | | ns |
| Data hold time (transmission) | $t_{HD:DAT}$ | 2.7 V ≤ $V_b$ ≤ 4.0 V, $C_b$ = 100 pF, $R_b$ = 1.4 kΩ | 0 | 140 | ns |

**Note** $f_{SCL}$ ≤ $f_{MCK}$/4 must also be satisfied.

Simplified I²C mode connection diagram (during communication at different potential)




Simplified I²C mode serial transfer timing (during communication at different potential)




**Caution Select the TTL input buffer and the N-ch open-drain output mode for the SDAr pin and N-ch open-drain output mode for the SCLr pin.**

**Remarks 1.** $R_b$ [Ω]: Communication line (SDAr, SCLr) pull-up resistance, $C_b$ [F]: Communication line (SDAr, SCLr) load capacitance, $V_b$ [V]: Communication line voltage

**2.** $f_{MCK}$: Serial array unit operation clock frequency

### 36.5.2 Serial Interface IICA

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | Normal Mode | | Fast Mode | | Fast Mode Plus | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| SCLA0 clock frequency | $f_{SCL}$ | Fast mode plus: 10 MHz ≤ $f_{CLK}$ | | | | | 0 | 1000 | kHz |
| | | Fast mode: 3.5 MHz ≤ $f_{CLK}$ | | | 0 | 400 | | | kHz |
| | | Normal mode: 1 MHz ≤ $f_{CLK}$ | 0 | 100 | | | | | kHz |
| Setup time of restart condition[Note 1] | $t_{SU:STA}$ | | 4.7 | | 0.6 | | 0.26 | | µs |
| Hold time | $t_{HD:STA}$ | | 4.0 | | 0.6 | | 0.26 | | µs |
| Hold time when SCLA0 = "L" | $t_{LOW}$ | | 4.7 | | 1.3 | | 0.5 | | µs |
| Hold time when SCLA0 = "H" | $t_{HIGH}$ | | 4.0 | | 0.6 | | 0.26 | | µs |
| Data setup time (reception) | $t_{SU:DAT}$ | | 250 | | 100 | | 50 | | ns |
| Data hold time (transmission)[Note 2] | $t_{HD:DAT}$ | | 0 | 3.45 | 0 | 0.9 | 0 | | µs |
| Setup time of stop condition | $t_{SU:STO}$ | | 4.0 | | 0.6 | | 0.26 | | µs |
| Bus-free time | $t_{BUF}$ | | 4.7 | | 1.3 | | 0.5 | | µs |

**Notes 1.** The first clock pulse is generated after this period when the start/restart condition is detected.

**2.** The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a wait state is inserted in the $\overline{ACK}$ (acknowledge) timing.

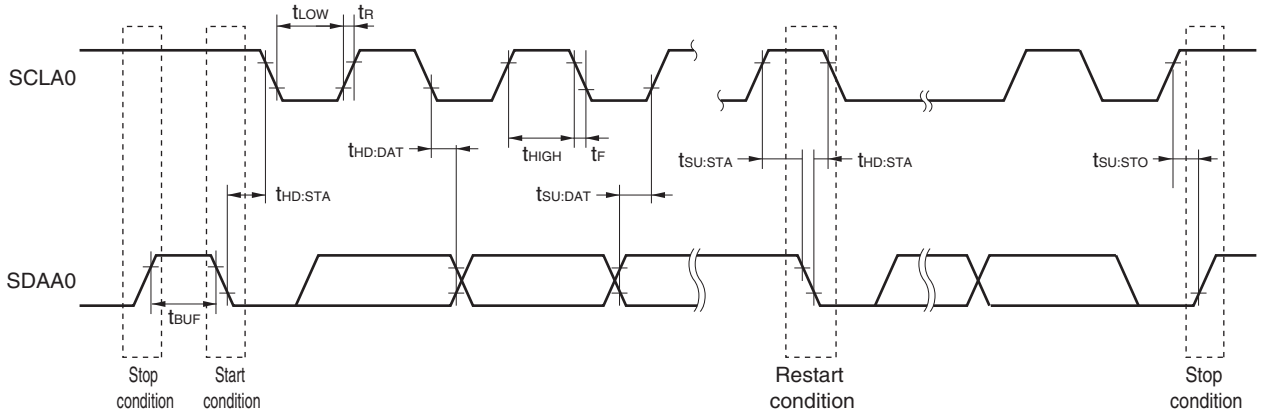
**Remark** The maximum value of $C_b$ (communication line capacitance) and the value of $R_b$ (communication line pull-up resistor) at that time in each mode are as follows.

Standard mode: $C_b$ = 400 pF, $R_b$ = 2.7 kΩ

Fast mode: $C_b$ = 320 pF, $R_b$ = 1.1 kΩ

Fast mode plus: $C_b$ = 120 pF, $R_b$ = 1.1 kΩ

IICA serial transfer timing



### 36.5.3 On-chip Debug (UART)

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Transfer rate | - | | 115.2 k | | 1 M | bps |

### 36.5.4 LIN/UART Module (RLIN3) UART Mode

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Transfer rate | - | Operation mode, HALT mode | LIN communication clock source ($f_{CLK}$ or $f_{MX}$): 4 to 24 MHz | | | 4000 | kbps |
| | | SNOOZE mode | LIN communication clock source ($f_{CLK}$): 1 to 24 MHz FRQSEL4 = 0 in the user option byte (000C2H/020C2H) | | | 4.8 | |
| | | | LIN communication clock source ($f_{CLK}$): 1 to 24 MHz FRQSEL4 = 1 in the user option byte (000C2H/020C2H) | | | 2.4 | |

## 36.6 Analog Characteristics

### 36.6.1 A/D Converter Characteristics

**(1) When $AV_{REF}$ (+) = $AV_{REFP}$/ANI0 (ADREFP1 = 0, ADREFP0 = 1), $AV_{REF}$ (-) = $AV_{REFM}$/ANI1 (ADREFM = 1), target ANI pin: ANI2 to ANI23 (power supply: $V_{DD}$)**

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V, Reference voltage (+) = $AV_{REFP}$, Reference voltage (-) = $AV_{REFM}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | 8 | | 10 | bit |
| Overall error[Note 1] | AINL | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.2 | ±3.0 | LSB |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.2 | ±3.5 | LSB |
| Conversion time | $t_{CONV}$ | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 2.125 | | 39 | μs |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 3.1875 | | 39 | μs |
| Zero-scale error[Notes 1, 2] | EZS | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.25 | %FSR |
| Full-scale error[Notes 1, 2] | EFS | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.25 | %FSR |
| Integral linearity error[Note 1] | ILE | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.5 | LSB |
| Differential linearity error[Note 1] | DLE | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±1.5 | LSB |
| Reference voltage (+) | $AV_{REFP}$ | | | 2.7 | | $V_{DD}$ | V |
| Analog input voltage | $V_{AIN}$ | | | 0 | | $AV_{REFP}$ | V |
| Internal reference voltage (+) | $V_{BGR}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.38 | 1.45 | 1.52 | V |

**Notes 1.** Excludes quantization error (±1/2 LSB).

**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

**(2) When $AV_{REF}$ (+) = $AV_{REFP}$/ANI0 (ADREFP1 = 0, ADREFP0 = 1), $AV_{REF}$ (-) = $AV_{REFM}$/ANI1 (ADREFM = 1), target ANI pin: ANI24 to ANI30 (power supply: $EV_{DD0}$)**

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V, Reference voltage (+) = $AV_{REFP}$, Reference voltage (-) = $AV_{REFM}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | 8 | | 10 | bit |
| Overall error[Note 1] | AINL | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.2 | ±4.5 | LSB |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.2 | ±5.0 | LSB |
| Conversion time | $t_{CONV}$ | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 2.125 | | 39 | μs |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 3.1875 | | 39 | μs |
| Zero-scale error[Notes 1, 2] | EZS | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.35 | %FSR |
| Full-scale error[Notes 1, 2] | EFS | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.35 | %FSR |
| Integral linearity error[Note 1] | ILE | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±3.5 | LSB |
| Differential linearity error[Note 1] | DLE | 10-bit resolution $AV_{REFP}$ = $V_{DD}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.0 | LSB |
| Reference voltage (+) | $AV_{REFP}$ | | | 2.7 | | $V_{DD}$ | V |
| Analog input voltage | $V_{AIN}$ | | | 0 | | $AV_{REFP}$ and $EV_{DD0}$ | V |
| Internal reference voltage (+) | $V_{BGR}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.38 | 1.45 | 1.52 | V |

**Notes 1.** Excludes quantization error (±1/2 LSB).

**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

**(3) When $AV_{REF}$ (+) = $V_{DD}$ (ADREFP1 = 0, ADREFP0 = 0), $AV_{REF}$ (-) = $V_{SS}$ (ADREFM = 0), target ANI pin: ANI0 to ANI23, ANI24 to ANI30**

