User's Manual

16

RL78/F22, F25

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.1.20 Jun 2025

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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended 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; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.

© 2025 Renesas Electronics Corporation. All rights reserved.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can 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 must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. 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. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. 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 V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, 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 V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the RL78/F22, F25 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/F22, F25 manual is separated into two parts: this manual and the software edition (common to the RL78 family).

RL78/F22, F25 User's Manual Hardware	**RL78 family User's Manual Software**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • 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/F22, F25 Microcontroller instructions:
  → Refer to the separate document **RL78 Family User's Manual: Software**.

Conventions

Data significance:	Higher digits on the left and lower digits on the right	
Active low representations:	$\overline{\times\times\times}$ (overscore over 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/F22, F25 User's Manual: Hardware	R01UH1061E
RL78 Family User's Manual: Software	R01US0015E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP6 Flash Memory Programmer User's Manual	R20UT4469E
E2 Emulator User's Manual	R20UT3538E
E2 Lite Emulator User's Manual	R20UT3240E
Renesas Flash Programmer Flash Memory Programming Software User's Manual	R20UT5190E

Other Documents

Document Name	Document No.
Renesas RL78 Family Microcontrollers	R01CP0003E
Semiconductor Package Mount Manual	R50ZZ0003E
Semiconductor Reliability Handbook	R51ZZ0001E

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.
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/F25 Functions List 4

1.3.2 RL78/F22 Functions List 6

1.4 Block Diagram 8

1.4.1 RL78/F25: Block Diagram of R7F125FPL 100-pin Products 8

1.4.2 RL78/F25: Block Diagram of R7F125FML 80-pin Products 9

1.4.3 RL78/F25: Block Diagram of R7F125FLL 64-pin Products 10

1.4.4 RL78/F25: Block Diagram of R7F125FGL 48-pin Products 11

1.4.5 RL78/F22: Block Diagram of R7F122FGG 48-pin Products 12

1.4.6 RL78/F22: Block Diagram of R7F122FBG 32-pin Products 13

1.4.7 RL78/F22: Block Diagram of R7F122F7G 24-pin Products 14

1.5 Pin Configurations 15

1.5.1 RL78/F25 Pin Configuration for 100-pin Products 15

1.5.2 RL78/F25 Pin Configuration for 80-pin Products 16

1.5.3 RL78/F25 Pin Configuration for 64-pin Products 17

1.5.4 RL78/F25 Pin Configuration for 48-pin Products 18

1.5.5 RL78/F22 Pin Configuration for 48-pin Products 19

1.5.6 RL78/F22 Pin Configuration for 32-pin Products 20

1.5.7 RL78/F22 Pin Configuration for 24-pin Products 21

CHAPTER 2 PIN FUNCTIONS 22

2.1 Pin Function List 22

2.1.1 RL78/F25 100-pin Products 23

2.1.2 RL78/F22 48-pin Products 25

2.1.3 Pins for Each Product (Pins Other Than Port Pins) 27

2.2 Description of Pin Functions 39

2.2.1 P00 to P03 (Port 0) 39

2.2.2 P10 to P17 (Port 1) 40

2.2.3 P30 to P32 (Port 3) 42

2.2.4 P40 to P47 (Port 4) 43

2.2.5 P50 to P57 (Port 5) 45

2.2.6 P60 to P67 (Port 6) 47

2.2.7 P70 to P77 (Port 7) 49

2.2.8 P80 to P87 (Port 8) 51

2.2.9 P90 to P97 (Port 9) 52

2.2.10 P100 to P107 (Port 10) 53

2.2.11 P120 to P127 (Port 12) 54

2.2.12 P130, P137 (Port 13) 56

2.2.13 P140 (Port 14) 57

2.2.14 P150 to P157 (Port 15) 58

2.2.15 V_{DD}, AV_{DD}, EV_{DD0}, EV_{DD1}, V_{SS}, AV_{SS}, EV_{SS0}, EV_{SS1} 59

2.2.16 RESET 59

2.2.17 REGC 59
2.3 Recommended Connection of Unused Pins 60

CHAPTER 3 CPU ARCHITECTURE 66

3.1 Memory Space 66
3.1.1 Internal Program Memory Space 72
3.1.2 Mirror Area 76
3.1.3 Internal Data Memory Space 78
3.1.4 Special Function Register (SFR) Area 79
3.1.5 Extended Special Function Register (2nd SFR: 2nd Special Function Register) Area 79
3.1.6 Data Memory Addressing 79
3.2 Processor Registers 82
3.2.1 Control Registers 82
3.2.2 General-purpose Registers 85
3.2.3 ES and CS Registers 87
3.2.4 Special Function Registers (SFRs) 88
3.2.5 Extended Special Function Registers (2nd SFRs: 2nd Special Function Registers) 90
3.3 Instruction Address Addressing 93
3.3.1 Relative Addressing 93
3.3.2 Immediate Addressing 93
3.3.3 Table Indirect Addressing 94
3.3.4 Register Indirect Addressing 95
3.4 Addressing for Processing Data Addresses 96
3.4.1 Implied Addressing 96
3.4.2 Register Addressing 96
3.4.3 Direct Addressing 97
3.4.4 Short Direct Addressing 98
3.4.5 SFR Addressing 99
3.4.6 Register Indirect Addressing 100
3.4.7 Based Addressing 101
3.4.8 Based Indexed Addressing 105
3.4.9 Stack Addressing 106
3.5 Register Descriptions 110
3.5.1 RAM Guard Function 111
3.5.2 SFR Guard Function 113

CHAPTER 4 PORT FUNCTIONS 114

4.1 Port Functions 114
4.2 Port Configuration 115
4.2.1 Port 0 116
4.2.2 Port 1 121
4.2.3 Port 3 138
4.2.4 Port 4 143
4.2.5 Port 5 153
4.2.6 Port 6 163
4.2.7 Port 7 176
4.2.8 Port 8 190
4.2.9 Port 9 200
4.2.10 Port 10 203

4.2.11 Port 12 ... 208
4.2.12 Port 13 ... 217
4.2.13 Port 14 ... 220
4.2.14 Port 15 ... 222
4.3 Registers Controlling Port Function ... 231
4.3.1 Port Mode Registers (PMm) ... 238
4.3.2 Port Registers (Pm) ... 239
4.3.3 Pull-up Resistor Option Registers (PUm) ... 241
4.3.4 Port Input Mode Registers (PIMm) ... 242
4.3.5 Port Output Mode Registers (POMm) ... 243
4.3.6 Port Mode Control Registers (PMCm) ... 244
4.3.7 Touch Sensor Port Mode Control Register (TSPMCm) ... 245
4.3.8 Port Input Threshold Control Registers (PITHLm) ... 246
4.3.9 Peripheral I/O Redirection Register 0 (PIOR0) ... 248
4.3.10 Peripheral I/O Redirection Register 1 (PIOR1) ... 249
4.3.11 Peripheral I/O Redirection Register 2 (PIOR2) ... 250
4.3.12 Peripheral I/O Redirection Register 3 (PIOR3) ... 251
4.3.13 Peripheral I/O Redirection Register 4 (PIOR4) ... 252
4.3.14 Peripheral I/O Redirection Register 5 (PIOR5) ... 254
4.3.15 Peripheral I/O Redirection Register 6 (PIOR6) ... 255
4.3.16 Peripheral I/O Redirection Register 7 (PIOR7) ... 256
4.3.17 Peripheral I/O Redirection Register 8 (PIOR8) ... 257
4.3.18 Peripheral I/O Redirection Register 9 (PIOR9) ... 258
4.3.19 Port Output Slew Rate Register (PSRSEL) ... 259
4.3.20 SNOOZE Status Output Control Register 0 (PSNZCNT0) ... 260
4.3.21 SNOOZE Status Output Control Register 1 (PSNZCNT1) ... 261
4.3.22 SNOOZE Status Output Control Register 2 (PSNZCNT2) ... 262
4.3.23 SNOOZE Status Output Control Register 3 (PSNZCNT3) ... 263
4.3.24 Port Mode Select Register (PMS) ... 264
4.4 Port Function Operations ... 265
4.4.1 Writing to I/O Port ... 265
4.4.2 Reading from I/O Port ... 265
4.4.3 Operations on I/O Port ... 265
4.4.4 Connecting to External Device with Different Potential (3 V) ... 266
4.5 Setting of Port Mode Register and Output Latch When Using Alternate Function ... 269
4.6 Cautions When Using Port Function ... 278
4.6.1 Cautions on 1-bit Manipulation Instruction for Port Register n (Pm) ... 278
4.6.2 Notes on Specifying the Pin Settings ... 278
4.6.3 Cautions When Setting Port Related Registers ... 279
4.6.4 Cautions When Setting Peripheral I/O Redirection Register ... 279

CHAPTER 5 CLOCK GENERATOR ... 280

5.1 Functions of Clock Generator ... 280
5.2 Configuration of Clock Generator ... 283
5.3 Registers Controlling Clock Generator ... 287
5.3.1 Clock Operation Mode Control Register (CMC) ... 288
5.3.2 System Clock Control Register (CKC) ... 290
5.3.3 Clock Operation Status Control Register (CSC) ... 292
5.3.4 Oscillation Stabilization Time Counter Status Register (OSTC) ... 294

5.3.5 Oscillation Stabilization Time Select Register (OSTS) ... 296
5.3.6 Peripheral Enable Registers 0, 1, 2 (PER0, PER1, PER2) ... 297
5.3.7 Operation Speed Mode Control Register (OSMC) ... 304
5.3.8 High-Speed On-chip Oscillator Frequency Select Register (HOCODIV) ... 305
5.3.9 High-Speed On-chip Oscillator Trimming Register (HIOTRM) ... 306
5.3.10 CAN Clock Select Register (CANCKSEL) ... 307
5.3.11 LIN Clock Select Register (LINCKSEL) ... 308
5.3.12 Clock Select Register (CKSEL) ... 309
5.3.13 PLL Control Register (PLLCTL) ... 311
5.3.14 PLL Status Register (PLLSTS) ... 314
5.3.15 f_{MP} Clock Division Register (MDIV) ... 315
5.3.16 A/D Conversion Clock Control Register (ADCKS) ... 316
5.4 System Clock Oscillator ... 317
5.4.1 X1 Oscillator ... 317
5.4.2 XT1 Oscillator ... 317
5.4.3 High-Speed On-chip Oscillator ... 321
5.4.4 PLL Circuit ... 321
5.4.5 Low-Speed On-chip Oscillator ... 321
5.4.6 WDT-Dedicated Low-Speed On-chip Oscillator ... 321
5.5 Clock Generator Operation ... 322
5.6 Controlling Clock ... 324
5.6.1 Example of Setting High-Speed On-chip Oscillator ... 324
5.6.2 Example of Setting X1 Oscillator ... 326
5.6.3 Example of Setting XT1 Oscillator ... 327
5.6.4 Examples of Setting PLL Circuit ... 328
5.6.5 Example of Setting Low-Speed On-chip Oscillator ... 330
5.6.6 CPU Clock Status Transition Diagram ... 331
5.6.7 Conditions Before Changing CPU Clock and Processing After Changing CPU Clock ... 335
5.6.8 Time Required for Switchover of CPU Clock, Main System/PLL Select Clock, and Main System Clock ... 342
5.6.9 Conditions Before Clock Oscillation is Stopped ... 344
5.7 Usage Notes ... 345
5.7.1 CPU/Peripheral Hardware Clock ... 345
5.7.2 High-Speed On-chip Oscillator ... 345

CHAPTER 6 TIMER ARRAY UNIT ... 346

6.1 Functions of Timer Array Unit ... 348
6.1.1 Independent Channel Operation Function ... 348
6.1.2 Simultaneous Channel Operation Function ... 349
6.1.3 8-bit Timer Operation Function (Channels 1 and 3 Only) ... 350
6.1.4 LIN-bus Supporting Function (Channel 7 of Unit 0 Only) ... 351
6.2 Configuration of Timer Array Unit ... 352
6.2.1 Timer Count Register mn (TCRmn) ... 358
6.2.2 Timer Data Register mn (TDRmn) ... 360
6.3 Registers Controlling Timer Array Unit ... 361
6.3.1 Peripheral Enable Register 0 (PER0) ... 363
6.3.2 Timer Clock Select Register m (TPSm) ... 364
6.3.3 Timer Mode Register mn (TMRmn) ... 367
6.3.4 Timer Status Register mn (TSRmn) ... 373

6.3.5 Timer Channel Enable Status Register m (TEm) ... 374
6.3.6 Timer Channel Start Register m (TSm) ... 375
6.3.7 Timer Channel Stop Register m (TTm) ... 376
6.3.8 Timer Input Select Register 0 (TIS0) ... 377
6.3.9 Timer Input Select Register 1 (TIS1) ... 378
6.3.10 Timer Input Select Register 2 (TIS2) ... 379
6.3.11 Timer Output Enable Register m (TOEm) ... 380
6.3.12 Timer Output Register m (TOm) ... 381
6.3.13 Timer Output Level Register m (TOLm) ... 382
6.3.14 Timer Output Mode Register m (TOMm) ... 383
6.3.15 Noise Filter Enable Registers 1, 2 (NFEN1, NFEN2) ... 384
6.3.16 Port Mode Registers 0, 1, 3, 4, 5, 6, 7, 12 (PM0, PM1, PM3, PM4, PM5, PM6, PM7, PM12) ... 387
6.3.17 PWM Output Delay Control Register 1 (PWMDLY1) ... 389
6.3.18 PWM Output Delay Control Register 2 (PWMDLY2) ... 390
6.3.19 Unit Select Register (UTSEL) ... 391
6.4 Basic Rules of Timer Array Unit ... 392
6.4.1 Basic Rules of Simultaneous Channel Operation Function ... 392
6.4.2 Basic Rules of 8-bit Timer Operation Function (Channels 1 and 3 Only) ... 394
6.5 Operation Timing of Counter ... 395
6.5.1 Count Clock (f_{TCLK}) ... 395
6.5.2 Start Timing of Counter ... 397
6.5.3 Operation of Counter ... 398
6.6 Channel Output (TOmn pin) Control ... 403
6.6.1 TOmn Pin Output Circuit Configuration ... 403
6.6.2 TOmn Pin Output Setting ... 404
6.6.3 Cautions on Channel Output Operation ... 405
6.6.4 Collective Manipulation of TOmn Bit ... 410
6.6.5 Timer Interrupt and TOmn Pin Output at Operation Start ... 411
6.7 Independent Channel Operation Function of Timer Array Unit ... 412
6.7.1 Operation as Interval Timer/Square Wave Output ... 412
6.7.2 Operation as External Event Counter ... 418
6.7.3 Operation as Frequency Divider ... 423
6.7.4 Operation as Input Pulse Interval Measurement ... 427
6.7.5 Operation as Input Signal High-/Low-level Width Measurement ... 432
6.7.6 Operation as Delay Counter ... 437
6.8 Simultaneous Channel Operation Function of Timer Array Unit ... 442
6.8.1 Operation as One-shot Pulse Output Function ... 442
6.8.2 Operation as PWM Function ... 450
6.8.3 Operation as Multiple PWM Output Function ... 457
6.9 Cautions When Using Timer Array Unit ... 466
6.9.1 Cautions When Using Timer Output ... 466

CHAPTER 7 TIMER RJ ... 467

7.1 Overview ... 467
7.2 I/O Pins ... 468
7.3 Registers ... 469
7.3.1 Peripheral Enable Register 1 (PER1) ... 469
7.3.2 Operation Speed Mode Control Register (OSMC) ... 470
7.3.3 Clock Select Register (CKSEL) ... 470

7.3.4 Timer RJ Counter Register 0 (TRJ0), Timer RJ Reload Register 471
7.3.5 Timer RJ Control Register 0 (TRJCR0) 472
7.3.6 Timer RJ I/O Control Register 0 (TRJIOC0) 474
7.3.7 Timer RJ Mode Register 0 (TRJMR0) 476
7.3.8 Timer RJ Event Pin Select Register 0 (TRJISR0) 477
7.3.9 Port Mode Registers 1, 4 (PM1, PM4) 478
7.4 Operation 479
7.4.1 Reload Register and Counter Rewrite Operation 479
7.4.2 Timer Mode 480
7.4.3 Pulse Output Mode 481
7.4.4 Event Counter Mode 482
7.4.5 Pulse Width Measurement Mode 484
7.4.6 Pulse Period Measurement Mode 485
7.4.7 Coordination with Event Link Controller (ELC) 486
7.4.8 Output Settings for Each Mode 486
7.5 Notes on Timer RJ 487
7.5.1 Count Operation Start and Stop Control 487
7.5.2 Access to Flags (Bits TEDGF and TUNDF in TRJCR0 Register) 487
7.5.3 Access to Counter Register 487
7.5.4 When Changing Mode 488
7.5.5 Procedure for Setting Pins TRJO0 and TRJIO0 488
7.5.6 When Timer RJ is not Used 488
7.5.7 When Timer RJ Operating Clock is Stopped 488
7.5.8 Procedure for Setting STOP Mode (Event Counter Mode) 489
7.5.9 Functional Restriction in STOP Mode (Event Counter Mode Only) 489
7.5.10 When Count is Forcibly Stopped by TSTOP Bit 489
7.5.11 Digital Filter 489
7.5.12 When Selecting f_{IL} as Count Source 489

CHAPTER 8 TIMER RDe 490
8.1 Overview 490
8.2 Registers 493
8.2.1 Peripheral enable register 1 (PER1) 495
8.2.2 Clock Select Register (CKSEL) 496
8.2.3 Timer RDe ELC Register (TRDELC) 497
8.2.4 Timer RDe Start Register (TRDSTR) 498
8.2.5 Timer RDe Mode Register (TRDMR) 499
8.2.6 Timer RDe PWM Function Select Register (TRDPMR) 500
8.2.7 Timer RDe Function Control Register (TRDFCR) 501
8.2.8 Timer RDe Output Master Enable Register 1 (TRDOER1) 503
8.2.9 Timer RDe Output Master Enable Register 2 (TRDOER2) 504
8.2.10 Timer RDe Output Control Register (TRDOCR) 505
8.2.11 Timer RDe Digital Filter Function Select Register i (TRDDFi) (i = 0 or 1) 509
8.2.12 Timer RDe Control Register i (TRDCRi) (i = 0 or 1) 511
8.2.13 Timer RDe I/O Control Register Ai (TRDIORAi) (i = 0 or 1) 516
8.2.14 Timer RDe I/O Control Register Ci (TRDIORCi) (i = 0 or 1) 518
8.2.15 Timer RDe Status Register i (TRDSRi) (i = 0 or 1) 520
8.2.16 Timer RDe Interrupt Enable Register i (TRDIERi) (i = 0 or 1) 524
8.2.17 Timer RDe PWM Function Output Level Control Register i (TRDPOCRi) (i = 0 or 1) 525