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V, Reference voltage (+) = $V_{DD}$, Reference voltage (-) = $V_{SS}$)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | 8 | | 10 | bit |
| Overall error[Note 1] | AINL | 10-bit resolution ANI0 to ANI23 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.2 | ±5.0 | LSB |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.2 | ±5.5 | LSB |
| | | 10-bit resolution ANI24 to ANI30 | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.2 | ±6.5 | LSB |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | | 1.2 | ±7.0 | LSB |
| Conversion time | $t_{CONV}$ | 10-bit resolution | 4.0 V ≤ $V_{DD}$ ≤ 5.5 V | 2.125 | | 39 | μs |
| | | | 2.7 V ≤ $V_{DD}$ < 4.0 V | 3.1875 | | 39 | μs |
| Zero-scale error[Notes 1, 2] | EZS | 10-bit resolution ANI0 to ANI23 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.50 | %FSR |
| | | 10-bit resolution ANI24 to ANI30 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.60 | %FSR |
| Full-scale error[Notes 1, 2] | EFS | 10-bit resolution ANI0 to ANI23 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.50 | %FSR |
| | | 10-bit resolution ANI24 to ANI30 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.60 | %FSR |
| Integral linearity error[Note 1] | ILE | 10-bit resolution ANI0 to ANI23 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±3.5 | LSB |
| | | 10-bit resolution ANI24 to ANI30 | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±4.0 | LSB |
| Differential linearity error[Note 1] | DLE | 10-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.0 | LSB |
| Analog input voltage | $V_{AIN}$ | ANI0 to ANI23[Note 3] | | 0 | | $V_{DD}$ | V |
| | | ANI24 to ANI30[Note 3] | | $EV_{SS}$ | | $EV_{DD0}$ | V |
| Internal reference voltage (+) | $V_{BGR}$ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | 1.38 | 1.45 | 1.52 | V |

**Notes 1.** Excludes quantization error (±1/2 LSB).

**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

**3.** The number of pins depends on the product. For details, refer to **2.1 Pin Function List**.

**(4) When $AV_{REF}$ (+) = internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), $AV_{REF}$ (-) = $AV_{REFM}$/ANI1 (ADREFM = 1), target ANI pin: ANI0 to ANI23, ANI24 to ANI30**

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V, Reference voltage (+) = $V_{BGR}$, Reference voltage (-) = $AV_{REFM}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | | 8 | | bit |
| Conversion time | $t_{CONV}$ | 8-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | 17 | | 39 | μs |
| Zero-scale error[Notes 1, 2] | EZS | 8-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±0.60 | %FSR |
| Integral linearity error[Note 1] | ILE | 8-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±2.0 | LSB |
| Differential linearity error[Note 1] | DLE | 8-bit resolution | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | ±1.0 | LSB |
| Reference voltage (+) | $V_{BGR}$ | | | 1.38 | 1.45 | 1.52 | V |
| Analog input voltage | $V_{AIN}$ | | | 0 | | $V_{BGR}$ | V |

**Notes 1.** Excludes quantization error (±1/2 LSB).

**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

### 36.6.2 Temperatures Sensor Characteristics

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Temperature sensor output voltage | $V_{TMPS25}$ | Setting ADS register = 80H, $T_A$ = +25°C | | 1.1 | | V |
| Reference output voltage | $V_{CONST}$ | Setting ADS register = 81H | 1.38 | 1.45 | 1.52 | V |
| Temperature coefficient | $F_{VTMPS}$ | Temperature sensor that depends on the temperature | | -3.3 | | mV/°C |
| Operation stabilization wait time | $t_{AMP}$ | | 5 | | | μs |

### 36.6.3 D/A Converter Characteristics

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Resolution | RES | | | | | 8 | bit |
| Overall error | AINL | Rload = 4 MΩ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | -2.5/+3.0 | LSB |
| | | Rload = 8 MΩ | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | -2.5/+3.0 | LSB |
| Settling time | $t_{SET}$ | Cload = 20 pF | 2.7 V ≤ $V_{DD}$ ≤ 5.5 V | | | 3 | μs |

### 36.6.4 Comparator Characteristics

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input offset voltage | $V_{IOCMP}$ | | | ±5 | ±90 | mV |
| Input voltage range | $V_{ICMP}$ | | 0 | | $V_{DD}$ | V |
| Response time | $t_{CR}$, $t_{CF}$ | Input amplitude ±100 mV | | 70 | 700 | ns |
| Stabilization wait time during input channel switching[Note 1] | $t_{WAIT}$ | Input amplitude ±100 mV | 800 | | | ns |
| Operation stabilization wait time[Note 2] | $t_{CMP}$ | 3.3 V ≤ $V_{DD}$ ≤ 5.5 V | 1 | | | μs |
| | | 2.7 V ≤ $V_{DD}$ < 3.3 V | 3 | | | μs |

**Notes 1.** Period of time from when the comparator input channel is switched until the comparator is switched to output

**2.** Period of time from when the comparator operation is enabled (HCMPON bit in CMPCTL is set to 1) until the comparator satisfies the DC/AC characteristics.

### 36.6.5 POR Circuit Characteristics

**($T_A$ = -40 to +150°C, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Detection voltage[Note] | $V_{POR}$ | Power supply rise time | 1.48 | 1.56 | 1.67 | V |
| | $V_{PDR}$ | Power supply fall time | 1.47 | 1.55 | 1.66 | V |
| Minimum pulse width | $T_{PW}$ | | 300 | | | μs |
| Detection delay time | $T_{PD}$ | | | | 350 | μs |

**Note** This indicates the POR circuit characteristics, and normal operation is not guaranteed under the condition of less than lower limit operation voltage (2.7 V).

### 36.6.6 LVD Circuit Characteristics

#### (1) LVD detection voltage of interrupt mode or reset mode

**($T_A$ = -40 to +150°C, $V_{PDR} \le EV_{DD0} = EV_{DD1} = V_{DD} \le 5.5$ V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)**

| Parameter | | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Detection voltage | Supply voltage level | $V_{LVD0}$ | Power supply rise time | 4.62 | 4.74 | 5.22 | V |
| | | | Power supply fall time | 4.52 | 4.64 | 5.11 | V |
| | | $V_{LVD1}$ | Power supply rise time | 4.50 | 4.62 | 5.09 | V |
| | | | Power supply fall time | 4.40 | 4.52 | 4.98 | V |
| | | $V_{LVD2}$ | Power supply rise time | 4.30 | 4.42 | 4.87 | V |
| | | | Power supply fall time | 4.21 | 4.32 | 4.76 | V |
| | | $V_{LVD3}$ | Power supply rise time | 3.13 | 3.22 | 3.55 | V |
| | | | Power supply fall time | 3.07 | 3.15 | 3.47 | V |
| | | $V_{LVD4}$ | Power supply rise time | 2.95 | 3.02 | 3.33 | V |
| | | | Power supply fall time | 2.89 | 2.96 | 3.23 | V |
| | | $V_{LVD5}$ | Power supply rise time | 2.74 | 2.81 | 3.11 | V |
| | | | Power supply fall time | 2.68[Note] | 2.75 | 3.00 | V |
| Minimum pulse width | | $t_{LW}$ | | 300 | | | μs |
| Detection delay time | | $t_{LD}$ | | | | 300 | μs |

**Note** The minimum value exceeds below the lower limit operation voltage (2.7 V), however, in reset mode, normal operation (same behavior when $V_{DD}$ = 2.7 V) is possible until a reset is effected at the power supply falling time.

#### (2) LVD detection voltage of interrupt and reset mode

**($T_A$ = -40 to +150°C, $V_{PDR} \le EV_{DD0} = EV_{DD1} = V_{DD} \le 5.5$ V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Interrupt and reset mode | $V_{LVD5}$ | VPOC2, VPOC1, VPOC0 = 0, 0, 1[Note 1], falling reset voltage: 2.75 V | | 2.68[Note 2] | 2.75 | 3.00 | V |
| | $V_{LVD2}$ | LVIS1, LVIS0 = 1, 0 | Rising release reset voltage | 4.30 | 4.42 | 4.87 | V |
| | | | Falling interrupt voltage | 4.21 | 4.32 | 4.76 | V |
| | $V_{LVD5}$ | VPOC2, VPOC1, VPOC0 = 0, 1, 0[Note 1], falling reset voltage: 2.75 V | | 2.68[Note 2] | 2.75 | 3.00 | V |
| | $V_{LVD1}$ | LVIS1, LVIS0 = 0, 0 | Rising release reset voltage | 4.50 | 4.62 | 5.09 | V |
| | | | Falling interrupt voltage | 4.40 | 4.52 | 4.98 | V |
| | $V_{LVD5}$ | VPOC2, VPOC1, VPOC0 = 0, 1, 1[Note 1], falling reset voltage: 2.75 V | | 2.68[Note 2] | 2.75 | 3.00 | V |
| | $V_{LVD3}$ | LVIS1, LVIS0 = 0, 1 | Rising release reset voltage | 3.13 | 3.22 | 3.55 | V |
| | | | Falling interrupt voltage | 3.07 | 3.15 | 3.47 | V |
| | $V_{LVD0}$ | LVIS1, LVIS0 = 0, 0 | Rising release reset voltage | 4.62 | 4.74 | 5.22 | V |
| | | | Falling interrupt voltage | 4.52 | 4.64 | 5.11 | V |

**Notes 1.** These values indicate setting values of option bytes.

**2.** The minimum value exceeds below the lower limit operation voltage (2.7 V), however, in reset mode, normal operation (same behavior when $V_{DD}$ = 2.7 V) is possible until a reset is effected at the power supply falling time.