8.2.18 Timer RDe Counter i (TRDi) (i = 0 or 1) ... 526
8.2.19 Timer RDe General Registers ji (TRDGRji) (i = 0 or 1, j = A, B, C, or D) ... 529
8.2.20 Timer RDe Extended Compare Register m (TRDCMPm) (m = B0, D0, A1, B1, C1, or D1) ... 541
8.2.21 Timer RDe A/D Trigger Compare Register 0 (TRDADTC0) ... 543
8.2.22 Timer RDe A/D Trigger Buffer Register 0 (TRDADTB0) ... 544
8.2.23 Timer RDe A/D Trigger Compare Register 1 (TRDADTC1) ... 545
8.2.24 Timer RDe A/D Trigger Buffer Register 1 (TRDADTB1) ... 546
8.2.25 Timer RDe Reload Trigger Register (TRDRDT01) ... 547
8.2.26 Timer RDe Reload Status Flag Register (TRDRSF01) ... 549
8.2.27 Timer RDe A/D Trigger Control Register (TRDADCR) ... 551
8.2.28 Timer RDe Extended PWM Mode Register i (TRDEMRi) (i = 0 or 1) ... 552
8.2.29 Timer RDe Dithering Number Register i (TRDDNRi) (i = 0 or 1) ... 553
8.2.30 Timer RDe Gate Pattern Register i (TRDGPRi) (i = 0 or 1) ... 554
8.2.31 Timer RDe Dithering / Gate Control Register i (TRDDGCRi) (i = 0 or 1) ... 555
8.2.32 PWM Output Delay Control Register 0 (PWMDLY0) ... 556
8.2.33 Port Mode Registers (PM1, PM3, PM4, PM12, PM14) ... 557
8.3 Operation ... 558
8.3.1 Items Common to Multiple Modes ... 558
8.3.2 Input Capture Function ... 571
8.3.3 Output Compare Function ... 575
8.3.4 PWM Function ... 580
8.3.5 Reset Synchronous PWM Mode ... 584
8.3.6 Complementary PWM Mode ... 587
8.3.7 PWM3 Mode ... 591
8.3.8 Extended PWM Mode ... 594
8.3.9 Extended Complementary PWM Mode ... 607
8.4 Timer RDe Interrupt ... 614
8.5 Notes on Timer RDe ... 615
8.5.1 SFR Read/Write Access ... 615
8.5.2 Mode Switching ... 616
8.5.3 Count Source ... 616
8.5.4 Input Capture Function ... 616
8.5.5 Procedure for Setting Pins TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi (i = 0 or 1) ... 616
8.5.6 External Clock TRDCLK0 ... 617
8.5.7 Timer Mode PWM Function ... 617
8.5.8 Reset Synchronous PWM Mode ... 617
8.5.9 Complementary PWM Mode ... 618
8.5.10 Extended PWM Mode ... 622
8.5.11 Extended Complementary PWM Mode ... 623
8.6 PWM Option Unit A (PWMOPA) ... 627
8.6.1 Overview of PWM Option Unit A ... 628
8.6.2 Registers Controlling PWM Option Unit A ... 628
8.6.3 Operation ... 636
8.7 Interrupt Decimation Module (TRDMBK) ... 658
8.7.1 Overview ... 658
8.7.2 TRDMBK Registers ... 658
8.7.3 Operation ... 661
8.7.4 Setting Procedure ... 663

CHAPTER 9 REAL-TIME CLOCK 664
9.1 Functions of Real-time Clock 664
9.2 Configuration of Real-time Clock 664
9.3 Registers Controlling Real-time Clock 666
9.3.1 Peripheral Enable Register 0 (PER0) 667
9.3.2 Operation Speed Mode Control Register (OSMC) 668
9.3.3 Timer Input Select Register 1 (TIS1) 669
9.3.4 Timer Input Select Register 2 (TIS2) 670
9.3.5 RTC Clock Select Register (RTCCL) 671
9.3.6 Real-time Clock Control Register 0 (RTCC0) 672
9.3.7 Real-time Clock Control Register 1 (RTCC1) 673
9.3.8 Second Count Register (SEC) 675
9.3.9 Minute Count Register (MIN) 675
9.3.10 Hour Count Register (HOUR) 676
9.3.11 Day Count Register (DAY) 678
9.3.12 Week Count Register (WEEK) 679
9.3.13 Month Count Register (MONTH) 680
9.3.14 Year Count Register (YEAR) 680
9.3.15 Watch Error Correction Register (SUBCUD) 681
9.3.16 16-bit Watch Error Correction Register (SUBCUDW) 682
9.3.17 Alarm Minute Register (ALARMWM) 683
9.3.18 Alarm Hour Register (ALARMWH) 683
9.3.19 Alarm Week Register (ALARMWW) 683
9.4 Real-time Clock Operation 685
9.4.1 Starting Operation of Real-time Clock 685
9.4.2 Shifting to HALT/STOP Mode After Starting Operation 686
9.4.3 Reading/Writing Real-time Clock 687
9.4.4 Setting Alarm of Real-time Clock 689
9.4.5 1 Hz Output of Real-time Clock 690
9.4.6 Example of Watch Error Correction of Real-time Clock 691

CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER 694
10.1 Functions of Clock Output/Buzzer Output Controller 694
10.2 Configuration of Clock Output/Buzzer Output Controller 696
10.3 Registers Controlling Clock Output/Buzzer Output Controller 696
10.3.1 Clock Output Select Register 0 (CKS0) 696
10.3.2 Clock Select Register (CKSEL) 698
10.3.3 Port Mode Register 14 (PM14) 699
10.4 Operations of Clock Output/Buzzer Output Controller 700
10.4.1 Operation as Output Pin 700
10.5 Notes on Clock Output/Buzzer Output Controller 700

CHAPTER 11 WATCHDOG TIMER 701
11.1 Functions of Watchdog Timer 701
11.2 Configuration of Watchdog Timer 701
11.3 Register to Controlling Watchdog Timer 703
11.3.1 Watchdog Timer Enable Register (WDTE) 703
11.4 Operation of Watchdog Timer 704
11.4.1 Controlling Operation of Watchdog Timer 704

11.4.2 Setting Overflow Time of Watchdog Timer 705
11.4.3 Setting Window Open Period of Watchdog Timer 706
11.4.4 Setting Watchdog Timer Interval Interrupt 707

CHAPTER 12 12-BIT A/D CONVERTER 708

12.1 Overview 708
12.2 Register Descriptions 712
12.2.1 Peripheral Enable Register 0 (PER0) 715
12.2.2 A/D Data Registers (ADDRy, ADOCDR, ADDRxM) [x = 0 to 7, y = 0 to 29] 716
12.2.3 A/D Self-diagnosis Data Register (ADRD) 718
12.2.4 A/D Control Register (ADCSR) 719
12.2.5 A/D Channel Select Register A (ADANSA0, ADANSA1) 721
12.2.6 A/D Channel Select Register B (ADANSB0, ADANSB1) 722
12.2.7 A/D-converted Value Addition/Average Function Channel Select Register (ADADS0, ADADS1) 723
12.2.8 A/D-converted Value Addition/Average Count Select Register (ADADC) 725
12.2.9 A/D Control Extended Register (ADCER) 726
12.2.10 A/D Conversion Start Trigger Select Register (ADSTRGR) 727
12.2.11 A/D Conversion Extended Input Control Register (ADEXICR) 728
12.2.12 A/D Sampling State Register n (ADSSTRn) [n = 0 to 15, L, O] 729
12.2.13 A/D Sample-and-Hold Circuit Control Register (ADSHCR) 730
12.2.14 A/D Disconnection Detection Control Register (ADDISCR) 731
12.2.15 A/D Group Scan Priority Control Register (ADGSPCR) 732
12.2.16 A/D High-/Low-potential Reference Voltage Control Register (ADHVREFCNT) 734
12.2.17 A/D Conversion Clock Control Register (ADCKS) 735
12.2.18 Port Mode Control Registers (PMC7 to PMC10, PMC12) 736
12.2.19 Port Mode Registers (PM7 to PM10, PM12) 737
12.2.20 A/D Converter Access Window Register (ADWINR) 739
12.3 Operation 740
12.3.1 Scanning Operation 740
12.3.2 Single Scan Mode 741
12.3.3 Continuous Scan Mode 744
12.3.4 Group Scan Mode 746
12.3.5 Analog Input Sampling Time and Scan Conversion Time 755
12.3.6 Usage Example of A/D Data Register Automatic Clearing Function 759
12.3.7 A/D-converted Value Addition/Average Mode 759
12.3.8 Disconnection Detection Assist Function 759
12.3.9 Self-diagnosis Function 761
12.3.10 Starting A/D Conversion with Synchronous Trigger from Peripheral Function 761
12.4 Interrupt Sources and DTC Transfer Requests 762
12.4.1 Interrupt Requests 762
12.5 Event Link Function (For RL78/F25 only) 762
12.5.1 12-bit A/D converter operation by event from the ELC 762
12.5.2 Note on 12-bit A/D Converter When an Events Input from the ELC 762
12.6 Reference Voltage 762
12.7 Usage Notes 762
12.7.1 Procedure for A/D Converter Initialization 762
12.7.2 Notes on Reading Data Registers 764
12.7.3 Procedure for Stopping A/D Conversion 764
12.7.4 Notes on Self-diagnosis Function 765

12.7.5 A/D Conversion Restarting Timing and Termination Timing ... 765
12.7.6 Notes on Scan End Interrupt Handling ... 765
12.7.7 Clock Supply Stop Setting ... 765
12.7.8 Notes on Entering Low Power Consumption States ... 765
12.7.9 Notes on Canceling STOP Mode ... 766
12.7.10 Error in Absolute Accuracy When Disconnection Detection Assistance Is in Use ... 766
12.7.11 Voltage Range of Analog Power Supply Pins ... 766
12.7.12 Notes on Board Design ... 766
12.7.13 Notes on Noise Prevention ... 767
12.7.14 Allowable Impedance of Signal Source ... 768

CHAPTER 13 D/A CONVERTER (RL78/F25 Only) ... 769

13.1 Function of D/A Converter ... 769
13.2 Configuration of D/A Converter ... 770
13.3 Registers of D/A Converter ... 771
13.3.1 Port Mode Control Register 8 (PMC8) ... 772
13.3.2 Peripheral Enable Register 1 (PER1) ... 773
13.3.3 D/A Converter Mode Register (DAM) ... 774
13.3.4 D/A Converter Mode Register 2 (DAM2) ... 775
13.3.5 D/A Conversion Value Setting Register 0 (DACS0) ... 776
13.3.6 Port Mode Register 8 (PM8) ... 777
13.4 Operations of D/A Converter ... 778
13.4.1 Operation in Normal Mode ... 778
13.4.2 Operation in Real-time Output Mode ... 779
13.5 Cautions for D/A Converter ... 780

CHAPTER 14 COMPARATOR (RL78/F25 Only) ... 781

14.1 Overview ... 781
14.2 Registers to Control the Comparator ... 783
14.2.1 Peripheral Enable Register 1 (PER1) ... 784
14.2.2 Comparator Control Register (CMPCTL) ... 785
14.2.3 Comparator I/O Select Register (CMPSEL) ... 787
14.2.4 Comparator Output Monitor Register (CMPMON) ... 789
14.2.5 Port Mode Control Register 8 (PMC8) ... 790
14.2.6 D/A Converter Mode Register 2 (DAM2) ... 791
14.2.7 Port Mode Register 4 (PM4) ... 792
14.2.8 Port Mode Register 8 (PM8) ... 793
14.3 Operation ... 794
14.3.1 Noise Filter ... 795
14.3.2 Comparator Interrupts ... 796
14.3.3 Comparator ELC/PWMOPA Event Output ... 796
14.3.4 Comparator Pin Output ... 796
14.3.5 Stopping or Supplying Comparator Clock ... 796
14.3.6 Comparator Setting Flowchart ... 797

CHAPTER 15 CAPACITIVE SENSING UNIT (CTSU2SLa) ... 799

15.1 Overview ... 801
15.1.1 Overview of CTSU Functions ... 801
15.1.2 Measurement Status ... 803

15.2 Registers to Control the Capacitive Sensing Unit ... 804
15.2.1 Peripheral Enable Registers 1 (PER1) ... 806
15.2.2 CTSU Control Register A (CTSUCRAL, CTSUCRAH) ... 807
15.2.3 CTSU Control Register B (CTSUCRBL, CTSUCRBH) ... 814
15.2.4 CTSU Measurement Channel Register (CTSUMCHL, CTSUMCHH) ... 817
15.2.5 CTSU Channel Enable Control Register A (CTSUCHACAL, CTSUCHACAH) ... 819
15.2.6 CTSU Channel Transmit/Receive Control Register A (CTSUCHTRCAL, CTSUCHTRCAH) ... 820
15.2.7 CTSU Status Register (CTSUSRL) ... 821
15.2.8 CTSU Sensor Offset Register (CTSUSO0, CTSUSO1) ... 823
15.2.9 CTSU Sensor Counter Register (CTSUSC, CTSUUC) ... 825
15.2.10 CTSU Calibration Register (CTSUDBGR0, CTSUDBGR1) ... 826
15.2.11 CTSU Sensor Unit Clock Control Registers A, B (CTSUSUCLKx) [x: 0 to 3] ... 829
15.2.12 CTSU Option Setting Register (CTSUOPTL, CTSUOPTH) ... 831
15.2.13 CTSU Sensor Counter Automatic Correction Table Access Register (CTSUSCTACTL, CTSUSCTACTH) ... 834
15.2.14 CTSU Multi-Clock Automatic Correction Table Registers 1, 2, 3 (CTSUMACT1L, 1H, 2L, 2H, 3L, 3H) ... 835
15.2.15 CTSU Automatic Judgment Control Register (CTSUAJCRL, CTSUAJCRH) ... 838
15.2.16 CTSU Threshold Register (CTSUAJTHRL, CTSUAJTHRH) ... 841
15.2.17 CTSU Moving Average Result Register (CTSUAJMMARL, CTSUAJMMARH) ... 842
15.2.18 CTSU Baseline Average Calculation Register (CTSUAJBLACL, CTSUAJBLACH) ... 843
15.2.19 CTSU Baseline Average Result Register (CTSUAJBLARL, CTSUAJBLARH) ... 844
15.2.20 CTSU Automatic Judgment Result Register (CTSUAJRRL) ... 845
15.2.21 CTSU External Trigger Select Register (CTSUTRG) ... 847
15.2.22 CTSU Trimming Register A (CTSUTRIM0, CTSUTRIM1) ... 848
15.2.23 CTSU Trimming Register B (CTSUTRIM2, CTSUTRIM3) ... 849
15.3 Operation ... 850
15.3.1 Principles of Measurement Operation ... 850
15.3.2 Initial Setting Flowchart ... 850
15.3.3 Measurement Modes ... 851
15.3.4 Scan Mode ... 853
15.3.5 Multi-clock Measurement ... 853
15.3.6 Automatic Judgment Function ... 853
15.3.7 Multiple Electrode Connection Function ... 854
15.3.8 Interrupts ... 855
15.4 Usage Notes ... 856
15.4.1 Evaluation of Detection Operation by Capacitance Sensing Unit (CTSU2SLa) ... 856
15.4.2 Port Settings When Using CTSU ... 856
15.4.3 TSCAP Pin ... 856
15.4.4 Changing Clock and Port Settings During Measurement Operation ... 856
15.4.5 Software Trigger ... 856
15.4.6 External Trigger ... 857
15.4.7 Reading the CTSU Sensor Counter Register (CTSUSC, CTSUUC) ... 857
15.4.8 Forced Stop ... 857
15.4.9 Setting of the Sampling Cycle on Spread Spectrum ... 857
15.4.10 Transmit Pin Setting When Using the Self-capacitance Method ... 857
15.4.11 Notes on Noise Countermeasures ... 857

CHAPTER 16 SERIAL ARRAY UNIT 859
16.1 Functions of Serial Array Unit 860
16.1.1 3-wire Serial I/O (CSI00, CSI01, CSI10, CSI11) 860
16.1.2 UART (UART0, UART1) 861
16.1.3 Simplified I^2C (IIC00, IIC01, IIC10, IIC11) 862
16.2 Configuration of Serial Array Unit 863
16.3 Registers Controlling Serial Array Unit 868
16.3.1 Peripheral Enable Register 0 (PER0) 869
16.3.2 Serial Clock Select Register m (SPSm) 870
16.3.3 Serial Mode Register mn (SMRmn) 871
16.3.4 Serial Communication Operation Setting Register mn (SCRmn) 873
16.3.5 Higher 7 Bits of the Serial Data Register mn (SDRmn) 875
16.3.6 Serial Flag Clear Trigger Register mn (SIRmn) 876
16.3.7 Serial Status Register mn (SSRmn) 877
16.3.8 Serial Channel Start Register m (SSm) 879
16.3.9 Serial Channel Stop Register m (STm) 880
16.3.10 Serial Channel Enable Status Register m (SEm) 881
16.3.11 Serial Output Enable Register m (SOEm) 882
16.3.12 Serial Output Register m (SOm) 883
16.3.13 Serial Output Level Register m (SOLm) 884
16.3.14 Serial Slave Select Enable Register m (SSEm) 885
16.3.15 Input Switch Control Register (ISC) 886
16.3.16 Noise Filter Enable Register 0 (NFEN0) 887
16.3.17 Port Input Mode Registers (PIM1, PIM3, PIM5 to PIM7, PIM12) 888
16.3.18 Port Output Mode Registers (POM1, POM3, POM6, POM7, POM12) 889
16.3.19 Port Mode Registers (PM1, PM3 to PM7, PM12) 890
16.3.20 Port Input Threshold Control Registers (PITHL1, PITHL3 to PITHL7, PITHL10, PITHL12, PITHL15) 892
16.3.21 Unit Select Register (UTSEL) 894
16.3.22 I^2S Function Control Register (I2SER) 895
16.4 Operation Stop Mode 896
16.4.1 Stopping the Operation by Units 896
16.4.2 Stopping the Operation by Channels 897
16.5 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) Communication 898
16.5.1 Master Transmission 900
16.5.2 Master Reception 909
16.5.3 Master Transmission/Reception 917
16.5.4 Slave Transmission 926
16.5.5 Slave Reception 935
16.5.6 Slave Transmission/Reception 941
16.5.7 Calculating Transfer Clock Frequency 950
16.5.8 Procedure for Processing Errors that Occurred During 3-wire Serial I/O (CSI00, CSI01, CSI10, CSI11) Communication 952
16.6 Clock Synchronous Serial Communication with SPI Function 953
16.6.1 Master Transmission 957
16.6.2 Master Reception 965
16.6.3 Master Transmission/Reception 972
16.6.4 Slave Transmission 980
16.6.5 Slave Reception 989

16.6.6 Slave Transmission/Reception 995
16.6.7 Calculating Transfer Clock Frequency 1004
16.6.8 Procedure for Processing Errors that Occurred During Clock Synchronous Serial Communication with SPI Function 1006
16.7 UART (UART0, UART1) Communication 1007
16.7.1 UART Transmission 1009
16.7.2 UART Reception 1018
16.7.3 Calculating Baud Rate 1024
16.7.4 Procedure for Processing Errors That Occurred During UART (UART0, UART1) Communication 1028
16.8 LIN Communication 1029
16.8.1 LIN Transmission 1029
16.8.2 LIN Reception 1032
16.9 Simplified I^2C (IIC00, IIC01, IIC10, IIC11) Communication 1036
16.9.1 Address Field Transmission 1038
16.9.2 Data Transmission 1043
16.9.3 Data Reception 1047
16.9.4 Stop Condition Generation 1051
16.9.5 Calculating Transfer Rate 1053
16.9.6 Procedure for Processing Errors That Occurred During Simplified I^2C (IIC00, IIC01, IIC10, IIC11) Communication 1056
16.10 Simplified I^2S Communication 1057
16.10.1 Overview 1058
16.10.2 Master Transmission 1059
16.10.3 Operation of Simplified I^2S Communication 1060

CHAPTER 17 SERIAL INTERFACE IICA 1061

17.1 Functions of Serial Interface IICA 1061
17.2 Configuration of Serial Interface IICA 1064
17.3 Registers Controlling Serial Interface IICA 1067
17.3.1 Peripheral Enable Register 0 (PER0) 1067
17.3.2 IICA Control Register 00 (IICCTL00) 1068
17.3.3 IICA Status Register 0 (IICS0) 1073
17.3.4 IICA Flag Register 0 (IICF0) 1075
17.3.5 IICA Control Register 01 (IICCTL01) 1077
17.3.6 IICA Low-level Width Setting Register 0 (IICWL0) 1079
17.3.7 IICA High-level Width Setting Register 0 (IICWH0) 1079
17.3.8 Port Mode Register 6 (PM6) 1080
17.3.9 Port Output Mode Register (POM6) 1081
17.4 I^2C Bus Mode Functions 1082
17.4.1 Pin Configuration 1082
17.4.2 Setting Transfer Clock by Using IICWL0 and IICWH0 Registers 1083
17.5 I^2C Bus Definitions and Control Methods 1085
17.5.1 Start Conditions 1085
17.5.2 Addresses 1086
17.5.3 Transfer Direction Specification 1086
17.5.4 Acknowledge (ACK) 1087
17.5.5 Stop Condition 1088
17.5.6 Wait 1089
17.5.7 Canceling Wait 1091

17.5.8 Interrupt Request (INTIICA0) Generation Timing and Wait Control 1092
17.5.9 Address Match Detection Method 1093
17.5.10 Error Detection 1093
17.5.11 Extension Code 1093
17.5.12 Arbitration 1094
17.5.13 Wakeup Function 1096
17.5.14 Communication Reservation 1099
17.5.15 Cautions 1103
17.5.16 Communication Operations 1104
17.5.17 Timing of I^2C Interrupt Request (INTIICA0) Occurrence 1112
17.6 Timing Charts 1133
17.6.1 Example Timing Charts of Master to Slave Communications 1134
17.6.2 Example Timing Charts of Slave to Master Communications 1142

CHAPTER 18 LIN/UART MODULE (RLIN3) 1148

18.1 Overview 1148
18.2 Register Descriptions 1154
18.2.1 LIN Registers for Master Mode 1156
18.2.2 LIN Registers for Slave Mode 1184
18.2.3 Registers for UART 1214
18.3 Modes 1245
18.3.1 LIN Reset Mode 1247
18.3.2 LIN Mode 1248
18.3.3 UART Mode 1250
18.3.4 LIN Self-Test Mode 1250
18.4 LIN Mode 1251
18.4.1 Operation Overview 1251
18.4.2 Data Transmission/Reception 1259
18.4.3 Transmission/Reception Data Buffering 1261
18.4.4 Wake-up Transmission/Reception 1263
18.4.5 Status 1265
18.4.6 Error Status 1267
18.5 UART Mode 1273
18.5.1 Operation Overview 1273
18.5.2 Data Transmission/Reception 1287
18.5.3 Buffer Processing of Transmission Data 1289
18.5.4 Status 1290
18.5.5 Error Status 1291
18.6 LIN Self-Test Mode 1292
18.6.1 Change to LIN Self-Test Mode 1293
18.6.2 Transmission in LIN Master Self-Test Mode 1294
18.6.3 Reception in LIN Master Self-Test Mode 1295
18.6.4 Transmission in LIN Slave Self-Test Mode 1296
18.6.5 Reception in LIN Slave Self-Test Mode 1297
18.6.6 Terminating LIN Self-Test Mode 1298
18.7 Baud Rate Generator 1299
18.7.1 LIN Master Mode 1299
18.7.2 LIN Slave Mode 1301
18.7.3 UART Mode 1303

18.8 Noise Filter 1305
18.9 Interrupts 1307

CHAPTER 19 CAN INTERFACE (RS-CANFD lite) (RL78/F25 Only) 1308

19.1 Overview 1308
19.2 Input/Output Pins 1311
19.3 Register Descriptions 1312
19.3.1 CANs Nominal Bitrate Configuration Register (CsNCFGH, CsNCFGL) [s = 0, 1] 1323
19.3.2 CANs Control Register (CsCTRH, CsCTRL) [s = 0, 1] 1325
19.3.3 CANs Status Register (CsSTSH, CsSTSL) [s = 0, 1] 1330
19.3.4 CANs Error Flag Register (CsERFLH, CsERFLL) [s = 0, 1] 1333
19.3.5 CANs Data Bitrate Configuration Register (CsDCFGH, CsDCFGL) [s = 0, 1] 1340
19.3.6 CANs CAN-FD Configuration Register (CsFDCFGH, CsFDCFGL) [s = 0, 1] 1342
19.3.7 CANs CAN-FD Control Register (CsFDCTRH, CsFDCTRL) [s = 0, 1] 1345
19.3.8 CANs CAN-FD Status Register (CsFDSTSH, CsFDSTSL) [s = 0, 1] 1347
19.3.9 CANs CAN-FD CRC Register (CsFDCRCH, CsFDCRCL) [s =0, 1] 1350
19.3.10 Global IP Version Register (GIPVHs, GIPVLs) [s = 0, 1] 1351
19.3.11 Global Configuration Register (GCFGHs, GCFGLs) [s = 0, 1] 1352
19.3.12 Global Control Register (GCTRHs, GCTRLs) [s = 0, 1] 1354
19.3.13 Global Status Register (GSTSs) [s = 0, 1] 1356
19.3.14 Global Error Flag Register (GERFLHs, GERFLLs) [s = 0, 1] 1357
19.3.15 Global TX Interrupt Status Register (GTINTSTSs) [s = 0, 1] 1359
19.3.16 Global Timestamp Counter Register (GTSCs) [s = 0, 1] 1360
19.3.17 Global Acceptance Filter List Entry Control Register (GAFLECTRs) [s = 0, 1] 1361
19.3.18 Global Acceptance Filter List Configuration Register (GAFLCFGs) [s = 0, 1] 1362
19.3.19 Global Acceptance Filter List ID Register i (GAFLIDiHs, GAFLIDiLs) [i = 0 to 15] [s = 0, 1] 1363
19.3.20 Global Acceptance Filter List Mask Register i (GAFLMiHs, GAFLMiLs) [i = 0 to 15] [s = 0, 1] 1365
19.3.21 Global Acceptance Filter List Pointer 0 Register i (GAFLP0iHs, GAFLP0iLs) [i = 0 to 15] [s = 0, 1] 1367
19.3.22 Global Acceptance Filter List Pointer 1 Register i (GAFLP1iLs) [i = 0 to 15] [s = 0, 1] 1369
19.3.23 RX Message Buffer Number Register (RMNBs) [s = 0, 1] 1370
19.3.24 RX Message Buffer New Data Register (RMNDs) [s = 0, 1] 1371
19.3.25 RX FIFO Configuration / Control Register k (RFCCks) [k = 0, 1] [s = 0, 1] 1372
19.3.26 RX FIFO Status Register k (RFSTSks) [k = 0, 1] [s = 0, 1] 1374
19.3.27 RX FIFO Pointer Control Register ks (RFPCTRks) [k = 0, 1] [s = 0, 1] 1376
19.3.28 Common FIFO Configuration / Control Register (CFCCLs, CFCCHs) [s = 0, 1] 1377
19.3.29 Common FIFO Status Register (CFSTSs) [s = 0, 1] 1380
19.3.30 Common FIFO Pointer Control Register (CFPCTRs) [s = 0, 1] 1382
19.3.31 FIFO Empty Status Register (FESTSs) [s = 0, 1] 1383
19.3.32 FIFO Full Status Register (FFSTSs) [s = 0, 1] 1384
19.3.33 FIFO Message Lost Status Register (FMSTSs) [s = 0, 1] 1385
19.3.34 RX FIFO Interrupt Flag Status Register (RFISTSs) [s = 0, 1] 1386
19.3.35 CANs TX Message Buffer Control Register m (TMCms) [m = 0 to 3] [s = 0, 1] 1387
19.3.36 CANs TX Message Buffer Status Register m (TMSTSms) [m = 0 to 3] [s = 0, 1] 1389
19.3.37 CANs TX Message Buffer Transmission Request Status Register (TMTRSTSs) [s = 0, 1] 1391
19.3.38 CANs TX Message Buffer Transmission Abort Request Status Register (TMTARSTSs) [s = 0, 1] 1392
19.3.39 CANs TX Message Buffer Transmission Completion Status Register (TMTCSTSs) [s = 0, 1] 1393
19.3.40 CANs TX Message Buffer Transmission Abort Status Register (TMTASTSs) [s = 0, 1] 1394
19.3.41 CANs TX Message Buffer Interrupt Enable Configuration Register (TMIECs) [s= 0, 1] 1395

19.3.42 CANs TX History List Configuration / Control Register (THLCCs) [s = 0, 1] 1396
19.3.43 CANs TX History List Status Register (THLSTSs) [s = 0, 1] 1397
19.3.44 CANs TX History List Access Register 0 (THLACC0Hs, THLACC0Ls) [s = 0, 1] 1399
19.3.45 CANs TX History List Access Register 1 (THLACC1Hs, THLACC1Ls) [s = 0, 1] 1400
19.3.46 CANs TX History List Pointer Control Register (THLPCTRs) [s = 0, 1] 1401
19.3.47 Global Reset Control Register (GRSTCs) [s = 0, 1] 1402
19.3.48 Global Test Configuration Register (GTSTCFGs) [s = 0, 1] 1403
19.3.49 Global Test Control Register (GTSTCTRs) [s = 0, 1] 1404
19.3.50 Global FD Configuration Register (GFDCFGs) [s = 0, 1] 1405
19.3.51 Global Lock Key Register (GLOCKKs) [s = 0, 1] 1406
19.3.52 RAM Test Page Access Register r (RPGACCrHs, RPGACCrLs) [r = 0 to 63] [s = 0, 1] 1407
19.3.53 Global Pretended Network Filter List Entry Control Register (GPFLECTRs) [s = 0, 1] 1408
19.3.54 Global Pretended Network Filter List Configuration Register (GPFLCFGs) [s = 0, 1] 1409
19.3.55 Global Pretended Network Filter List ID Register j (GPFLIDjHs, GPFLIDjLs) [j = 0, 1] [s = 0, 1] 1410
19.3.56 Global Pretended Network Filter List Mask Register j (GPFLMjHs, GPFLMjLs) [j = 0, 1] [s = 0, 1] 1412
19.3.57 Global Pretended Network Filter List Pointer 0 Register j (GPFLP0jHs, GPFLP0jLs) [j = 0, 1] [s = 0, 1] 1414
19.3.58 Global Pretended Network Filter List Pointer 1 Register j (GPFLP1jLs) [j = 0, 1] [s = 0, 1] 1417
19.3.59 Global Pretended Network Filter List Payload Type Register j (GPFLPTjHs, GPFLPTjLs) [j = 0, 1] [s = 0, 1] 1418
19.3.60 Global Pretended Network Filter List Payload Data 0 Register j (GPFLPD0jHs, GPFLPD0jLs) [j = 0, 1] [s = 0, 1] 1420
19.3.61 Global Pretended Network Filter List Payload Data 1 Register j (GPFLPD1jHs, GPFLPD1jLs) [j = 0, 1] [s = 0, 1] 1421
19.3.62 Global Pretended Network Filter List Payload Mask 0 Register j (GPFLPM0jHs, GPFLPM0jLs) [j= 0, 1] [s = 0, 1] 1422
19.3.63 Global Pretended Network Filter List Payload Mask 1 Register j (GPFLPM1jHs, GPFLPM1jLs) [j = 0, 1] [s = 0, 1] 1423
19.3.64 Global AFL Ignore Entry Register (GAFLIGENTs) [s = 0, 1] 1424
19.3.65 Global AFL Ignore Control Register (GAFLIGCTRs) [s = 0, 1] 1425
19.3.66 RX Message Buffer Interrupt Enable Configuration Register (RMIECs) [s = 0, 1] 1426
19.3.67 RS-CANFD lite Global RAM Windows Control Register (CFDGRWCs) [s = 0, 1] 1427
19.3.68 CANFD Window Access Control Register (CFDWINR) 1428
19.4 Identifier Bits Alignment 1429
19.5 Message Buffer Component Structure 1430
19.5.1 RX Message Buffer ID Register n (RMIDnHs, RMIDnLs) [n = 0 to 15] [s = 0, 1] 1431
19.5.2 RX Message Buffer Pointer Register n (RMPTRnHs, RMPTRnLs) [n = 0 to 15] [s = 0, 1] 1432
19.5.3 RX Message Buffer CAN-FD Status Register n (RMSTSnHs, RMSTSnLs) [n = 0 to 15] [s = 0, 1] 1433
19.5.4 RX Message Buffer Data Field p Register n (RMDn_pHs, RMDn_pLs) [n = 0 to 15] [p = 0 to 15] [s = 0, 1] 1435
19.5.5 RX FIFO Access ID Register k (RFIDkHs, RFIDkLs) [k = 0, 1] [s = 0, 1] 1436
19.5.6 RX FIFO Access Pointer Register k (RFPTRkHs, RFPTRkLs) [k = 0, 1] [s = 0, 1] 1437
19.5.7 RX FIFO Access CAN-FD Status Register k (RFFDSTSkHs, RFFDSTSkLs) [k = 0, 1] [s = 0, 1] 1438
19.5.8 RX FIFO Access Data Field p Register k (RFDFk_pHs, RFDFk_pLs) [k = 0, 1] [p = 0 to 15] [s = 0, 1] 1439
19.5.9 Common FIFO Access ID Register (CFIDHs, CFIDLs) [s = 0, 1] 1440
19.5.10 Common FIFO Access Pointer Register (CFPTRHs, CFPTRLs) [s = 0, 1] 1442

19.5.11 Common FIFO Access CAN-FD Control/Status Register (CFFDCSTSHs, CFFDCSTSLs) [s = 0, 1] ... 1443
19.5.12 Common FIFO Access Data Field p Register (CFDFpHs, CFDFpLs) [p = 0 to 15] [s = 0, 1] ... 1445
19.5.13 CANs TX Message Buffer ID Register m (TMIDmHs, TMIDmLs) [m = 0 to 3] [s = 0, 1] ... 1446
19.5.14 CANs TX Message Buffer Pointer Register m (TMPTRmHs) [m = 0 to 3] [s = 0, 1] ... 1447
19.5.15 CANs TX Message Buffer CAN-FD Control Register m (TMFDCTRmHs, TMFDCTRmLs) [m = 0 to 3] [s = 0, 1] ... 1448
19.5.16 CANs TX Message Buffer Data Field p Register m (TMDFm_pHs, TMDFm_pLs) [m = 0 to 3], [p = 0 to 15] [s = 0, 1] ... 1450
19.6 CAN Modes ... 1451
19.6.1 Global Modes ... 1451
19.6.2 Channel Modes ... 1459
19.6.3 Global Mode – Channel Mode Transition Interactions ... 1465
19.7 Acceptance Filtering Function using Global Acceptance Filter List (AFL) ... 1475
19.7.1 Overview ... 1475
19.7.2 Allocation of AFL Entries ... 1477
19.7.3 AFL Entry Description ... 1478
19.7.4 Entering Entries in the AFL ... 1480
19.7.5 Loopback Modes ... 1481
19.7.6 IDE Masking ... 1482
19.7.7 Updating AFL Entry During Communication ... 1483
19.8 FIFO Buffers and Normal MB Configuration ... 1486
19.8.1 Normal RX Message Buffers ... 1487
19.8.2 FIFO Buffers ... 1487
19.9 Reception Function ... 1492
19.9.1 Message Storage in RX Message Buffers ... 1492
19.9.2 Message Storage in FIFO Buffers ... 1494
19.9.3 Timestamp ... 1496
19.10 Transmission Functions ... 1497
19.10.1 Transmission Priority ... 1498
19.10.2 Normal Transmission ... 1499
19.10.3 TX FIFO Transmission ... 1503
19.10.4 TX History List ... 1507
19.10.5 TX Data Padding ... 1510
19.11 Test Function ... 1511
19.11.1 Channel Specific Test Modes ... 1511
19.11.2 Global Test Modes ... 1514
19.12 Interrupt ... 1519
19.12.1 Interrupt Sources ... 1519
19.12.2 Interrupt Processing Flow ... 1523
19.13 RAM Area Configuration ... 1524
19.13.1 Message Buffer Area Configuration Examples ... 1527
19.13.2 RAM Initialization ... 1527
19.14 RAM Window ... 1528
19.15 Initial Settings ... 1529
19.15.1 Initialization of CAN Clock, Bit Timing and Baud Rate ... 1529
19.15.2 CAN Module Configuration After Reset ... 1536
19.16 Pretended Network Filter (PNF) ... 1538