## 36.7 Power Supply Voltage Rising Time

**($T_A$ = -40 to +150°C, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Maximum power supply voltage rising slope | $S_{vrmax}$ | 0 V → $V_{DD}$ (VPOC2 = 0 or 1[Note 2]) | | | 50[Note 3] | V/ms |
| Minimum power supply voltage rising slope [Note 1] | $S_{vrmin}$ | 0 V → 2.7 V | 6.5 | | | V/ms |

**Notes 1.** The minimum power supply voltage rising slope is applied only under the following condition.
When the voltage detection (LVD) circuit is not used (VPOC2 = 1) and an external reset circuit is not used or when a reset is not effected until $V_{DD}$ = 2.7 V.

**2.** These values indicate setting values of option bytes.

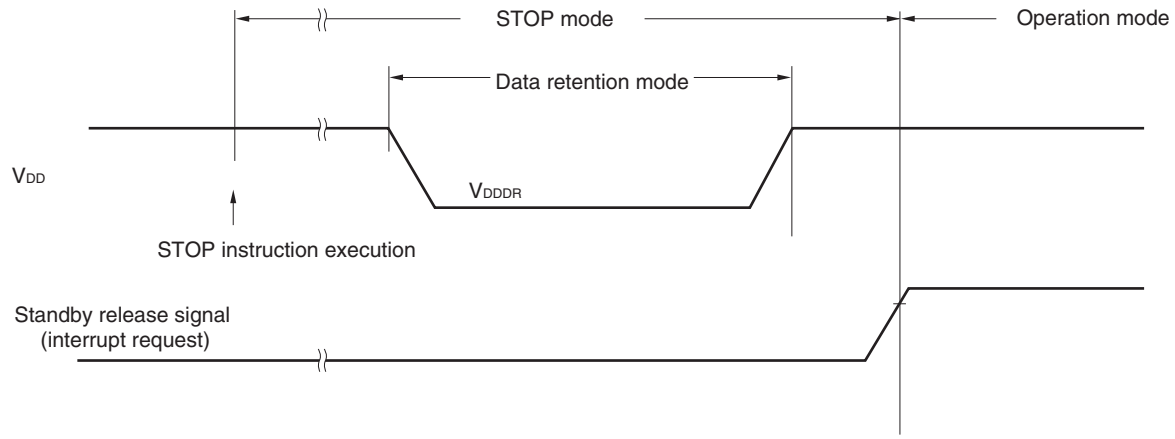
**3.** If the power supply drops below $V_{PDR}$ and a POR reset is effected, this specification is also applied when the power supply is recovered without dropping to 0 V.

## 36.8 STOP Mode Memory Retention Characteristics

**($T_A$ = -40 to +150°C, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Data retention supply voltage | $V_{DDDR}$ | | 1.47[Note] | | 5.5 | V |

**Note** The value depends on the POR detection voltage. When the voltage drops, the data is retained before a POR reset is effected, but data is not retained when a POR reset is effected.



## 36.9 Flash Memory Programming Characteristics

**($T_A$ = -40 to +150°C, 2.7 V ≤ $EV_{DD0}$ = $EV_{DD1}$ = $V_{DD}$ ≤ 5.5 V, $V_{SS}$ = $EV_{SS0}$ = $EV_{SS1}$ = 0 V)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| System clock frequency | $f_{CLK}$ | | 1 | | 24 | MHz |
| Number of code flash rewrites Notes 1, 2, 3 | $C_{erwr}$ | Retained for 20 years (after rewrite) $T_A$ = +85°C Note 4 | 1,000 | | | Times |
| Number of data flash rewrites Notes 1, 2, 3 | | Retained for 20 years (after rewrite) $T_A$ = +85°C Note 4 | 10,000 | | | |
| | | Retained for 5 years (after rewrite) $T_A$ = +85°C Note 4 | 100,000 | | | |
| <R> Erase time | $T_{erasa}$ | Block erase | 5 | | | ms |
| Write time | $T_{wrwa}$ | 1 word write | 10 | | | μs |

**Notes 1.** 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

**2.** When using flash memory programmer and Renesas Electronics self programming library

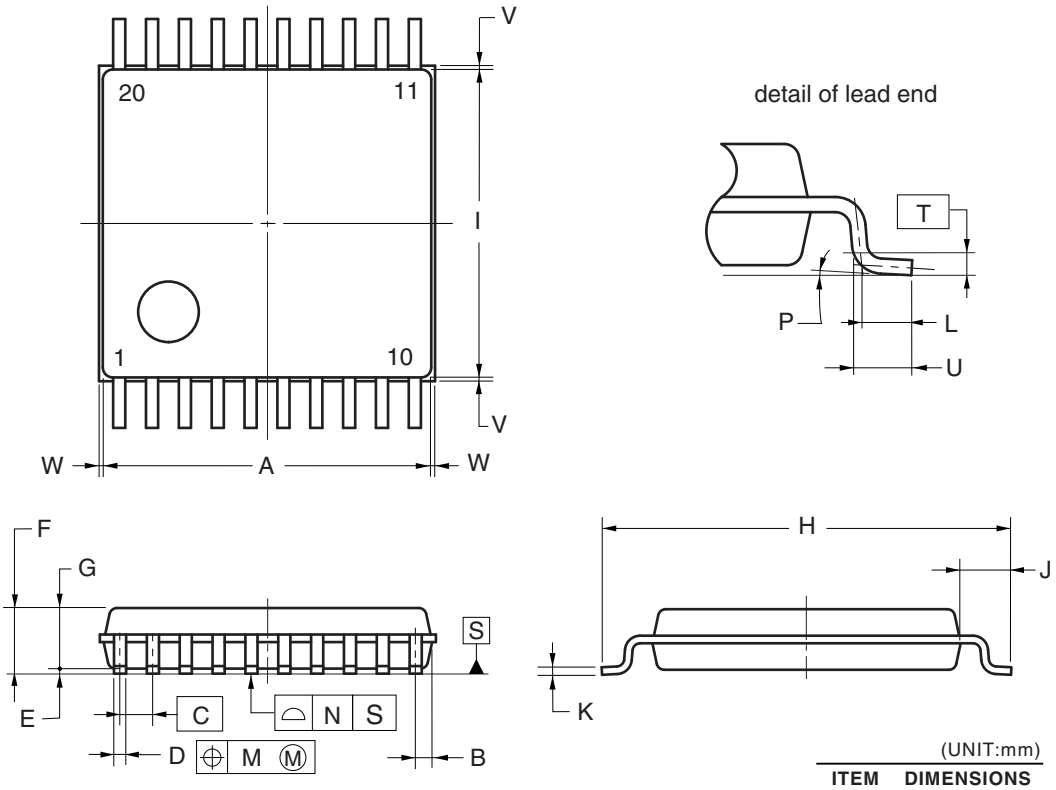
**3.** These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

**4.** The specified data retention time is given under the condition that the average temperature ($T_A$) is 85°C or below.

# CHAPTER 37 PACKAGE DRAWING

## 37.1 20-pin products

### 20-PIN PLASTIC SSOP (7.62 mm (300))




(UNIT:mm)

| ITEM | DIMENSIONS |
|---|---|
| A | 6.50±0.10 |
| B | 0.325 |
| C | 0.65 (T.P.) |
| D | $0.22^{+0.10}_{-0.05}$ |
| E | 0.10±0.05 |
| F | 1.30±0.10 |
| G | 1.20 |
| H | 8.10±0.20 |
| I | 6.10±0.10 |
| J | 1.00±0.20 |
| K | $0.15^{+0.05}_{-0.01}$ |
| L | 0.50 |
| M | 0.13 |
| N | 0.10 |
| P | $3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$ |
| T | 0.25(T.P) |
| U | 0.60±0.15 |
| V | 0.25 MAX. |
| W | 0.15 MAX. |

P20MC-65-CAA-1

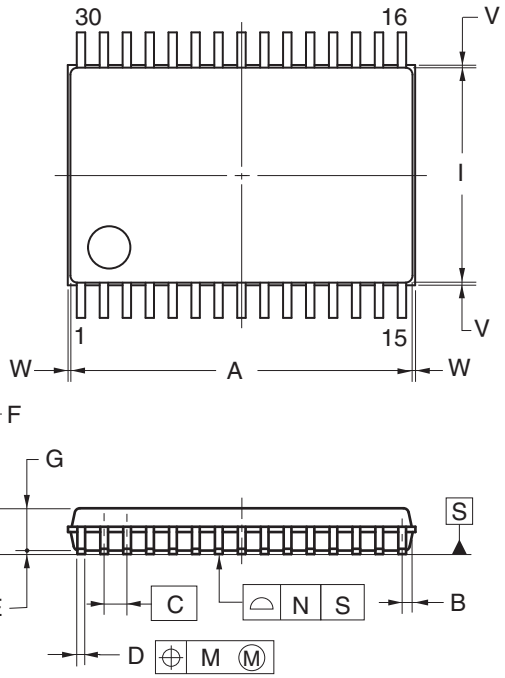
**NOTE**

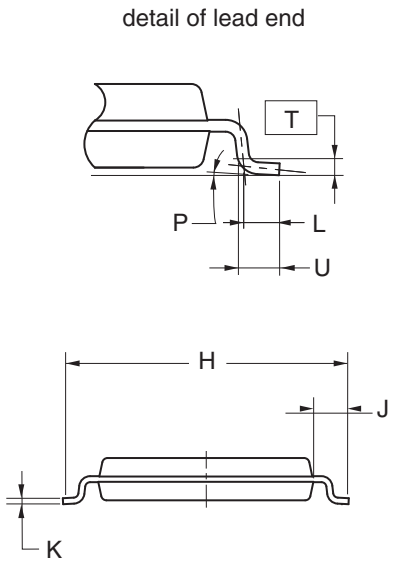
Each lead centerline is located within 0.13 mm of its true position (T.P. ) at maximum material condition.