19.16.1 Explanation in Each Mode ... 1539
19.16.2 Details of a Filtering Function ... 1541
19.17 Notes on the CAN Module ... 1546

CHAPTER 20 DTC ... 1547
20.1 Overview ... 1547
20.2 Registers ... 1549
20.2.1 Allocation of DTC Control Data Area and DTC Vector Table Area ... 1551
20.2.2 DTC Control Data Allocation ... 1552
20.2.3 DTC Vector Table ... 1553
20.2.4 Peripheral Enable Register 1 (PER1) ... 1556
20.2.5 DTC Activation Enable Register i (DTCENi) ... 1557
20.2.6 DTC Base Address Register (DTCBAR) ... 1559
20.2.7 DTC Control Register j (DTCCRj) (j = 0 to 23) ... 1560
20.2.8 DTC Block Size Register j (DTBLSj) (j = 0 to 23) ... 1561
20.2.9 DTC Transfer Count Register j (DTCCTj) (j = 0 to 23) ... 1561
20.2.10 DTC Transfer Count Reload Register j (DTRLDj) (j = 0 to 23) ... 1562
20.2.11 DTC Source Address Register j (DTSARj) (j = 0 to 23) ... 1562
20.2.12 DTC Destination Address Register j (DTDARj) (j = 0 to 23) ... 1562
20.2.13 High-speed DTC Channel Select Register 0 (SELHS0) ... 1563
20.2.14 High-speed DTC Channel Select Register 1 (SELHS1) ... 1564
20.2.15 High-speed DTC Control Register m (HDTCCRm) (m = 0, 1) ... 1565
20.2.16 High-speed DTC Transfer Count Register m (HDTCCTm) (m = 0, 1) ... 1566
20.2.17 DTC Transfer Count Reload Register m (HDTRLDm) (m = 0, 1) ... 1567
20.2.18 High-speed DTC Source Address Register m (HDTSARm) (m = 0, 1) ... 1567
20.2.19 High-speed DTC Destination Address Register m (HDTDARm) (m = 0, 1) ... 1567
20.3 Operation ... 1568
20.3.1 Activation Sources ... 1569
20.3.2 Normal Mode ... 1570
20.3.3 Repeat Mode ... 1572
20.3.4 Chain Transfers ... 1576
20.3.5 High-Speed Transfer Operation ... 1577
20.4 Notes on DTC ... 1578
20.4.1 Setting DTC Registers and Vector Table ... 1578
20.4.2 Allocation of DTC Control Data Area and DTC Vector Table Area ... 1578
20.4.3 DTC Pending Instruction ... 1579
20.4.4 Operations When an Instruction Which Accesses an SFR Register That Requires a Wait Is Executed ... 1579
20.4.5 Operation When Accessing Data Flash Memory Space ... 1579
20.4.6 Number of DTC Execution Clock Cycles ... 1580
20.4.7 Number of High-speed DTC Execution Clock Cycles ... 1581
20.4.8 DTC Response Time ... 1582
20.4.9 DTC Activation Sources ... 1582
20.4.10 Operation in Standby Mode Status ... 1583
20.4.11 Notes When the RAM Area Is the Source of the Data for Transfer ... 1583
20.4.12 Vector Address for High-Speed Transfer ... 1583

CHAPTER 21 EVENT LINK CONTROLLER (ELC) (RL78/F25 Only) ... 1584
21.1 Overview ... 1584

21.2 Registers 1585
21.2.1 Event Output Destination Select Register n (ELSELRn) (n = 00 to 25) 1586
21.2.2 Timer Input Select Register 0 (TIS0) 1589
21.2.3 A/D Conversion Start Trigger Select Register (ADSTRGR) 1589
21.2.4 D/A Converter Mode Register (DAM) 1589
21.3 Operation 1590

CHAPTER 22 INTERRUPT FUNCTIONS 1591

22.1 Interrupt Function Types 1591
22.2 Interrupt Sources and Configuration 1592
22.3 Registers Controlling Interrupt Functions 1603
22.3.1 Interrupt Request Flag Registers (IFxL, IFxH) 1612
22.3.2 Interrupt Mask Flag Registers (MKxL, MKxH) 1614
22.3.3 Priority Specification Flag Registers (PRxxL, PRxxH) 1616
22.3.4 External Interrupt Rising/Falling Edge Enable Registers (EGP0, EGP1, EGN0, EGN1) 1619
22.3.5 Interrupt Source Determination Flag Register 0 (INTFLG0) 1621
22.3.6 Interrupt Source Determination Flag Register 1 (INTFLG1) 1622
22.3.7 Interrupt Source Determination Flag Register 2 (INTFLG2) 1623
22.3.8 Interrupt Source Determination Flag Register 3 (INTFLG3) 1624
22.3.9 Interrupt Mask Register (INTMSK) 1625
22.3.10 Input Switch Control Register (ISC) 1626
22.3.11 Program Status Word (PSW) 1627
22.4 Interrupt Servicing Operations 1628
22.4.1 Determination of Interrupt Source 1628
22.4.2 Maskable Interrupt Request Acknowledgment 1630
22.4.3 Software Interrupt Request Acknowledgment 1632
22.4.4 Multiple Interrupt Servicing 1632
22.4.5 Interrupt Servicing During Division Instruction 1636
22.4.6 Interrupt Request Hold 1638

CHAPTER 23 KEY INTERRUPT FUNCTION 1639

23.1 Functions of Key Interrupt 1639
23.2 Configuration of Key Interrupt 1640
23.3 Register Controlling Key Interrupt 1641
23.3.1 Key Return Mode Register (KRM) 1641

CHAPTER 24 STANDBY FUNCTION 1642

24.1 Standby Function and Configuration 1642
24.1.1 Standby Function 1642
24.2 Registers Controlling Standby Function 1643
24.2.1 Oscillation Stabilization Time Counter Status Register (OSTC) 1644
24.2.2 Oscillation Stabilization Time Select Register (OSTS) 1645
24.2.3 STOP Status Output Control Register (STPSTC) 1646
24.2.4 Memory Power Reduction Control Register (PSMCR) 1647
24.3 Standby Function Operation 1649
24.3.1 HALT Mode 1649
24.3.2 STOP Mode 1656
24.3.3 SNOOZE Mode 1662

CHAPTER 25 RESET FUNCTION 1668
25.1 Register for Confirming Reset Source 1675
25.1.1 Reset Control Flag Register (RESF) 1675
25.1.2 POR/CLM Reset Confirmation Register (POCRES) 1677

CHAPTER 26 POWER-ON-RESET CIRCUIT 1678
26.1 Functions of Power-on-reset Circuit 1678
26.2 Configuration of Power-on-reset Circuit 1679
26.3 Operation of Power-on-reset Circuit 1679
26.4 Cautions for Power-on-reset Circuit 1682

CHAPTER 27 VOLTAGE DETECTOR 1684
27.1 Functions of Voltage Detector 1684
27.2 Configuration of Voltage Detector 1685
27.3 Registers Controlling Voltage Detector 1686
27.3.1 Voltage Detection Register (LVIM) 1686
27.3.2 Voltage Detection Level Register (LVIS) 1687
27.3.3 LVD Detection Flag Clearing Register (LVDFCLR) 1689
27.4 Operation of Voltage Detector 1690
27.4.1 When Used as Reset Mode 1690
27.4.2 When Used as Interrupt Mode 1691
27.5 Cautions for Voltage Detector 1693
27.5.1 Checking Reset Source 1693
27.5.2 Delay from the Time LVD0 Reset Source Is Generated Until the Time LVD0 Reset Has Been Generated or Released 1694
27.5.3 Turning Power on When LVD0 Is Off 1694
27.5.4 Operating Voltage Fall When LVD0 Is Off 1694
27.5.5 Procedure for Setting the LVD1 Detection Voltage 1695

CHAPTER 28 APPLICATION ACCELERATOR UNIT 1696
28.1 Overview 1696
28.2 Registers 1698
28.2.1 Peripheral Enable Register 2 (PER2) 1700
28.2.2 Application Accelerator Unit Access Window Register (AAUWINR) 1701
28.2.3 AAU Data Register i (ADTREGi, i = 0 to 3) 1702
28.2.4 AAU Control Register (ACTLREG) 1703
28.2.5 PI Control Coefficient Range Setting Register/ Current Limit Duty Register (AKRAG/ ADUTYMX) 1704
28.2.6 D-axis Current Reference Register/ DC/DC Channel 1 Current Reference Register (AIDREF/AL1REF) 1705
28.2.7 Q-axis Current Reference Register/ DC/DC channel 2 Current Reference Register (AIQREF/AL2REF) 1706
28.2.8 D-axis Proportional Coefficient Register/ DC/DC Channel 3 Current Reference Register (AKPD/AL3REF) 1707
28.2.9 D-axis Integral Coefficient Register/ DC/DC Channel 1 Offset Current Register (AKID/AL1OFS) 1708
28.2.10 Q-axis Proportional Coefficient Register/ DC/DC Channel 2 Offset Current Register (AKPQ/AL2OFS) 1709
28.2.11 Q-axis Integral Coefficient Register/ DC/DC Channel 3 Offset Current Register (AKIQ/AL3OFS) 1710
28.2.12 Current Limit Register/ DC/DC Proportional Coefficient 1 Register (AILIM/AKI1) 1711

28.2.13 PI Control Limit Register/ DC/DC Proportional Coefficient 2 Register (APILIM/AKI2) 1712
28.2.14 D-axis Current Buffer Register L/ DC/DC Channel 1 Previous Duty Register (AIDBFL/ADUTYL1) 1713
28.2.15 D-axis Current Buffer Register H/ DC/DC Channel 2 Previous Duty Register (AIDBFH/ADUTYL2) 1714
28.2.16 Q-axis Current Buffer Register L/ DC/DC Channel 3 Previous Duty Register (AIQBFL/ADUTYL3) 1715
28.2.17 Q-axis Current Buffer Register H/ DC/DC Channel 1 Previous Current Register (AIQBFH/AIPL1) 1716
28.2.18 D-axis Limit Over Current Register/ DC/DC Channel 2 Previous Current Register (ADOVER/AIPL2) 1717
28.2.19 Q-axis Limit Over Current Register/ DC/DC Channel 3 Previous Current Register (AQOVER/AIPL3) 1718
28.3 Operation of the AAU 1719
28.3.1 Operation Overview 1719
28.3.2 Description for the Operation of Each Algorithm Mode 1722
28.4 Notes on Application Accelerator Unit 1743

CHAPTER 29 FUNCTIONAL SAFETY 1744
29.1 Overview of Functional Safety 1744
29.2 Registers Used by Functional Safety 1746
29.3 Operation of Functional Safety 1747
29.3.1 Flash Memory CRC Operation Function (High-speed CRC) 1747
29.3.2 CRC Operation Function (General-purpose CRC) 1751
29.3.3 Internal RAM-ECC Function 1755
29.3.4 CAN RAM-ECC Function (RL78/F25 Only) 1761
29.3.5 Code Flash Memory ECC Function 1774
29.3.6 CPU Stack Pointer Monitor Function 1783
29.3.7 Clock Monitor Function 1786
29.3.8 Invalid Memory Access Detection Function 1790
29.3.9 Frequency Detection Function 1792
29.3.10 A/D Test Function 1794
29.3.11 Watchdog Timer Function 1794

CHAPTER 30 SECURITY FUNCTIONS 1795

CHAPTER 31 REGULATOR 1796
31.1 Regulator Overview 1796

CHAPTER 32 OPTION BYTE 1797
32.1 Functions of Option Bytes 1797
32.1.1 User Option Byte (000C0H/040C0H to 000C2H/040C2H) 1797
32.1.2 On-chip Debug Option Byte (000C3H/040C3H) 1798
32.1.3 Security Option Byte (000C4H/040C4H) 1798
32.2 Format of User Option Byte 1799
32.3 Format of On-chip Debug Option Byte 1802
32.4 Format of Security Option Byte 1803
32.5 Setting of Option Byte 1804

CHAPTER 33 FLASH MEMORY 1805
33.1 Serial Programming Using Flash Memory Programmer 1807
33.1.1 Programming Environment 1808
33.1.2 Communication Mode 1808
33.1.3 Usage Notes 1809
33.2 Serial Programming Using External Device (That Incorporates UART) 1810
33.2.1 Programming Environment 1810
33.2.2 Communication Mode 1811
33.3 Connection of Pins on Board 1812
33.3.1 P40/TOOL0 Pin 1812
33.3.2 RESET Pin 1812
33.3.3 Port Pins 1813
33.3.4 REGC Pin 1813
33.3.5 X1 and X2 Pins 1813
33.3.6 Power Supply 1813
33.4 Flash Serial Programming Security ID 1813
33.5 Serial Programming Method 1814
33.5.1 Serial Programming Procedure 1814
33.5.2 Flash Memory Programming Mode 1814
33.5.3 Selecting Communication Mode 1816
33.5.4 Communication Commands 1816
33.6 Processing Time for Each Command When PG-FP6 Is in Use (Reference Value) 1818
33.7 Self-Programming 1819
33.7.1 Self-Programming Procedure 1820
33.7.2 Registers to Control the Flash Memory 1821
33.7.3 Setting the Flash Memory Control Mode 1838
33.7.4 Clearing the Registers for Use with the Flash Memory Sequencer 1840
33.7.5 Setting the Operating Frequency of the Flash Memory Sequencer 1840
33.7.6 Commands for Use with the Flash Memory Sequencer in the Respective Areas 1841
33.7.7 Boot Swap Function 1848
33.7.8 Flash Shield Window Function 1850
33.7.9 Example of Executing the Commands to Rewrite the Flash Memory Area 1851
33.7.10 Notes on Self-Programming 1854
33.8 Security Settings 1855
33.9 Data Flash Memory 1857
33.9.1 Overview of the Data Flash Memory 1857
33.9.2 Procedure for Accessing Data Flash Memory 1858

CHAPTER 34 ON-CHIP DEBUG FUNCTION 1859
34.1 Overview of On-chip Debug Function 1859
34.1.1 Hot Plug-in 1859
34.1.2 Real-time RAM Monitor (RRM) and Dynamic Memory Modification (DMM) by DTC 1859
34.2 Connecting E2/E2 Lite On-chip Debugging Emulator to RL78/F22 or RL78/F25 1861
34.3 On-chip Debug Security ID 1862
34.4 Securing of User Resources 1862
34.4.1 Securement of Memory Space 1862

CHAPTER 35 BCD CORRECTION CIRCUIT 1864
35.1 BCD Correction Circuit Function 1864
35.2 Registers Used by BCD Correction Circuit 1864
35.3 BCD Correction Circuit Operation 1865

CHAPTER 36 INSTRUCTION SET 1867
36.1 Conventions Used in Operation List 1867
36.1.1 Operand Identifiers and Specification Methods 1867
36.1.2 Description of Operation Column 1868
36.1.3 Description of Flag Operation Column 1869
36.1.4 PREFIX Instruction 1869
36.1.5 Multiply & Accumulate Instruction 1870
36.2 Operation List 1871

CHAPTER 37 ELECTRICAL SPECIFICATIONS (RL78/F25) 1889
37.1 Absolute Maximum Ratings 1890
37.2 Oscillator Characteristics 1893
37.2.1 Main System Clock Oscillator Characteristics 1893
37.2.2 On-chip Oscillator Characteristics 1894
37.2.3 Subsystem Clock Oscillator Characteristics 1895
37.2.4 PLL Circuit Characteristics 1896
37.3 DC Characteristics 1897
37.3.1 Pin Characteristics 1897
37.3.2 Supply Current Characteristics 1903
37.4 AC Characteristics 1909
37.4.1 Basic Operation 1909
37.5 Peripheral Functions Characteristics 1913
37.5.1 Serial Array Unit 1913
37.5.2 Serial Interface IICA 1932
37.5.3 On-chip Debug (UART) 1933
37.5.4 LIN/UART Module (RLIN3) UART Mode 1933
37.5.5 CAN-FD Communication Interface (RS-CANFD lite) Timing 1933
37.6 Analog Characteristics 1934
37.6.1 A/D Converter Characteristics 1934
37.6.2 D/A Converter Characteristics 1935
37.6.3 Comparator Characteristics 1935
37.6.4 POR Circuit Characteristics 1935
37.6.5 LVD Circuit Characteristics 1936
37.7 Power Supply Voltage Rising Time 1937
37.8 Regulator Output Voltage Characteristics 1937
37.9 RAM Data Retention Characteristics 1937
37.10 Flash Memory Programming Characteristics 1938
37.11 Dedicated Flash Memory Programmer Communication (UART) 1939
37.12 Timing of Entry to Flash Memory Programming Modes 1940

CHAPTER 38 ELECTRICAL SPECIFICATIONS (RL78/F22) 1941
38.1 Absolute Maximum Ratings 1942
38.2 Oscillator Characteristics 1945

38.2.1 Main System Clock Oscillator Characteristics 1945
38.2.2 On-chip Oscillator Characteristics 1946
38.2.3 Subsystem Clock Oscillator Characteristics 1947
38.2.4 PLL Circuit Characteristics 1948
38.3 DC Characteristics 1949
38.3.1 Pin Characteristics 1949
38.3.2 Supply Current Characteristics 1955
38.4 AC Characteristics 1961
38.4.1 Basic Operation 1961
38.5 Peripheral Functions Characteristics 1965
38.5.1 Serial Array Unit 1965
38.5.2 Serial Interface IICA 1984
38.5.3 On-chip Debug (UART) 1985
38.5.4 LIN/UART Module (RLIN3) UART Mode 1985
38.6 Analog Characteristics 1986
38.6.1 A/D Converter Characteristics 1986
38.6.2 POR Circuit Characteristics 1988
38.6.3 LVD Circuit Characteristics 1989
38.7 Power Supply Voltage Rising Time 1990
38.8 Regulator Output Voltage Characteristics 1990
38.9 RAM Data Retention Characteristics 1990
38.10 Flash Memory Programming Characteristics 1991
38.11 Dedicated Flash Memory Programmer Communication (UART) 1992
38.12 Timing of Entry to Flash Memory Programming Modes 1993

CHAPTER 39 PACKAGE DRAWINGS 1994

39.1 24-pin Products 1994
39.2 32-pin Products 1995
39.3 48-pin Products 1996
39.4 64-pin Products 1997
39.5 80-pin Products 1998
39.6 100-pin Products 1999

APPENDIX A REVISION HISTORY 2000

A.1 Major Revision History from Rev.1.10 to Rev.1.20 2000
A.2 Revision History of Preceding Editions 2001

CHAPTER 1 OVERVIEW

1.1 Features

- Minimum instruction execution time can be changed from high speed (0.025 µs: @ 40 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: 128 KB/512 KB
- RAM: 12 KB/40 KB
- Data flash memory: 8 KB/16 KB
- High-speed on-chip oscillator clock
  - Selectable from 40 MHz (Typ.), 32 MHz (Typ.), 20 MHz (Typ.), 16 MHz (Typ.), 8 MHz (Typ.), 4 MHz (Typ.), and 2 MHz (Typ.) (Selectable from 80 MHz (Typ.) and 64 MHz (Typ.) when using Timer RDe and RS-CANFD lite **Note 1**)
- 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
- 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: 18 to 90 (including one input-only pin)
- Timer
  - 16-bit timer array unit: 12 channels/16 channels
  - 16-bit timer RDe: 2 channels (with PWMOPA and Dithering / Gate function)
  - 16-bit timer RJ: 1 channel
  - Watchdog timer: 1 channel
  - Real-time clock: 1 channel
- Application accelerator unit
- Serial interface
  - CSI
  - UART/UART (LIN-bus supported)
  - LIN module (master/slave supported)
  - I^2C/simplified I^2C
  - Simplified I^2S
  - CAN interface (RS-CANFD lite) **Note 2**

- 12-bit resolution A/D converter: 4 to 29 channels
- DTC (Max. 49 sources)
- ELC (Max. 26 channels for event link source, Max. 10 channels for event link destination) Note 2
- Functional safety (CRC calculation, Clock monitor, A/D test, etc.)
- Security functions (Secure boot, Crypto engine (AES-128, 192, 256), Random Number Generator (TRNG))
- 8-bit D/A converter Note 2
- On-chip comparator: 1 unit (input pin: 4 channels) Note 2
- Capacitive sensing unit: 7 to 31 channels
- Power supply voltage: RL78/F22 V_{DD} = 1.8 to 5.5 V, RL78/F25 V_{DD} = 2.7 to 5.5 V
- Operating ambient temperature:
  T_A = −40°C to 105°C (grade-3) / T_A = −40°C to 125°C (grade-4)
- ASIL level: ASIL-B

Notes 1. f_{IH} cannot be used as a RS-CANFD lite communication clock.
2. Only available in the RL78/F25.

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/F22, F25 Lineup (Grade-3)

Operating Temperature (T_A)	Package	Pin	RL78/F22	RL78/F25
			Code Flash/Data Flash/RAM 128 KB/8 KB/12 KB	Code Flash/Data Flash/RAM 512 KB/16 KB/40 KB
−40°C to 105°C	QFN	24	R7F122F7G3ANP-C	–
		32	R7F122FBG3ANP-C	–
	QFP	48	R7F122FGG3AFB-C	R7F125FGL3AFB-C
		64	–	R7F125FLL3AFB-C
		80	–	R7F125FML3AFB-C
		100	–	R7F125FPL3AFB-C

Table 1-2. RL78/F22, F25 Lineup (Grade-4)

Operating Temperature (T_A)	Package	Pin	RL78/F22	RL78/F25
			Code Flash/Data Flash/RAM 128 KB/8 KB/12 KB	Code Flash/Data Flash/RAM 512 KB/16 KB/40 KB
−40°C to 125°C	QFN	24	R7F122F7G4ANP-C	–
		32	R7F122FBG4ANP-C	–
	QFP	48	R7F122FGG4AFB-C	R7F125FGL4AFB-C
		64	–	R7F125FLL4AFB-C
		80	–	R7F125FML4AFB-C
		100	–	R7F125FPL4AFB-C

1.3 Function Overview

1.3.1 RL78/F25 Functions List

Table 1-3. RL78/F25 Functions List (1/2)

Function Items			Series Name	R7F125FPL	R7F125FML	R7F125FLL	R7F125FGL
			Pin Count	100 pins	80 pins	64 pins	48 pins
Code flash				512 KB			
Data flash				16 KB			
RAM				40 KB			
Supply voltage range				2.7 V to 5.5 V			
Maximum operation frequency				40 MHz			
System clock	Main system clock oscillator	Crystal / ceramic / square wave		2 to 20 MHz (operating at 2.7 V to 5.5 V)			
	High-speed on-chip oscillator	Normal high accuracy		40 MHz (Typ.)			
	Low-speed on-chip oscillator	For low-speed operation		15 kHz (Typ.)			
	Subsystem clock oscillator			32.768 kHz			
	PLL			Yes			
Clock for peripherals	Low-speed on-chip oscillator	For peripherals other than WDT		15 kHz (Typ.)			
		For WDT		15 kHz (Typ.)			
POR				1.50 V (Typ.)			
LVD	V_{DD} voltage detection	LVD0		When power supply is rising: 2.97 V and 3.96 V (2 steps) When power supply is falling: 2.91 V and 3.88 V (2 steps)			
		LVD1		When power supply is rising: 2.82 V to 4.16 V (8 steps) When power supply is falling: 2.76 V to 4.08 V (8 steps)			
Functional safety **Note 6**	WWDT (window watchdog timer)			Yes			
	Flash memory fast CRC operation function			Yes			
	General purpose CRC operation			Yes			
	Code flash memory 1-bit error correction function			Yes			
	Code flash memory 2-bit error detection function			Yes			
	RAM 1-bit error correction function			Yes			
	RAM 2-bit error detection function			Yes			
	RS-CANFD lite RAM 1-bit error correction function			Yes			
	RS-CANFD lite RAM 2-bit error detection function			Yes			
	Invalid memory access detection function			Yes			
	Frequency detection function			Yes			
	Clock monitor function			Yes			
	Stack pointer monitor function			Yes			
	A/D test function			Yes			
I/O ports	Input/Output	CMOS		84 ch	66 ch	50 ch	36 ch
	Output	CMOS		1 ch			
	Input	Shared with oscillator pins		4 ch			
		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 (A/D, D/A, COMP)			AV_{DD}, AV_{SS} (AV_{REFP}, AV_{REFM} for A/D)			
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			17 ch **Notes 3, 4**	17 ch **Notes 3, 4**	16 ch **Notes 2, 4**	15 ch **Note 1, 5**
	Internal			63 ch **Note 3**	63 ch **Note 3**	63 ch **Note 2**	63 ch **Note 1**
Key return detection				8 ch			
DTC				49 sources			
Timer	TAU			16 bits (8 ch × 2)			
	RTC			1 ch			
	Timer RJ			16 bits × 1			
	Timer RDe			16 bits × 2 (with PWMOPA and dithering / gate function)			

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

Table 1-3. RL78/F25 Functions List (2/2)

Function Items		Series Name	R7F125FPL	R7F125FML	R7F125FLL	R7F125FGL
		Pin Count	100 pins	80 pins	64 pins	48 pins
Serial I/F	CSI / simplified I^2C / UART		4 ch / 4 ch / 2 ch			
		SPI	Yes			
	Multimaster I^2C		1 ch			
	Simplified I^2S		1 ch			
	LIN/UART module (RLIN3)		3 ch			
	CAN interface (RS-CANFD lite)		2 ch			
A/D converter 12 bit	High speed		16 ch	16 ch	15 ch	11 ch
	Normal speed		13 ch	7 ch	7 ch	6 ch
	Internal		1 ch (Internal reference voltage)			
D/A converter	8-bit		1 ch			
Comparator			1 unit (input 4 ch)			
Capacitive sensing unit			31 ch	31 ch	23 ch	15 ch
ELC			Link source: 26 ch Link destination: 10 ch			
PCLBUZ			1 ch			
Application accelerator unit			Yes			
Self-programming			Yes			
On-chip debug	Trace		Yes			
	Hot plug-in		Yes			
Option byte			Yes			
Security functions	AESEA		ECB/CBC mode and CMAC (AES-128, 192, 256)			
	Random number generator (TRNG)		Yes			

Notes 1. The following pairs of internal and external sources are each counted as a single source in this number:
INTP4 / INTSPM, INTP5 / INTCMP0, INTP6 / INTTM11H, INTP7 / INTTM13H,
INTP8 / INTRTC, INTP9 / INTTM01H

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

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

4. Both sources in the following pairs are counted as a single external interrupt source in this number:
INTP11 / INTLIN0WUP, INTP12 / INTLIN1WUP, INTP6 / INTLIN2WUP

5. Both sources in the following pairs are counted as a single external interrupt source in this number:
INTP6 / INTLIN2WUP

6. These functions are provided but they are not Safety Mechanism.
- Illegal instruction execution detection function
- SFR/RAM guard function
- I/O port output signal level detection function

1.3.2 RL78/F22 Functions List

Table 1-4. RL78/F22 Functions List (1/2)

Function Items			Series Name	R7F122FGG	R7F122FBG	R7F122F7G
			Pin Count	48 pins	32 pins	24 pins
Code flash				128 KB		
Data flash				8 KB		
RAM				12 KB		
Supply voltage range				1.8 V to 5.5 V		
Maximum operation frequency				40 MHz		
System clock	Main system clock oscillator	Crystal / ceramic / square wave		2 to 20 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V operation), 2 to 16 MHz (1.8 V ≤ V_{DD} < 2.7 V operation) *Grade 3 products only		
	High-speed on-chip oscillator	Normal high accuracy		40 MHz (Typ.)		
	Low-speed on-chip oscillator	For low-speed operation		15 kHz (Typ.)		
	Subsystem clock oscillator			32.768 kHz	None	
	PLL			Yes		
Clock for peripherals	Low-speed on-chip oscillator	For peripherals other than WDT		15 kHz (Typ.)		
		For WDT		15 kHz (Typ.)		
POR				1.50 V (Typ.)		
LVD	V_{DD} voltage detection	LVD0		When power supply is rising: 1.90 V to 3.96 V (3 steps) When power supply is falling: 1.86 V to 3.88 V (3 steps)		
		LVD1		When power supply is rising: 1.88 V to 4.16 V (11 steps) When power supply is falling: 1.84 V to 4.08 V (11 steps)		
Functional safety **Note 3**	WWDT (window watchdog timer)			Yes		
	Flash memory fast CRC operation function			Yes		
	General purpose CRC operation			Yes		
	Code flash memory 1-bit error correction function			Yes		
	Code flash memory 2-bit error detection function			Yes		
	RAM 1-bit error correction function			Yes		
	RAM 2-bit error detection function			Yes		
	Invalid memory access detection function			Yes		
	Frequency detection function			Yes		
	Clock monitor function			Yes		
	Stack pointer monitor function			Yes		
	A/D test function			Yes		
I/O ports	Input/Output	CMOS		36 ch	23 ch	15 ch
	Output	CMOS		1 ch	None	
	Input	Shared with oscillator pins		4 ch	2 ch	
		Input only		1 ch		
Power supply pins	For internal circuits			V_{DD}, V_{SS}, REGC		
	For I/O ports			None		
	For analog circuits (A/D)			AV_{DD}, AV_{SS} (AV_{REFP}, AV_{REFM} for A/D)		
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			12 ch **Note 2**	8 ch **Note 1**	7 ch **Note 1**
	Internal			39 ch **Note 2**	39 ch **Note 1**	39 ch **Note 1**
Key return detection				8 ch	6 ch	3 ch
DTC				37 sources	36 sources	35 sources
Timer	TAU			16 bits (8 ch + 4 ch)		
	RTC			1 ch		
	Timer RJ			16 bits × 1		
	Timer RDe			16 bits × 2 (with PWMOPA and dithering / gate function)		

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

Table 1-4. RL78/F22 Functions List (2/2)

Function Items		Series Name	R7F122FGG	R7F122FBG	R7F122F7G
		Pin Count	48 pins	32 pins	24 pins
Serial I/F	CSI/simplified I^2C /UART		4 ch / 4 ch / 2 ch	3 ch / 3 ch / 2 ch	2 ch / 2 ch / 2 ch
		SPI	Yes		
	Multimaster I^2C		1 ch		
	Simplified I^2S		1 ch		
	LIN/UART module (RLIN3)		1 ch		
	CAN interface (RS-CANFD lite)		None		
A/D converter 12 bit	High Speed		11 ch	6 ch	3 ch
	Normal Speed		6 ch	2 ch	1 ch
	Internal		1 ch (Internal reference voltage)		
D/A converter	8-bit		None		
Comparator			None		
Capacitive sensing unit			12 ch	10 ch	7 ch
ELC			None		
PCLBUZ			1 ch	None	
Application accelerator unit			Yes		
Self-programming			Yes		
On-chip debug	Trace		Yes		
	Hot plug-in		Yes		
Option byte			Yes		
Security Functions	AESEA		ECB/CBC mode and CMAC (AES-128, 192, 256)		
	Random Number Generator (TRNG)		Yes		

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

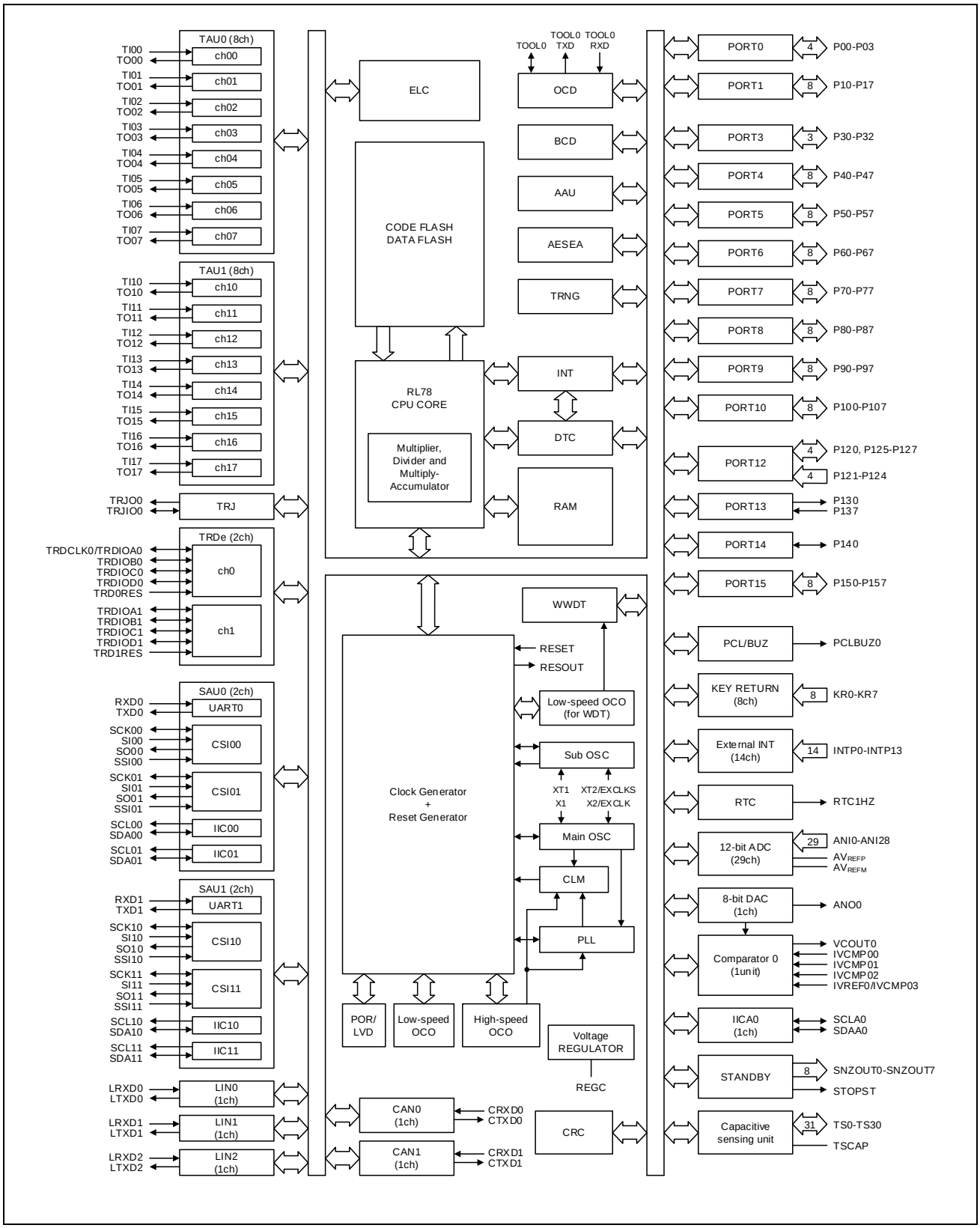
2. The following pairs of internal and external sources are each counted as a single source in this number:
INTP4 / INTSPM, INTP6 / INTTM11H, INTP7 / INTTM13H, INTP8 / INTRTC, INTP9 / INTTM01H

3. These functions are provided but they are not Safety Mechanism.
- Illegal instruction execution detection function
- SFR/RAM guard function
- I/O port output signal level detection function