©2010 Renesas Electronics Corporation. All rights reserved.

## 37.2 30-pin products

| JEITA Package Code | RENESAS Code | Previous Code | MASS (TYP.) [g] |
|---|---|---|---|
| P-LSSOP30-6.1x9.7-0.65 | PLSP0030JB-A | P30MC-65-CAB-2 | 0.18 |




(UNIT:mm)

| ITEM | DIMENSIONS |
|---|---|
| A | 9.70±0.10 |
| B | 0.30 |
| C | 0.65 (T.P. ) |
| D | $0.22^{+0.10}_{-0.05}$ |
| E | 0.10±0.05 |
| F | 1.30±0.10 |
| G | 1.20 |
| H | 8.10±0.20 |
| I | 6.10±0.10 |
| J | 1.00±0.20 |
| K | $0.15^{+0.05}_{-0.01}$ |
| L | 0.50 |
| M | 0.13 |
| N | 0.10 |
| P | $3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$ |
| T | 0.25(T.P. ) |
| U | 0.60±0.15 |
| V | 0.25 MAX. |
| W | 0.15 MAX. |

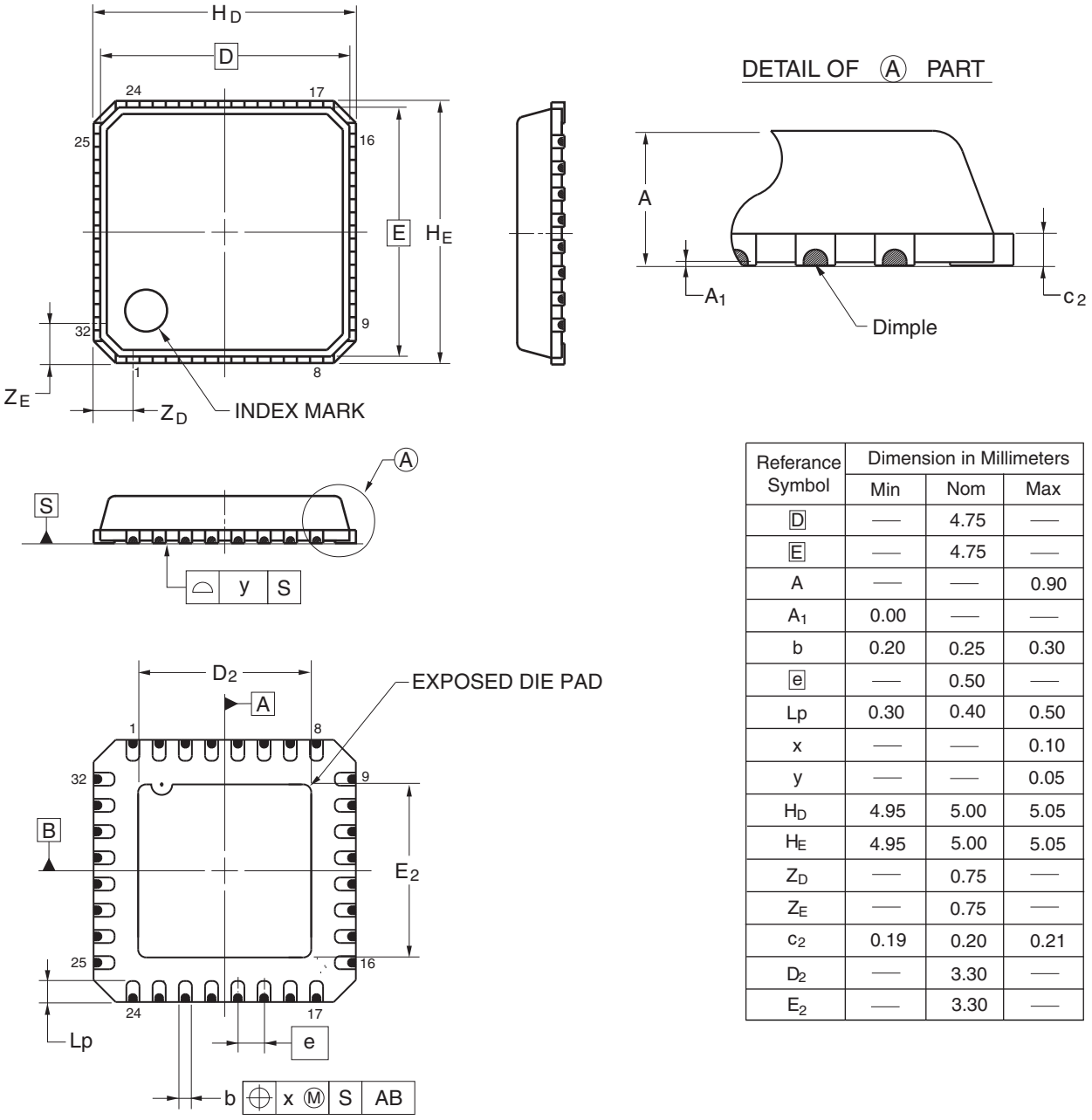
**NOTE**

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition .

©2012 Renesas Electronics Corporation. All rights reserved.

### 37.3 32-pin products

| JEITA Package code | RENESAS code | Previous code | MASS(TYP.)[g] |
|---|---|---|---|
| P-HVQFN32-5x5-0.50 | PVQN0032KD-A | P32K9-50A-BAH | 0.058 |

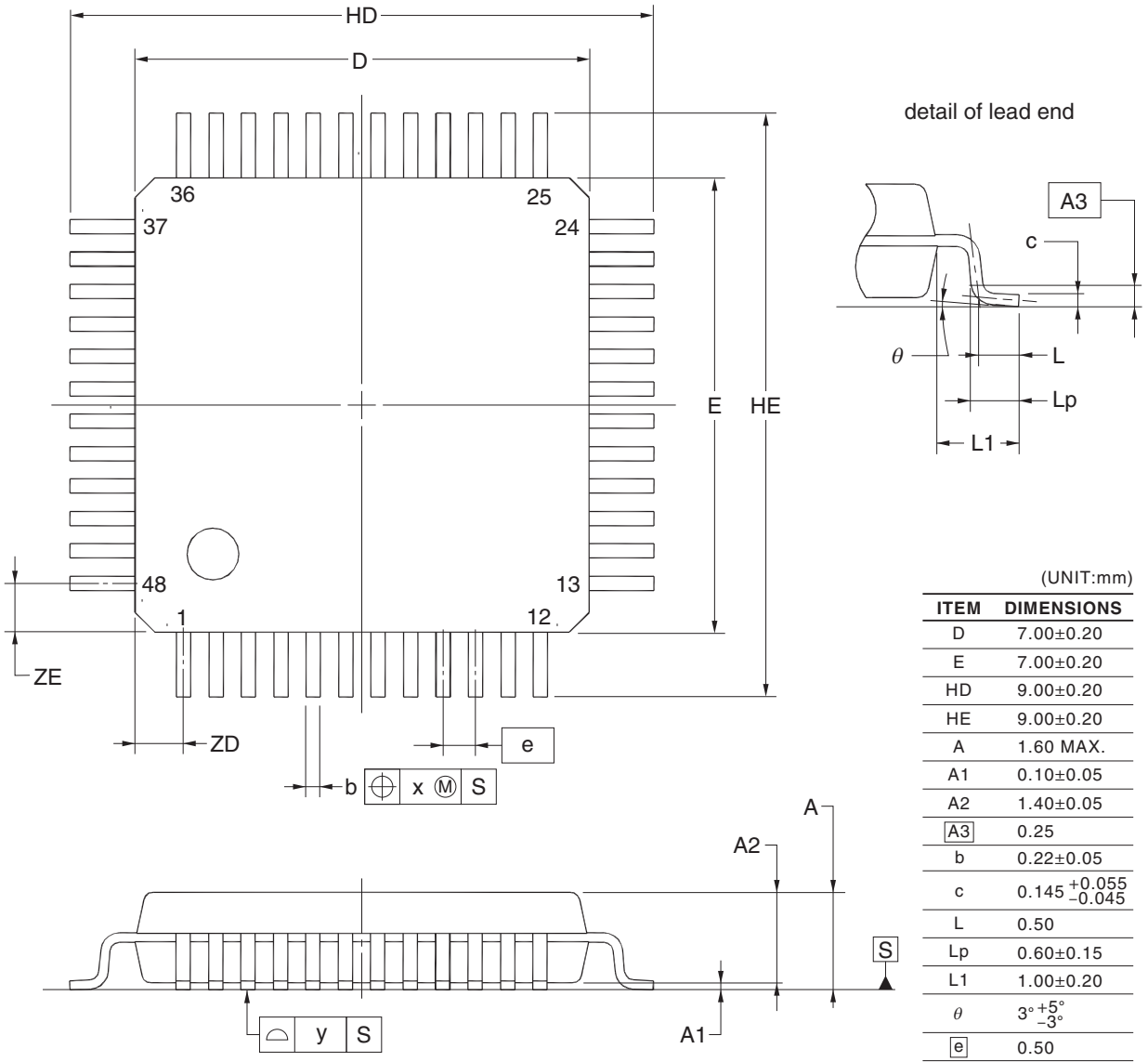



| Referance Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | — | 4.75 | — |
| E | — | 4.75 | — |
| A | — | — | 0.90 |
| $A_1$ | 0.00 | — | — |
| b | 0.20 | 0.25 | 0.30 |
| e | — | 0.50 | — |
| Lp | 0.30 | 0.40 | 0.50 |
| x | — | — | 0.10 |
| y | — | — | 0.05 |
| $H_D$ | 4.95 | 5.00 | 5.05 |
| $H_E$ | 4.95 | 5.00 | 5.05 |
| $Z_D$ | — | 0.75 | — |
| $Z_E$ | — | 0.75 | — |
| $c_2$ | 0.19 | 0.20 | 0.21 |
| $D_2$ | — | 3.30 | — |
| $E_2$ | — | 3.30 | — |