1.4 Block Diagram

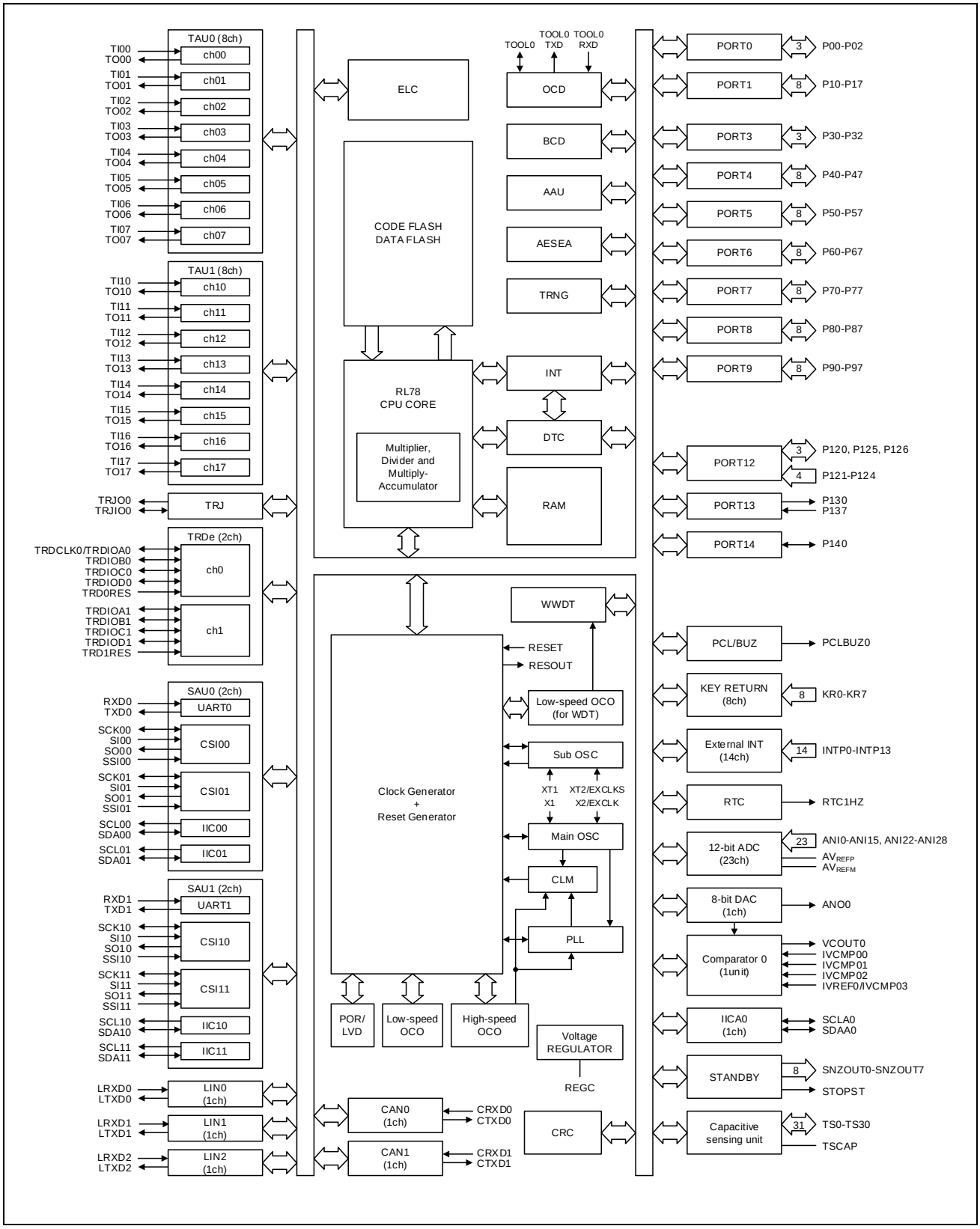
1.4.1 RL78/F25: Block Diagram of R7F125FPL 100-pin Products

Figure 1-1. Block Diagram



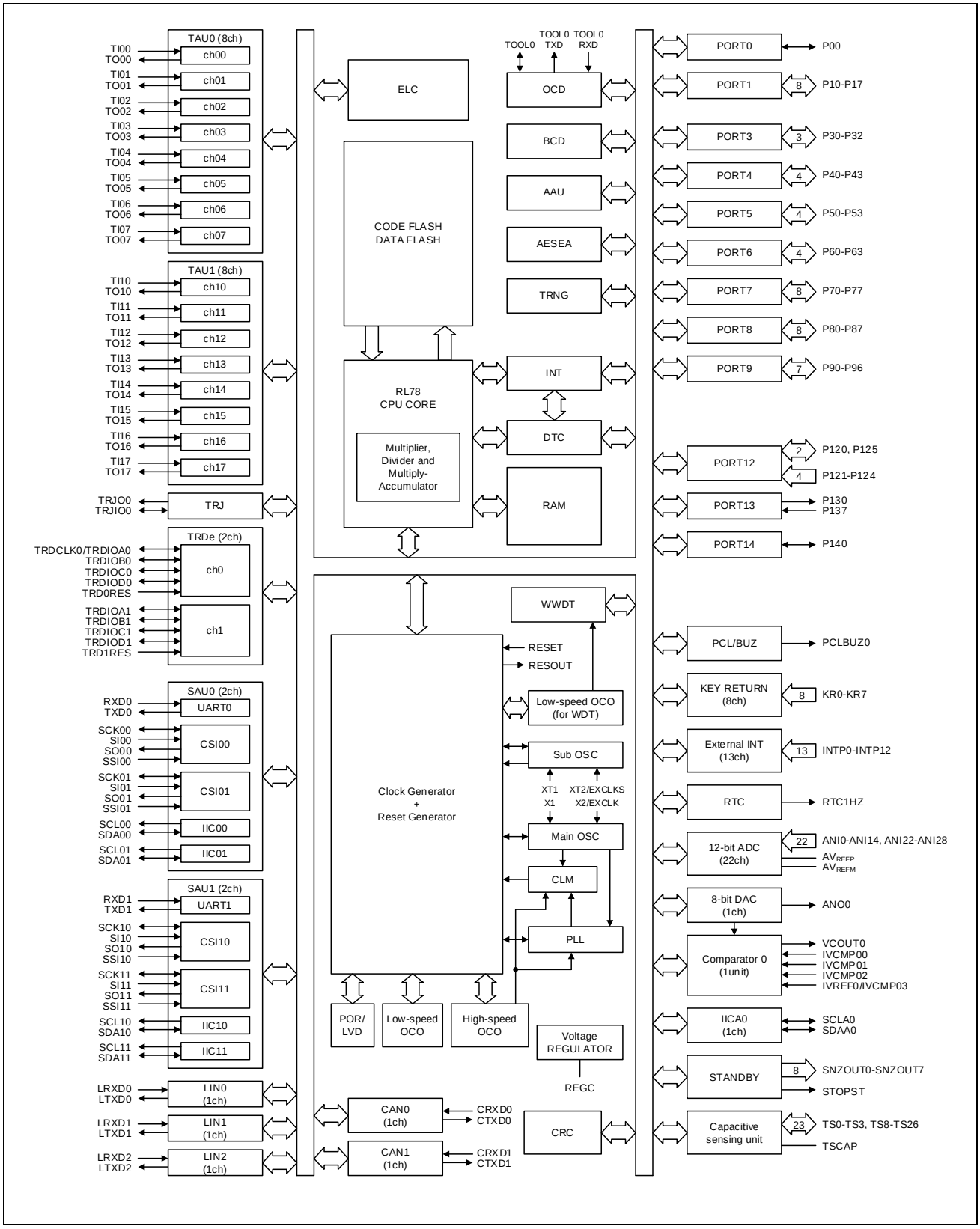
1.4.2 RL78/F25: Block Diagram of R7F125FML 80-pin Products

Figure 1-2. Block Diagram



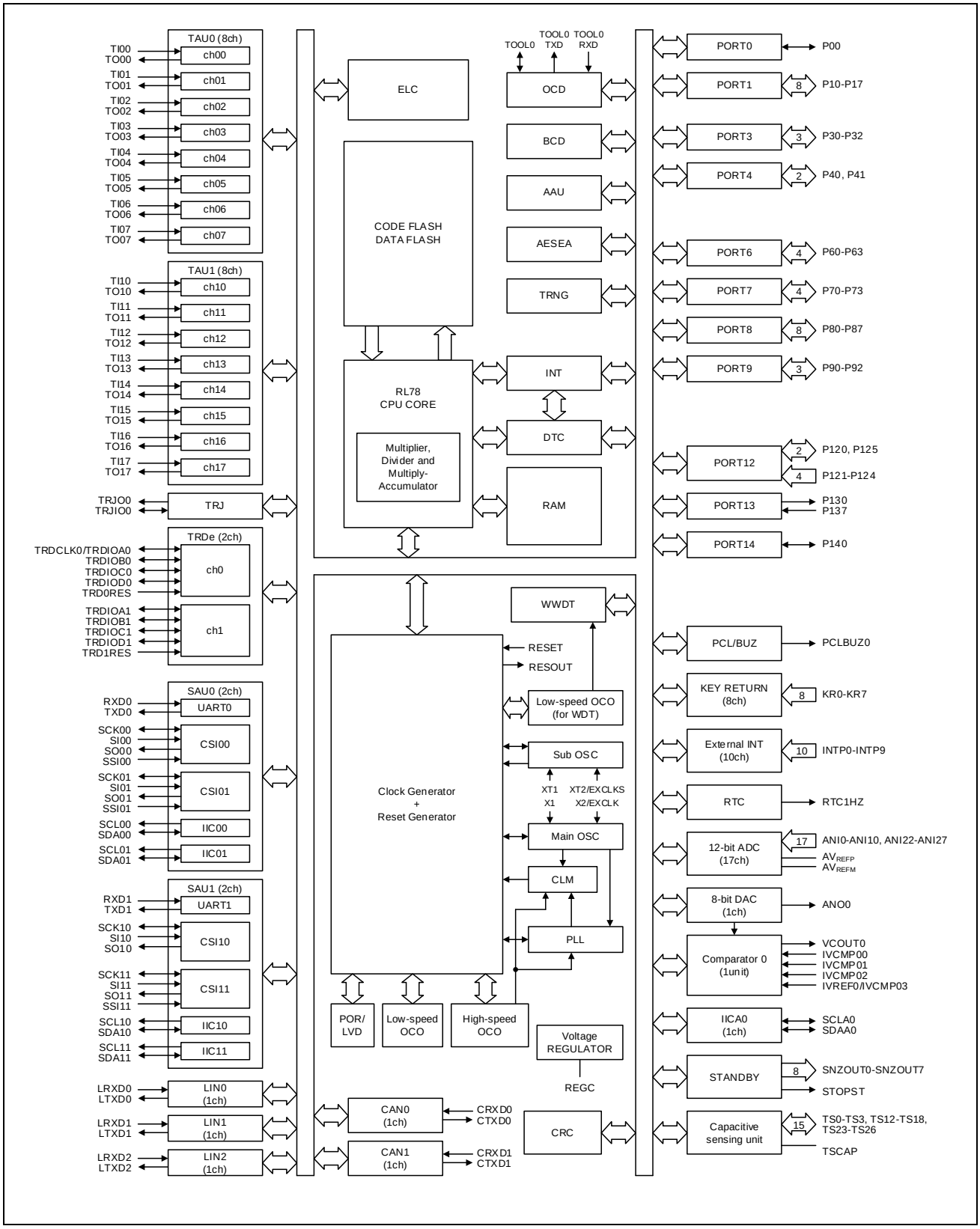
1.4.3 RL78/F25: Block Diagram of R7F125FLL 64-pin Products

Figure 1-3. Block Diagram



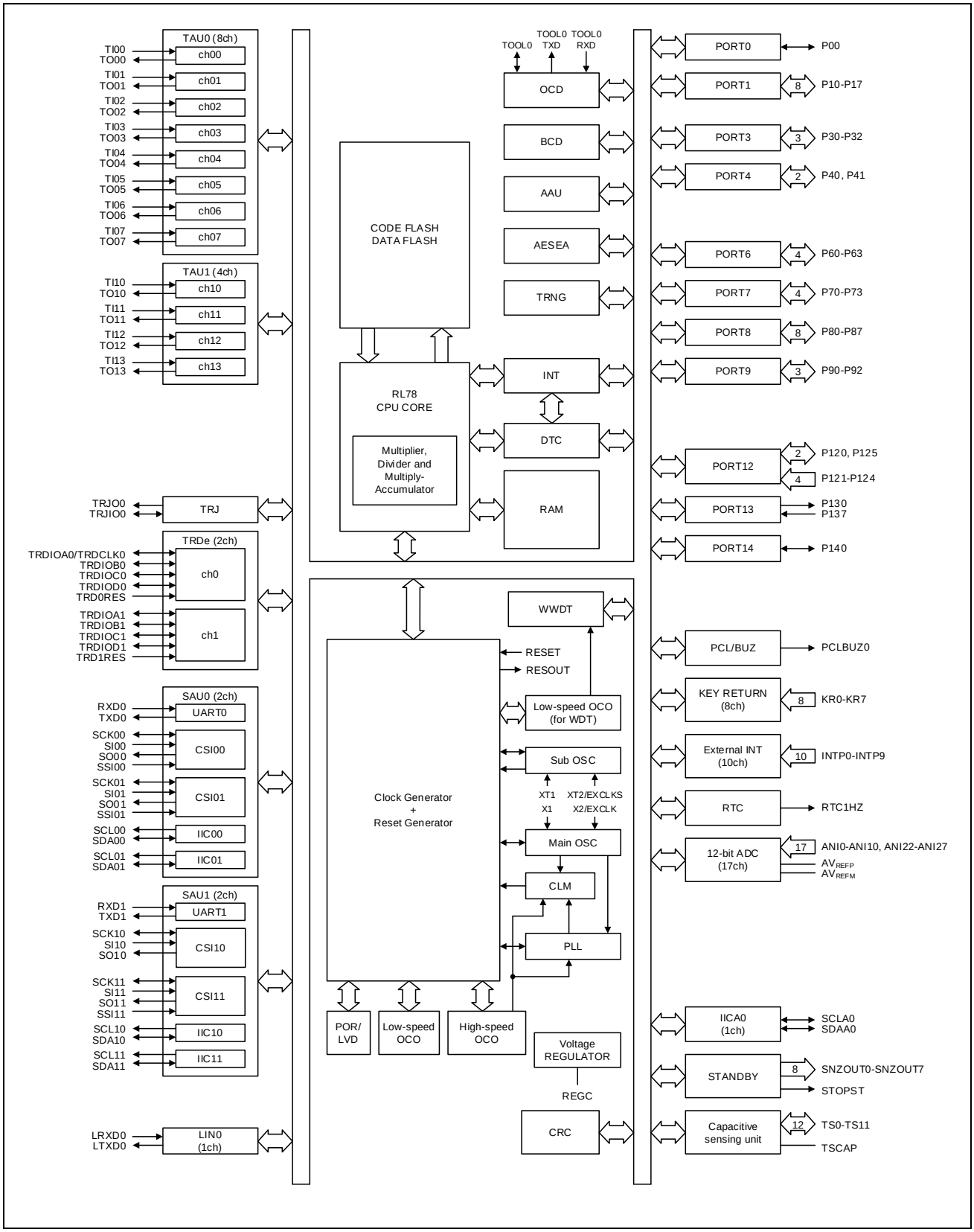
1.4.4 RL78/F25: Block Diagram of R7F125FGL 48-pin Products

Figure 1-4. Block Diagram



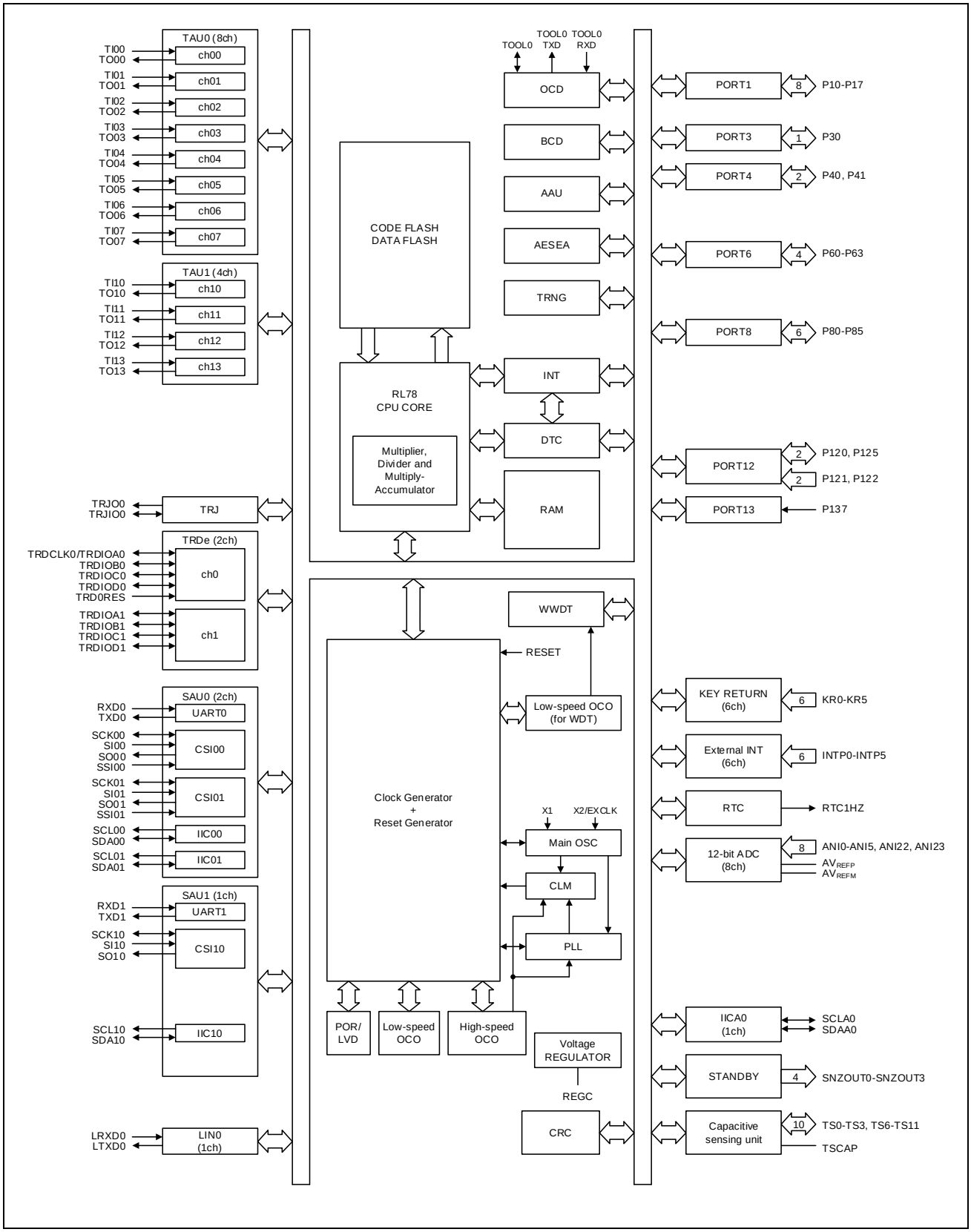
1.4.5 RL78/F22: Block Diagram of R7F122FGG 48-pin Products

Figure 1-5. Block Diagram



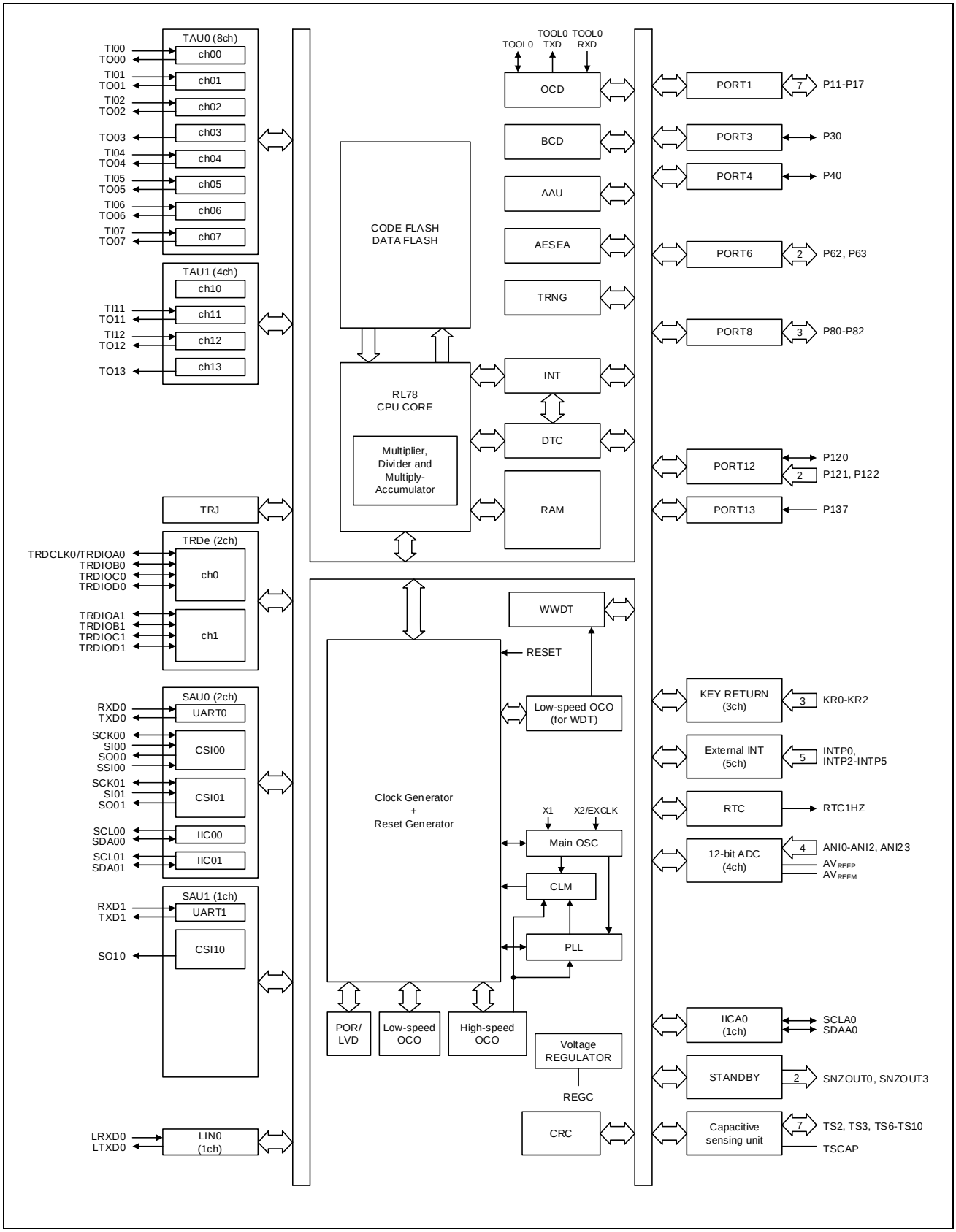
1.4.6 RL78/F22: Block Diagram of R7F122FBG 32-pin Products

Figure 1-6. Block Diagram



1.4.7 RL78/F22: Block Diagram of R7F122F7G 24-pin Products

Figure 1-7. Block Diagram

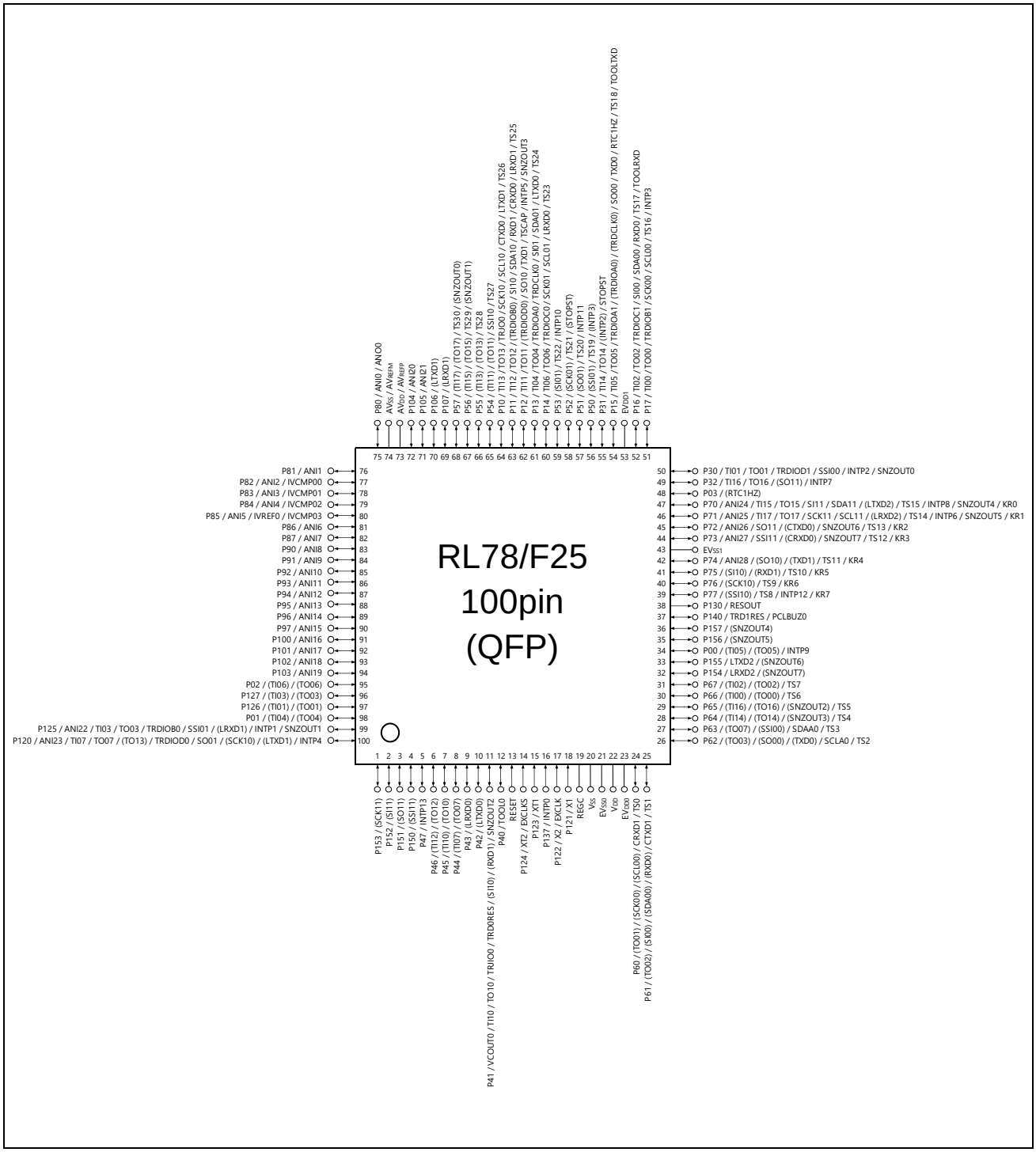


1.5 Pin Configurations

1.5.1 RL78/F25 Pin Configuration for 100-pin Products

- RL78/F25: 100-pin Plastic QFP (Fine Pitch) (14 × 14)

Figure 1-8. RL78/F25 Pin Configuration for 100-pin Products

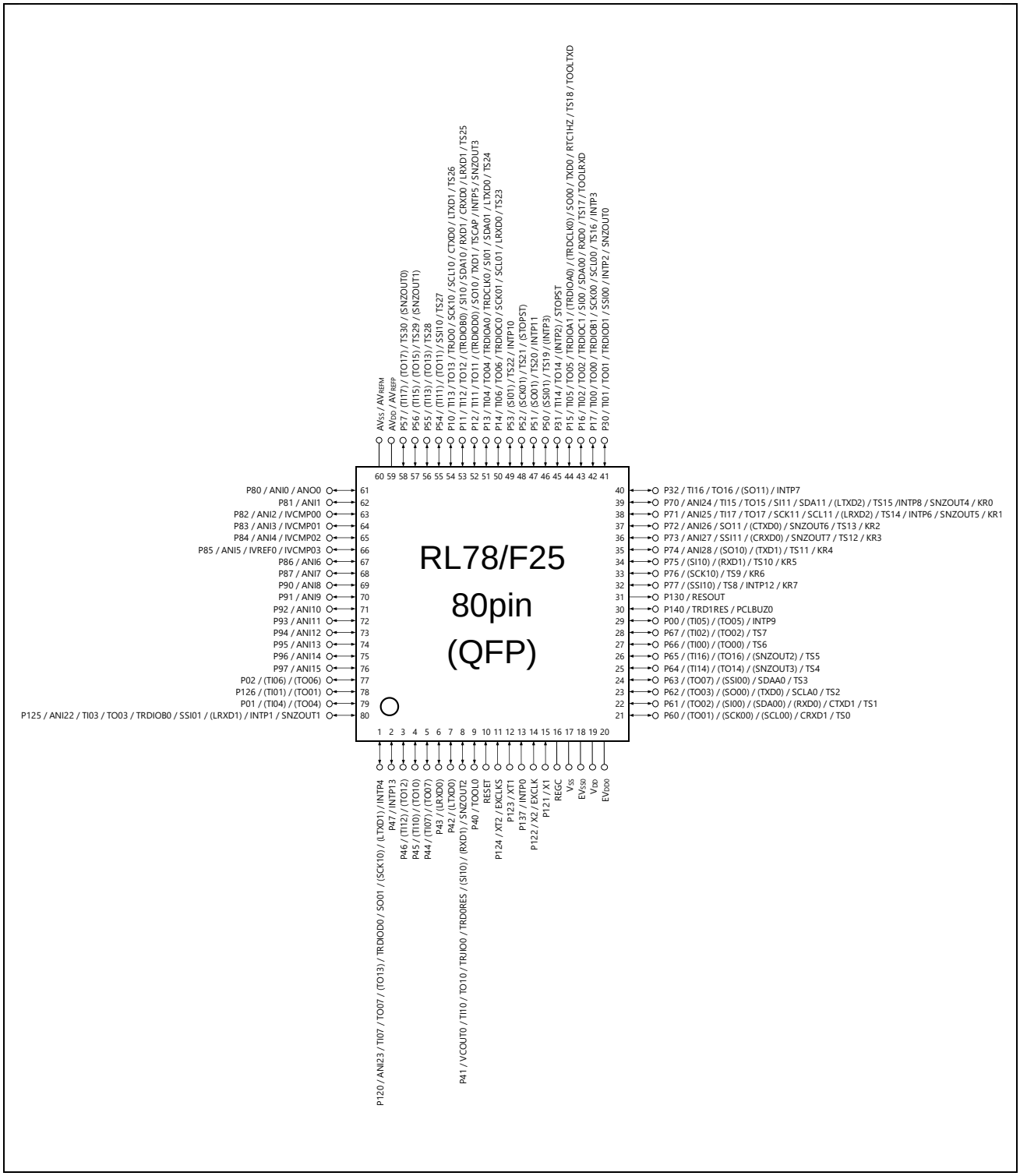


Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection registers (PIORx). Only the STOPST function of P52 can be assigned via settings in the STOP status output control register (STPSTC).

1.5.2 RL78/F25 Pin Configuration for 80-pin Products

- RL78/F25: 80-pin Plastic QFP (Fine Pitch) (12 × 12)

Figure 1-9. RL78/F25 Pin Configuration for 80-pin Products

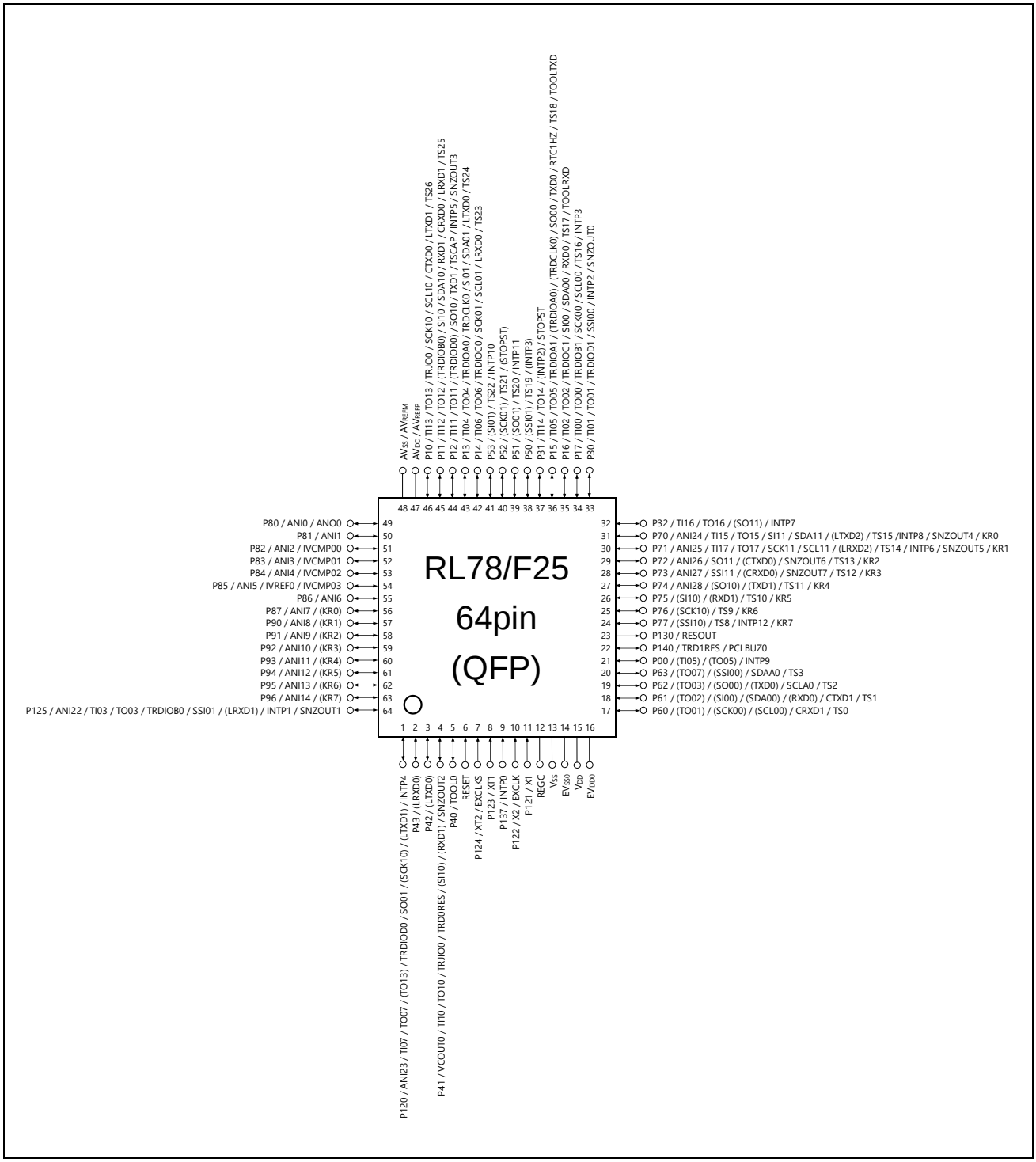


Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection registers (PIORx). Only the STOPST function of P52 can be assigned via settings in the STOP status output control register (STPSTC).