## 37.4 48-pin products

### 37.4.1 48-pin LQFP

| JEITA Package Code | RENESAS Code | Previous Code | MASS (TYP.) [g] |
|---|---|---|---|
| P-LFQFP48-7x7-0.50 | PLQP0048KF-A | P48GA-50-8EU-1 | 0.16 |




(UNIT:mm)

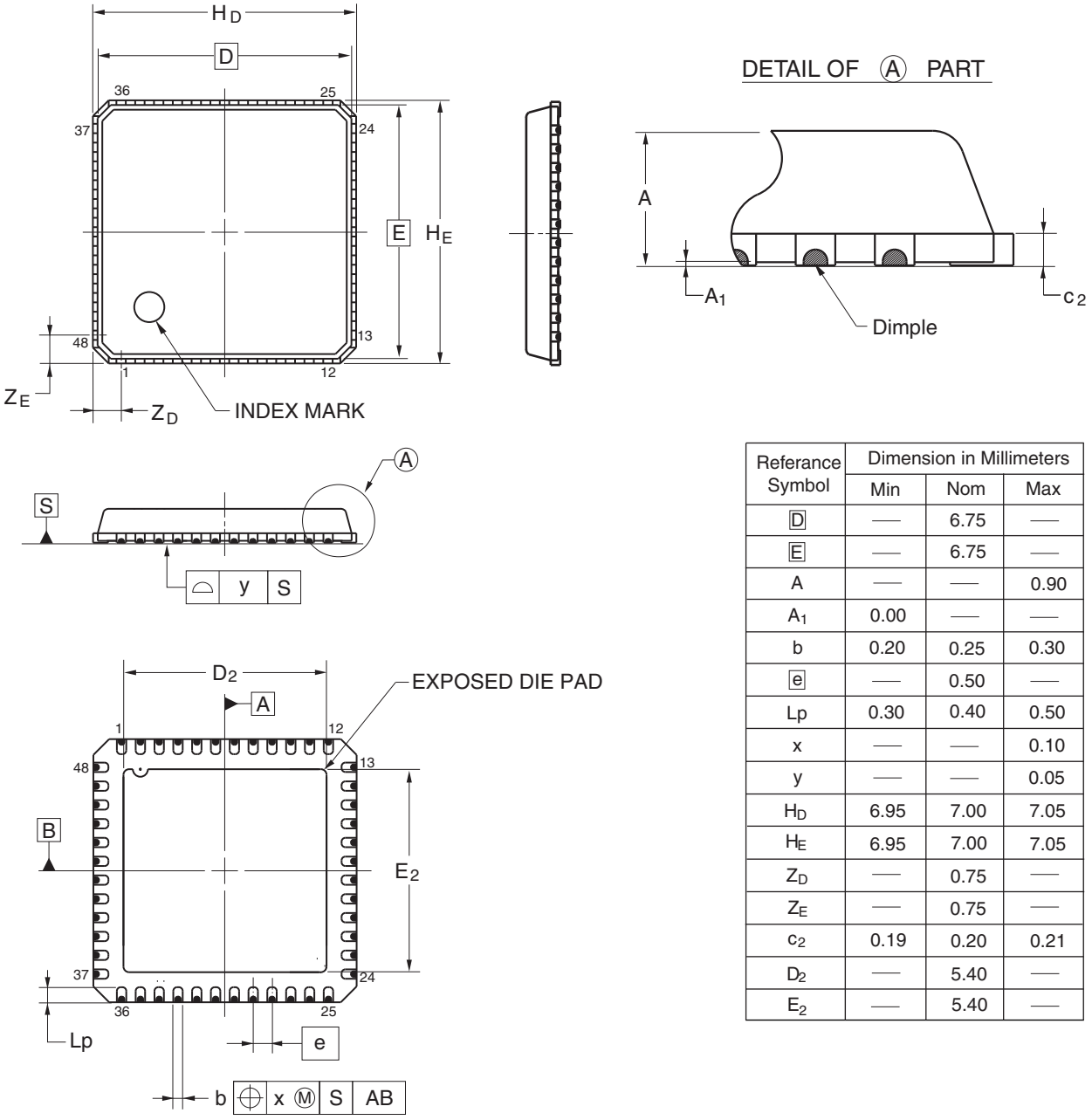
| ITEM | DIMENSIONS |
|---|---|
| D | 7.00±0.20 |
| E | 7.00±0.20 |
| HD | 9.00±0.20 |
| HE | 9.00±0.20 |
| A | 1.60 MAX. |
| A1 | 0.10±0.05 |
| A2 | 1.40±0.05 |
| A3 | 0.25 |
| b | 0.22±0.05 |
| c | $0.145^{+0.055}_{-0.045}$ |
| L | 0.50 |
| Lp | 0.60±0.15 |
| L1 | 1.00±0.20 |
| θ | $3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$ |
| e | 0.50 |
| x | 0.08 |
| y | 0.08 |
| ZD | 0.75 |
| ZE | 0.75 |

**NOTE**
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

©2012 Renesas Electronics Corporation. All rights reserved.

### 37.4.2 48-pin VQFN

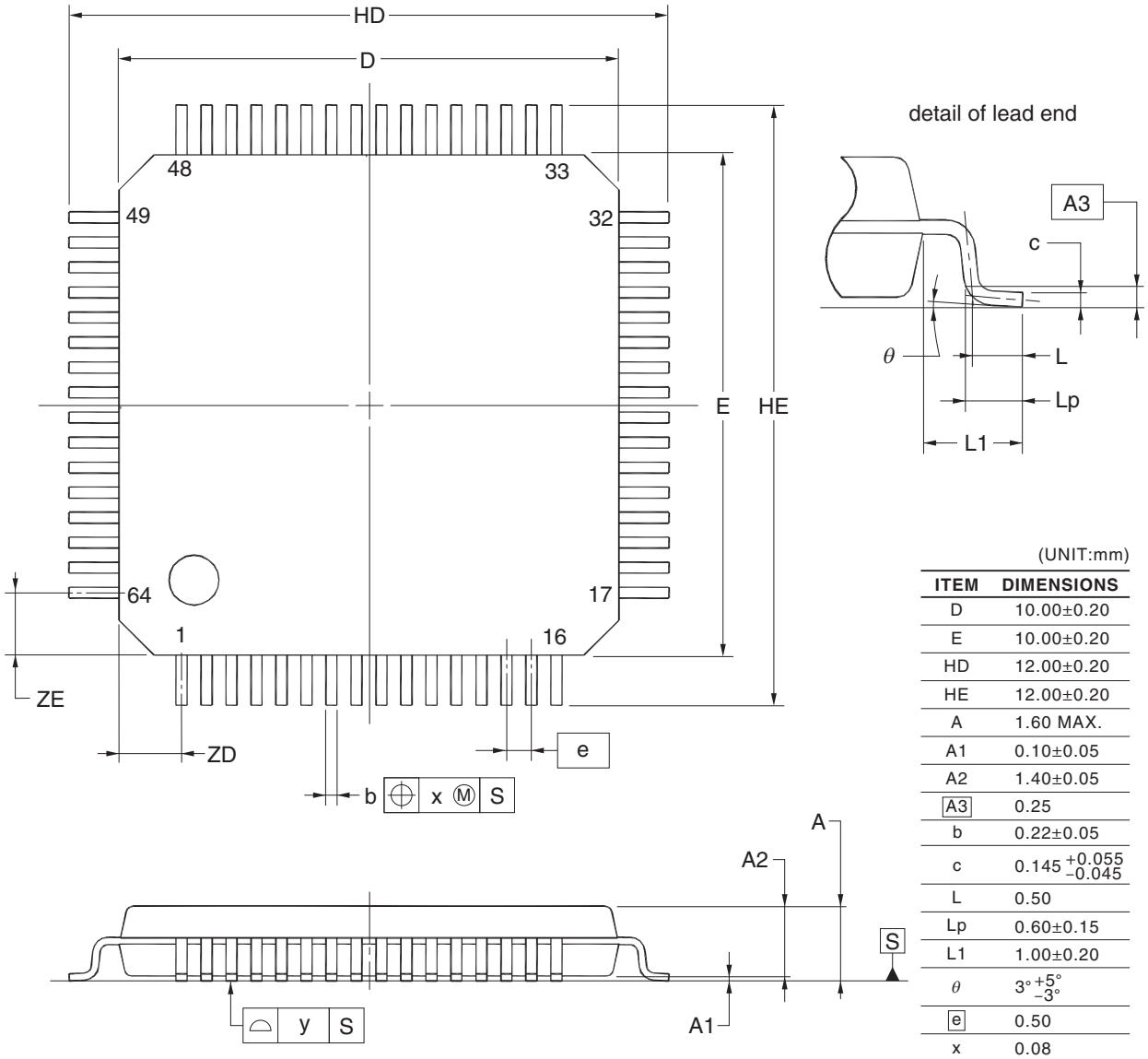
| JEITA Package code | RENESAS code | Previous code | MASS(TYP.)[g] |
|---|---|---|---|
| P-HVQFN48-7x7-0.50 | PVQN0048KG-A | P48K9-50A-BAJ | 0.13 |




| Referance Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| D | — | 6.75 | — |
| E | — | 6.75 | — |
| A | — | — | 0.90 |
| $A_1$ | 0.00 | — | — |
| b | 0.20 | 0.25 | 0.30 |
| e | — | 0.50 | — |
| Lp | 0.30 | 0.40 | 0.50 |
| x | — | — | 0.10 |
| y | — | — | 0.05 |
| $H_D$ | 6.95 | 7.00 | 7.05 |
| $H_E$ | 6.95 | 7.00 | 7.05 |
| $Z_D$ | — | 0.75 | — |
| $Z_E$ | — | 0.75 | — |
| $c_2$ | 0.19 | 0.20 | 0.21 |
| $D_2$ | — | 5.40 | — |
| $E_2$ | — | 5.40 | — |

## 37.5 64-pin products

| JEITA Package Code | RENESAS Code | Previous Code | MASS (TYP.) [g] |
|---|---|---|---|
| P-LFQFP64-10x10-0.50 | PLQP0064KF-A | P64GB-50-UEU-2 | 0.35 |




(UNIT:mm)

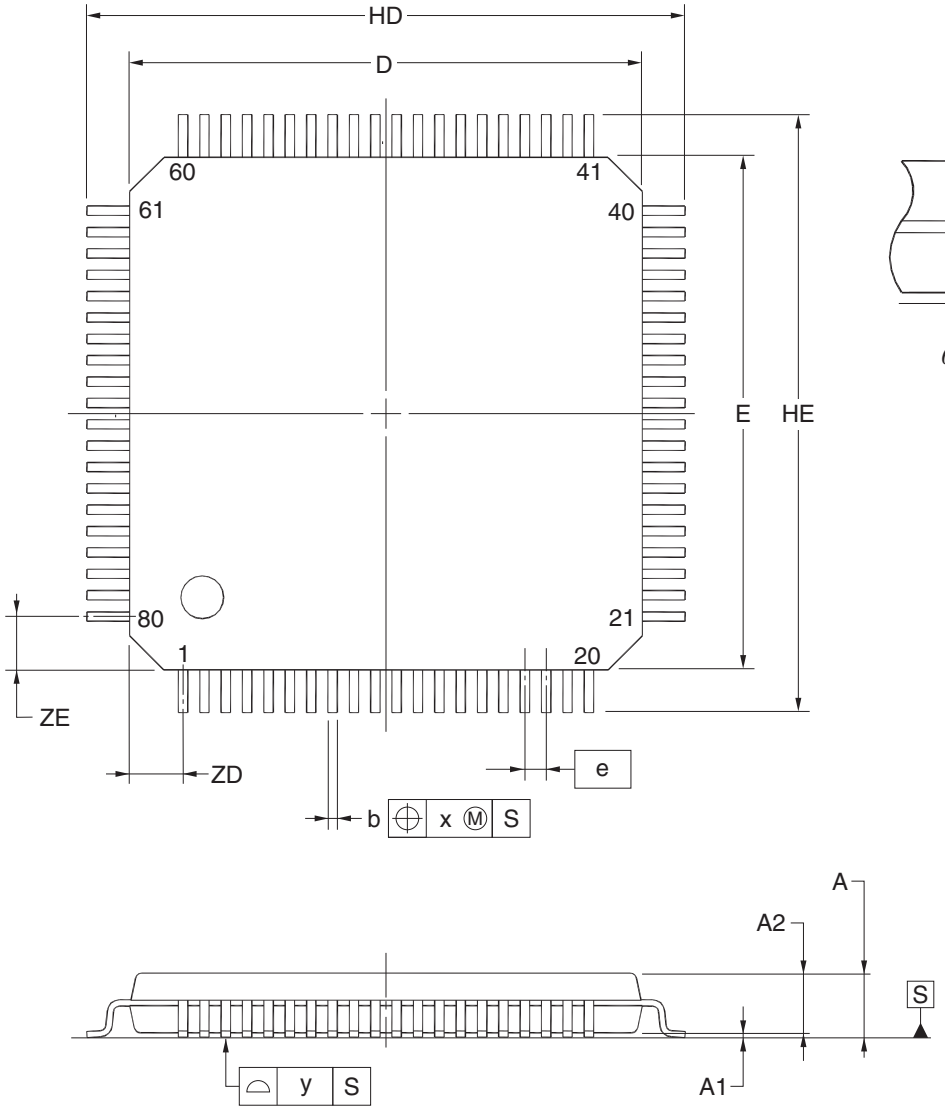
| ITEM | DIMENSIONS |
|---|---|
| D | 10.00±0.20 |
| E | 10.00±0.20 |
| HD | 12.00±0.20 |
| HE | 12.00±0.20 |
| A | 1.60 MAX. |
| A1 | 0.10±0.05 |
| A2 | 1.40±0.05 |
| A3 | 0.25 |
| b | 0.22±0.05 |
| c | $0.145^{+0.055}_{-0.045}$ |
| L | 0.50 |
| Lp | 0.60±0.15 |
| L1 | 1.00±0.20 |
| θ | $3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$ |
| e | 0.50 |
| x | 0.08 |
| y | 0.08 |
| ZD | 1.25 |
| ZE | 1.25 |

**NOTE**
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

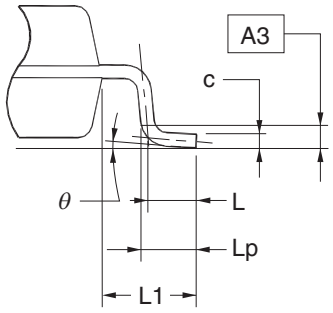
©2012 Renesas Electronics Corporation. All rights reserved.

## 37.6 80-pin products

| JEITA Package Code | RENESAS Code | Previous Code | MASS (TYP.) [g] |
|---|---|---|---|
| P-LFQFP80-12x12-0.50 | PLQP0080KE-A | P80GK-50-8EU-2 | 0.53 |




detail of lead end




(UNIT:mm)

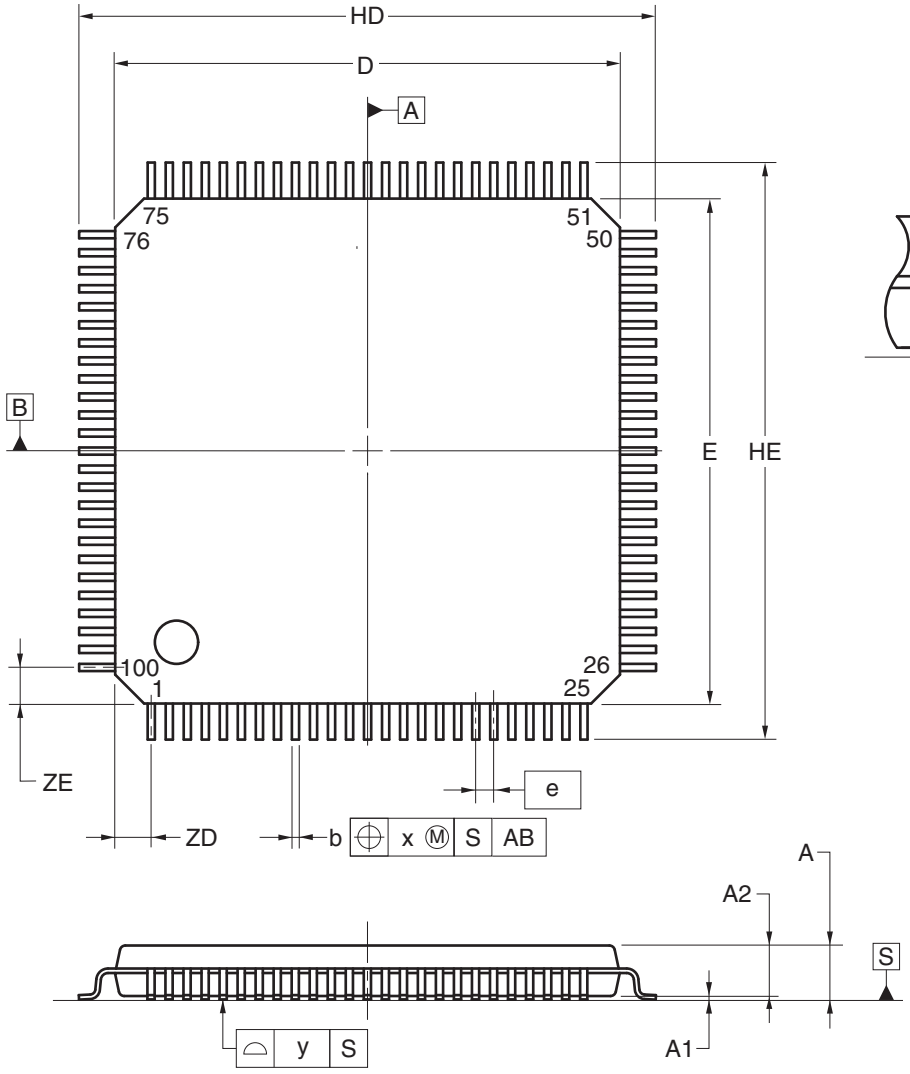
| ITEM | DIMENSIONS |
|---|---|
| D | 12.00±0.20 |
| E | 12.00±0.20 |
| HD | 14.00±0.20 |
| HE | 14.00±0.20 |
| A | 1.60 MAX. |
| A1 | 0.10±0.05 |
| A2 | 1.40±0.05 |
| A3 | 0.25 |
| b | 0.22±0.05 |
| c | $0.145^{+0.055}_{-0.045}$ |
| L | 0.50 |
| Lp | 0.60±0.15 |
| L1 | 1.00±0.20 |
| θ | $3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$ |
| e | 0.50 |
| x | 0.08 |
| y | 0.08 |
| ZD | 1.25 |
| ZE | 1.25 |

**NOTE**
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

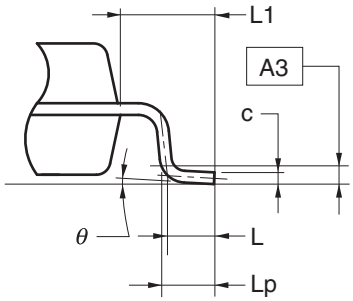
©2012 Renesas Electronics Corporation. All rights reserved.

## 37.7 100-pin products

| JEITA Package Code | RENESAS Code | Previous Code | MASS (TYP.) [g] |
|---|---|---|---|
| P-LFQFP100-14x14-0.50 | PLQP0100KE-A | P100GC-50-GBR-1 | 0.69 |




detail of lead end




(UNIT:mm)

| ITEM | DIMENSIONS |
|---|---|
| D | 14.00±0.20 |
| E | 14.00±0.20 |
| HD | 16.00±0.20 |
| HE | 16.00±0.20 |
| A | 1.60 MAX. |
| A1 | 0.10±0.05 |
| A2 | 1.40±0.05 |
| A3 | 0.25 |
| b | 0.22±0.05 |
| c | $0.145^{+0.055}_{-0.045}$ |
| L | 0.50 |
| Lp | 0.60±0.15 |
| L1 | 1.00±0.20 |
| θ | $3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$ |
| e | 0.50 |
| x | 0.08 |
| y | 0.08 |
| ZD | 1.00 |
| ZE | 1.00 |

©2012 Renesas Electronics Corporation. All rights reserved.

# APPENDIX A RELATED PRODUCTS

## A.1 List of Analog and Power Devices

(1/2)

| Field of application | Analog and power device | Related product | Feature |
|---|---|---|---|
| Lightning | Lamp drive IPD<br>LED headlight power switch | μPD166009T1F | 40 V/10 mΩ, protection function, overcurrent sensor, battery reverse connection protection, low power supply voltage off hold |
| | | μPD166010T1F | 40 V/10 mΩ, protection function, high-precision load current sensor, battery reverse connection protection, low power supply voltage off hold |
| | | μPD166011T1J | 40 V/25 mΩ, dual channel, protection function, high-precision load current sensor, low power supply voltage off hold |
| | | μPD166013T1J | 40 V/60 mΩ, dual channel, protection function, high-precision load current sensor, low power supply voltage off hold |
| | | μPD166014T1K | 40 V/60 mΩ, quad channel, protection function, high-precision load current sensor, low power supply voltage off hold |
| | | μPD166017T1F | 40 V/6 mΩ, protection function, high-precision load current sensor, battery reverse connection protection, low power supply voltage off hold |
| | | μPD166020T1F | 42 V/10 mΩ, protection function, high-precision load current sensor, battery reverse connection protection, low power supply voltage off hold |
| | | μPD166021T1F | 42V/10mΩ, protection function, high-precision load current sensor, battery reverse connection protection, low power supply voltage off hold, overheating shutoff latch |
| | Interior lamp drive thermal FET | HAF2017 | 60 V/43 mΩ, LDPAK, overheating protection function |
| | | HAF2011 | 60 V/20 mΩ, LDPAK, overheating protection function |
| | | HAF2007 | 60 V/75 mΩ, DPAK, overheating protection function |
| | Voltage step-up MOSFET | NP22N055SLE | 55 V/37 mΩ, Ciss = 1100 pF, TO-252 |
| | | NP32N055SLE | 55 V/24 mΩ, Ciss = 2000 pF, TO-252 |
| | | NP40N055KLE | 55 V/23 mΩ, Ciss = 1950 pF, TO-263 |
| | | NP40N10VDF | 100 V/26 mΩ, Ciss = 2400 pF, TO-252 |
| | | NP70N10KUF | 100 V/17 mΩ, Ciss = 3750 pF Typ.@10 V, TO-263 |
| | Battery reverse connection protection/<br>Abnormal current cutoff MOSFET | NP36P04SDG | 40 V/17 mΩ, TO-252 |
| | | NP36P04KDG | 40 V/17 mΩ, TO-263 |
| | Battery reverse connection protection MOSFET | NP55N04SUG | 40 V/6.5 mΩ, TO-252 |
| | | NP55N03SUG | 30 V/5.0 mΩ, TO-252 |
| | Flyback converter drive MOSFET | NP82N10PUF | 82 V/15 mΩ, TO-263 |
| | | NP70N10KUF | 100 V/20 mΩ, TO-263 |

For the sales situation and detailed specifications, contact your local sales representatives.

(2/2)

| Field of application | Analog and power device | Related product | Feature |
|---|---|---|---|
| Motor | Pre-driver and power IC for power door | R2A25111KFP | On-chip 3-phase pre-driver: Drive capacity Ciss = 10000 pF, dead time function, etc., 48-pin LQFP |
| | | R2A25108KFP | On-chip 3-phase pre-driver: Drive capacity Ciss = 10000 pF, current sense amplifier, motor position detection, dead time function, etc., 48-pin LQFP |
| | Motor drive MOSFET for power window, wiper, power door | NP60N04KUG | 40 V/60 A/6.1 mΩ/TO-263 |
| | | NP60N055KUG | 55 V/60 A/9.4 mΩ/TO-263 |
| | | NP82N04PLG | 60 V/82 A/6.7 mΩ/TO-263 |
| | | NP90N04VUG | 40 V/90 A/4.0 mΩ/TO-252 |
| | | NP52N055SUG | 55 V/52 A/14 mΩ/TO-252 |
| | | NP55N055SUG | 55 V/55 A/10 mΩ/TO-252 |
| | | R2J25953 | H-bridge driver: Pch high-side: 16 mΩ max / Nch low-side: 11 mΩ max, Iout= 50 A max, HSOP36 |
| | Motor drive MOSFET for mirror | uPA2793GR | 40 V/7 A/52 mΩ/SOP8 |
| | | uPA2794GR | 60 V/5.5 A/97 mΩ/SOP8 |
| | | RJM0306JSP | 30 V/3.5 A/210 mΩ/SOP8 |
| | | 2SK3408 | For mechanical relay drive: 43±5 V/1.0 A/195 mΩ/SC96 |
| | Motor drive MOSFET for blower motor | NP80N06MLG | 60 V, 80 A, 10 mΩ, TO-220 |
| | | NP82N06MLG | 60 V, 82 A, 7.4 mΩ, TO-220 |
| | | 2SK3755 | 40 V, 45 A, 12 mΩ, Isolated TO-220 |
| | | 2SK4144 | 60 V, 70 A, 5.8 mΩ, Isolated TO-220 |
| Seat Heater | Heater driver IPD | μPD166009T1F | 40 V/10 mΩ, protection function, overcurrent sensor, battery reverse connection protection, low power supply voltage off hold |
| | | μPD166010T1F | 40 V/10 mΩ, protection function, high-precision load current sensor, battery reverse connection protection, low power supply voltage off hold |
| | | μPD166017T1F | 40 V/6 mΩ, protection function, high-precision load current sensor, battery reverse connection protection, low power supply voltage off hold |
| | | μPD166011T1J | 40 V/25 mΩ, dual channel, protection function, high-precision load current sensor, low power supply voltage off hold |
| | | μPD166013T1J | 40 V/60 mΩ, dual channel, protection function, high-precision load current sensor, low power supply voltage off hold |
| | | μPD166014T1K | 40 V/60 mΩ, quad channel, protection function, high-precision load current sensor, low power supply voltage off hold |
| | | μPD166017T1F | 40 V/6 mΩ, protection function, high-precision load current sensor, battery reverse connection protection, low power supply voltage off hold |
| | | μPD166020T1F | 42 V/10 mΩ, protection function, high-precision load current sensor, battery reverse connection protection, low power supply voltage off hold |
| | | μPD166021T1F | 42 V/10 mΩ, protection function, high-precision load current sensor, battery reverse connection protection, low power supply voltage off hold, overheating shutoff latch |
| CAN | CAN line ESD protection | NNCD-ST series | Bidirectional ESD protection |
| | ESD protection | NNCD-DA series (200 mW) RD-FS series (1 W) | Surge protection |

For the sales situation and detailed specifications, contact your local sales representatives.

# APPENDIX B REVISION HISTORY

## Major Revisions in This Edition

(1/3)

| Page | Description | Classification |
|---|---|---|
| CHAPTER 1 OVERVIEW | | |
| p.10 to 31 | Figure 1-1 to 1-22 Block Diagram, added other products | (c) |
| CHAPTER 3 CPU ARCHITECTURE | | |
| p.147 | Table 3-5 SFR List (1/4), corrected After Reset value of P13 register | (c) |
| CHAPTER 4 PORT FUNCTIONS | | |
| p.256 | 4.2.7 Port 7, corrected PMC7 register | (a) |
| p.271 | Table 4-14 Setting Functions of P80/ANI2/ANO0 Pin, corrected | (c) |
| p.321 | Figure 4-75 Format of Port Register, corrected After Reset value of P13 register | (c) |
| p.325 | Figure 4-79 Format of Port Mode Control Register, added Note of PMC7 register | (c) |
| CHAPTER 5 CLOCK GENERATOR | | |
| p.389 | Figure 5-18 Format of LIN Clock Select Register (LINCKSEL), added Caution 3 | (c) |
| p.423 | 5.7.2 High-Speed On-Chip Oscillator, corrected high-speed on-chip oscillator frequency | (a) |
| CHAPTER 6 TIMER ARRAY UNIT | | |
| p.438 | 6.2.2 Timer data register mn (TDRmn), added description | (c) |
| CHAPTER 8 TIMER RD | | |
| p.569 | Table 8-2 Timer RD Register Configuration, corrected Access Size of TRDCR1 register | (a) |
| p.602 | 8.2.19 Timer RD General Registers Ai, Bi, Ci, and Di (TRDGRAi, TRDGRBi, TRDGRCi, TRDGRDi) [Complementary PWM Mode], deleted TRDDF0 and TRDDF1 register | (a) |
| p.621 | Figure 8-44 Pulse Output Forced Cutoff, corrected INTP0 circuit | (a) |
| CHAPTER 9 REAL-TIME CLOCK | | |
| p.666 | Figure 9-8 Format of Real-time Clock Control Register 1 (RTCC1), added Note 1 and 2 of RWAIT bit | (b) |
| CHAPTER 11 WATCHDOG TIMER | | |
| p.697 | Table 11-3 Setting of Overflow Time of Watchdog Timer, added Note | (b) |
| CHAPTER 12 A/D CONVERTER | | |
| p.703 | 12.2 (1) ANI0 to ANI23 ($V_{DD}$) and ANI24 to ANI30 ($EV_{DD}$) pins, corrected Note | (a) |
| p.730 | Figure 12-18 Formats of Port Mode Control Registers 7, 9 and 12, added Note of PMC7 register | (c) |
| CHAPTER 13 D/A CONVERTER (RL78/F14 Only) | | |
| p.772 | Table 13-1 Setting Functions of ANO0/ANI2/P80 Pin, corrected | (c) |
| CHAPTER 15 SERIAL ARRAY UNIT | | |
| p.792,833,895, 959,992 | Note of Tables “Group A, B, C-1, C-2, D-1, D-2 and E” | (b) |
| p.807 | Note of Figure 15-7. Format of Serial Communication Operation Setting Register mn (SCRmn) | (b) |
| p.821 | Note of Figure 15-17. Format of Serial Slave Select Enable Register m (SSEm) | (b) |

Remark: “Classification” in the above table classifies revisions as follows.

(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,

(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(2/3)

| Page | Description | Classification |
|---|---|---|
| CHAPTER 17 LIN/UART MODULE (RLIN3) | | |
| p.1112,1140 ,1168 | LIN Clock Select Register (LINCKSEL), added Note of LINnMCK | (c) |
| p.1123,1124 ,1150 | LIN/UART Error Detection Enable Register (LEDEn), added description of (Bit Error) and (Framing Error) | (c) |
| p.1153 | LIN/UART Transmission Control Register (LTRCn), deleted description of FTS bit | (c) |
| p.1160 | LIN/UART Data Field Configuration Register (LDFCn), corrected description of RFDL bits | (a) |
| p.1176 | LIN/UART Space Configuration Register (LSCn), added description of IBS bits | (c) |
| p.1181 | LIN/UART Status Register (LSTn), deleted description of FTC bit | (c) |
| p.1189 | UART Operation Enable Register (LUOERn), added description of UROE bit | (c) |
| p.1216,1217 | Table 17-14, Table 17-15 Types of Statuses in LIN Master Mode / LIN Slave Mode, and added Note 1 | (c) |
| p.1218,1221 | Table 17-16, Table 17-17 Types of Error Statuses in LIN Master Mode / LIN Slave Mode, corrected Notes 1 and 3 | (c) |
| p.1231 | Figure 17-30 Expansion Bit Reception Example (with Data Comparison), added Note | (c) |
| p.1246 to 1249 | 17.6.2 to 17.6.5 LIN Self-Test Mode, added description when LIN self-test mode is canceled | (c) |
| p.1251,1253 ,1255 | Figure 17-41, Figure 17-42, Figure 17-43 Block Diagram of Baud Rate Generator, corrected of LIN communications clock source | (b) |
| CHAPTER 18 CAN INTERFACE (RS-CAN LITE) | | |
| p.1295,1305, 1327,1342, 1372 | 18.3.7 CiERFLL, 18.3.14 GERFLL, 18.3.35 RFSTSm, 18.3.47 CFSTSk, 18.3.75 THLSTSi, corrected Note | (c) |
| p.1340 | 18.3.46 CANi Transmit/Receive FIFO Control Register kH (CFCCHk), corrected description of CFITR bit | (a) |
| CHAPTER 19 DTC | | |
| p.1461 | 19.4.3 DTC Pending Instruction, added multiply, divide, and multiply-accumulate instructions | (c) |
| CHAPTER 21 INTERRUPT FUCTIONS | | |
| p.1479 | Table 21-1 Interrupt Source List (3/4), corrected Note 5 | (a) |
| p.1512 | 21.4.4 Interrupt servicing during division instruction, added Caution | (b) |
| p.1514 | 21.4.5 Interrupt request hold, added divide, and multiply-accumulate instructions | (c) |
| CHAPTER 27 SAFETY FUNCTIONS | | |
| p.1591 | Figure 27-9 Format of 1-bit Error Detection Interrupt Enable Register (ECCIER), added Note 2 | (c) |
| p.1596 | 27.3.4 CPU stack pointer monitor function, added Caution | (b) |
| p.1601 | Figure 27-20 Format of Invalid Memory Access Detection Control Register (IAWCTL), deleted Note 2 | (c) |
| CHAPTER 33 INSTRUCTION SET | | |
| p.1672 | Table 33-5 Operation List (12/18), added Caution | (b) |
| CHAPTER 34 ELECTRICAL SPECIFICATIONS (GRADE L) | | |
| p.1686,1687 | 34.3.1 Pin Characteristics (1/4) and (2/4), corrected Note 1 | (c) |
| p.1697 | 34.4.1 Basic Operation (1/2), added RESET pin of Note | (c) |
| p.1708 | 34.5.1 Serial Array Unit (7) During communication at same potential (simplified $I^2C$ mode), corrected Conditions on Table | (a) |
| p.1729 | 34.9 Flash Memory Programming Characteristics, corrected Conditions at Erase time | (c) |

Remark: “Classification” in the above table classifies revisions as follows.

(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,

(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(3/3)

| Page | Description | Classification |
|---|---|---|
| CHAPTER 35 ELECTRICAL SPECIFICATIONS (GRADE K) | | |
| p.1737,1738 | 35.3.1 Pin Characteristics (1/4) and (2/4), corrected Note 1 | (c) |
| p.1748 | 35.4.1 Basic Operation (1/2), added RESET pin of Note | (c) |
| p.1759 | 35.5.1 Serial Array Unit (7) During communication at same potential (simplified $I^2C$ mode), corrected Conditions on Table | (a) |
| p.1780 | 35.9 Flash Memory Programming Characteristics, corrected Conditions at Erase time | (c) |
| CHAPTER 36 ELECTRICAL SPECIFICATIONS (GRADE Y) | | |
| p.1783 | 36.1 Absolute Maximum Ratings (2/2), corrected P106 pin name of Output current low ($I_{OL1}$) | (a) |
| p.1788,1789 | 36.3.1 Pin Characteristics (1/4) and (2/4), corrected Note 1 | (c) |
| p.1799 | 36.4.1 Basic Operation (1/2), added RESET pin of Note | (c) |
| p.1810 | 36.5.1 Serial Array Unit (7) During communication at same potential (simplified $I^2C$ mode), corrected Conditions on Table | (a) |
| p.1831 | 36.9 Flash Memory Programming Characteristics, corrected Conditions at Erase time | (c) |

Remark: “Classification” in the above table classifies revisions as follows.

(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,

(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

RL78/F13, F14 User’s Manual: Hardware

Publication Date: Rev.2.10 Dec 10, 2015

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.**
2801 Scott Boulevard Santa Clara, CA 95050-2549, 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: +49-211-6503-0, Fax: +49-211-6503-1327

**Renesas Electronics (China) Co., Ltd.**
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

**Renesas Electronics (Shanghai) Co., Ltd.**
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

**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: +852-2265-6688, Fax: +852 2886-9022/9044

**Renesas Electronics Taiwan Co., Ltd.**
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

**Renesas Electronics Singapore Pte. Ltd.**
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

**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.**
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141



© 2015 Renesas Electronics Corporation. All rights reserved.


Colophon 3.0

RL78/F13, F14

Renesas Electronics Corporation

R01UH0368EJ0210

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Renesas Electronics:

R5F10BGGLNA#U5 R5F10PBELNA#U5