1.5.3 RL78/F25 Pin Configuration for 64-pin Products

- RL78/F25: 64-pin Plastic QFP (Fine Pitch) (10 × 10)

Figure 1-10. RL78/F25 Pin Configuration for 64-pin Products

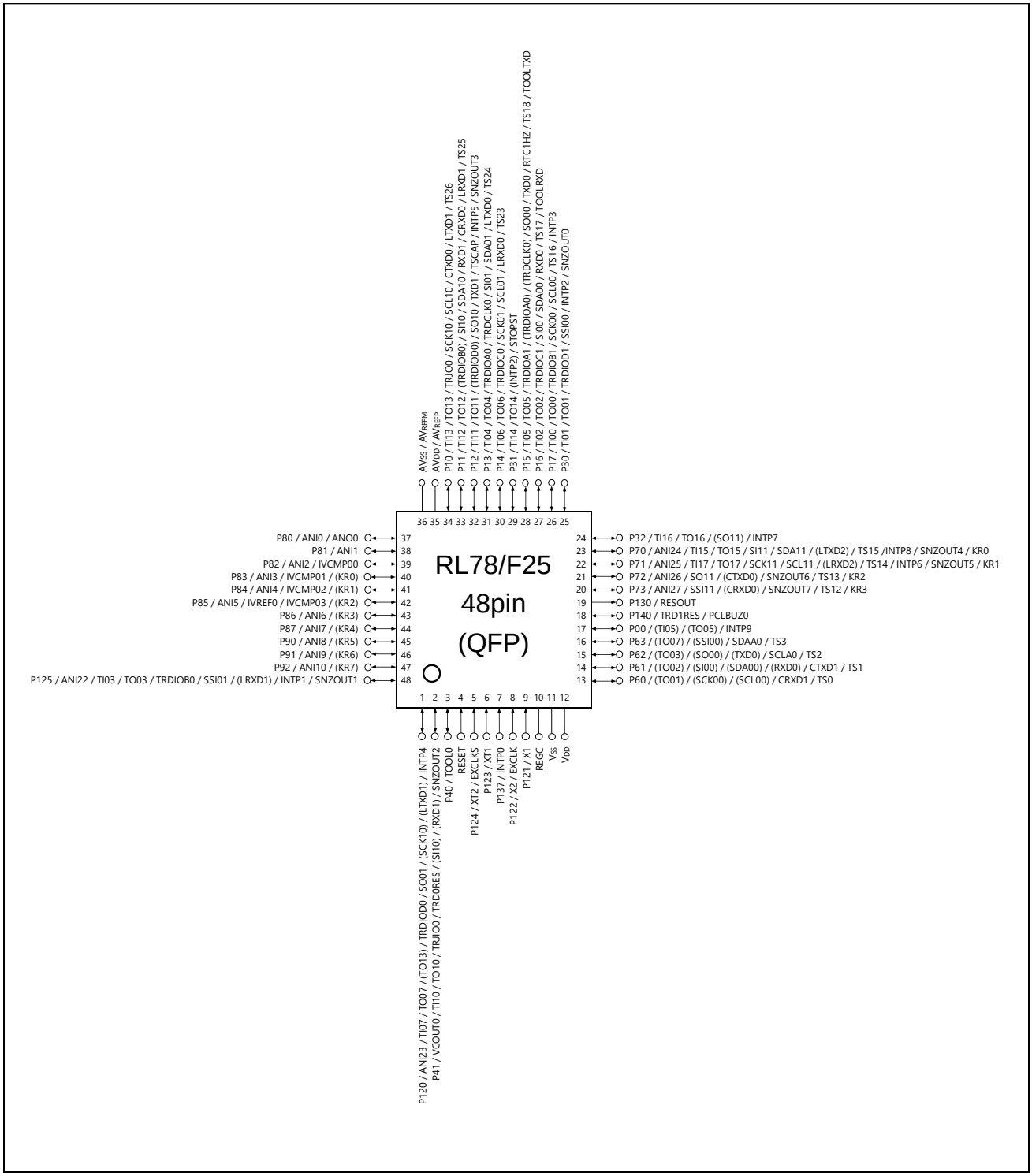


Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection registers (PIORx). Only the STOPST function of P52 can be assigned via settings in the STOP status output control register (STPSTC).

1.5.4 RL78/F25 Pin Configuration for 48-pin Products

- RL78/F25: 48-pin Plastic QFP

Figure 1-11. RL78/F25 Pin Configuration for 48-pin Products

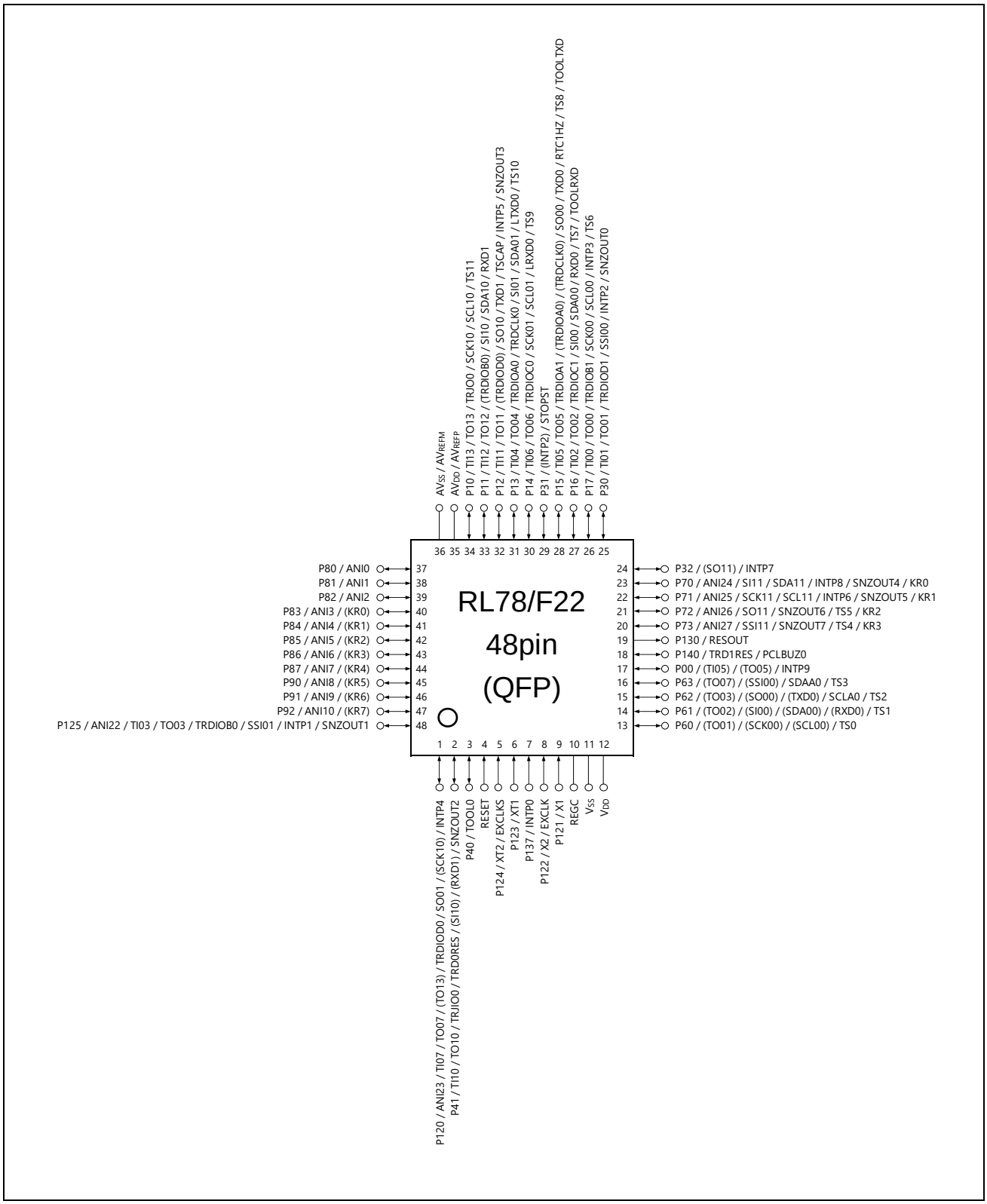


Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection registers (PIORx).

1.5.5 RL78/F22 Pin Configuration for 48-pin Products

- RL78/F22: 48-pin Plastic QFP

Figure 1-12. RL78/F22 Pin Configuration for 48-pin Products

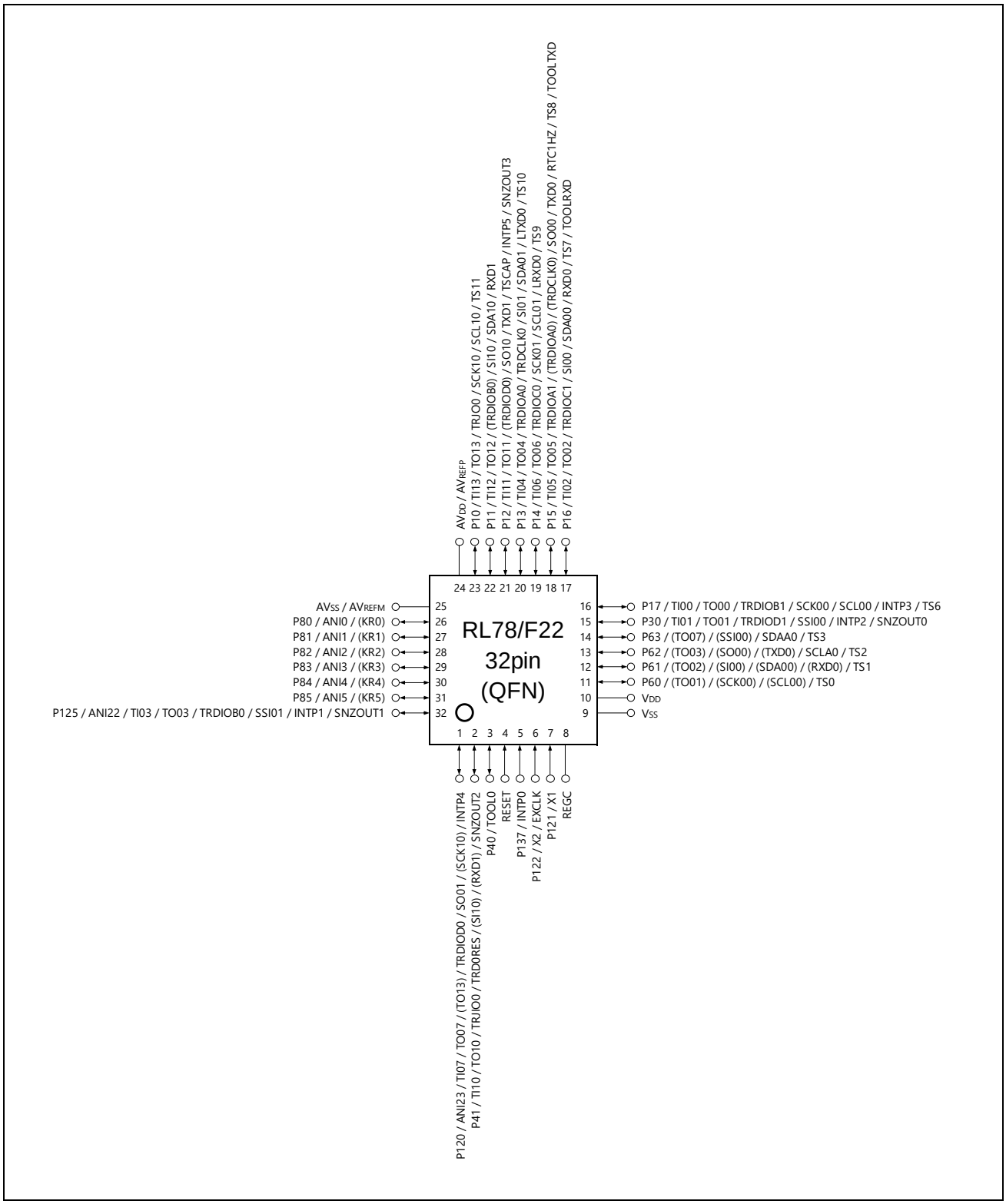


Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection registers (PIORx).

1.5.6 RL78/F22 Pin Configuration for 32-pin Products

- RL78/F22: 32-pin Plastic QFN

Figure 1-13. RL78/F22 Pin Configuration for 32-pin Products

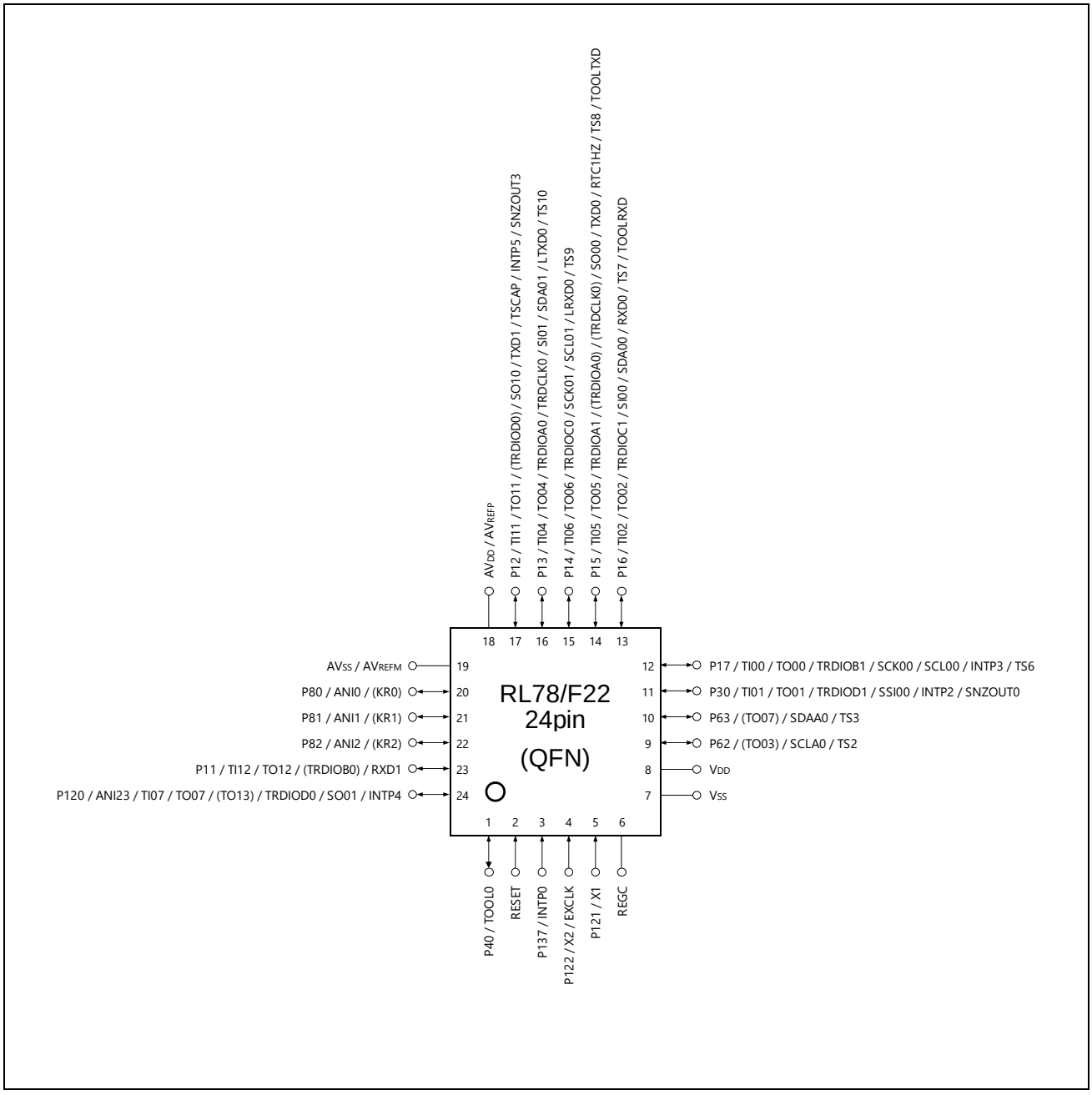


Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection registers (PIORx).

1.5.7 RL78/F22 Pin Configuration for 24-pin Products

- RL78/F22: 24-pin Plastic QFN

Figure 1-14. RL78/F22 Pin Configuration for 24-pin Products



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

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

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

Tabel 2-1. Pin I/O Buffer Power Supplies

(1) 24-pin products

Power Supply	Corresponding Pins
V_{DD}	• Port pins other than P80 to P82
AV_{DD}	• P80 to P82

(2) 32-pin products

Power Supply	Corresponding Pins
V_{DD}	• Port pins other than P80 to P85
AV_{DD}	• P80 to P85

(3) 48-pin products

Power Supply	Corresponding Pins
V_{DD}	• Port pins other than P80 to P87, P90 to P92
AV_{DD}	• P80 to P87, P90 to P92

(4) 64-pin products

Power Supply	Corresponding Pins
EV_{DD0}	• Port pins other than P80 to P87, P90 to P96, P121 to P124, P137
V_{DD}	• P121 to P124, P137 • Pins other than ports
AV_{DD}	• P80 to P87, P90 to P96

(5) 80-pin products

Power Supply	Corresponding Pins
EV_{DD0}	• Port pins other than P80 to P87, P90 to P97, P121 to P124, P137
V_{DD}	• P121 to P124, P137 • Pins other than ports
AV_{DD}	• P80 to P87, P90 to P97

(6) 100-pin products

Power Supply	Corresponding Pins
EV_{DD0}, EV_{DD1}	• Port pins other than P80 to P87, P90 to P97, P100 to P105, P121 to P124, P137
V_{DD}	• P121 to P124, P137 • Pins other than ports
AV_{DD}	• P80 to P87, P90 to P97, P100 to P105

This subchapter describes the 100-pin products of RL78/F25 and the 48-pin products of RL78/F22 as examples.

2.1.1 RL78/F25 100-pin Products

(1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0 • 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 • Input of P10, P11, P13, P14, P16, and P17 can be set to TTL input buffer. • Use of an on-chip pull-up resistor can be specified by a software setting. • Output from P10 to P17 can be set to N-ch open-drain output. • For input to P10, P11, P13, P14, P16, and P17, the threshold level can be specified.	Input port	TI13/TO13/TRJO0/SCK10/SCL10/ LTXD1/CTXD0/TS26
P11				TI12/TO12/(TRDIOB0)/SI10/ SDA10/RXD1/LRXD1/CRXD0/TS25
P12				TI11/TO11/(TRDIOD0)/INTP5/ SO10/TXD1/SNZOUT3/TSCAP
P13				TI04/TO04/TRDIOA0/TRDCLK0/ SI01/SDA01/LTXD0/TS24
P14				TI06/TO06/TRDIOC0/SCK01/ SCL01/LRXD0/TS23
P15				TI05/TO05/TRDIOA1/(TRDIOA0)/ (TRDCLK0)/SO00/TXD0/ TOOLTXD/RTC1HZ/TS18
P16				TI02/TO02/TRDIOC1/SI00/ SDA00/RXD0/TOOLRXD/TS17
P17				TI00/TO00/TRDIOB1/SCK00/ SCL00/INTP3/TS16
P30	I/O	Port 3 • Input of P30 can be set to TTL input buffer. • Output from P32 can be set to N-ch open-drain output. • For input to P30 to P32, use of an on-chip pull-up resistor can be specified by a software setting. • For input to P30, the threshold level can be specified.	Input port	TI01/TO01/TRDIOD1/SSI00/ INTP2/SNZOUT0
P31				TI14/TO14/STOPST/(INTP2)
P32				TI16/TO16/(SO11)/INTP7
P40	I/O	Port 4 • Use of an on-chip pull-up resistor can be specified by a software setting. • For input to P41 and P43, the threshold level can be specified.	Input port	TOOL0
P41				TI10/TO10/TRJIO0/TRD0RES/ (SI10)/(RXD1)/VCOUT0/SNZOUT2
P42				(LTXD0)
P43				(LRXD0)
P44				(TI07)/(TO07)
P45				(TI10)/(TO10)
P46				(TI12)/(TO12)
P47				INTP13
P50	I/O	Port 5 • Input of P54 can be set to TTL input buffer. • Use of an on-chip pull-up resistor can be specified by a software setting. • For input to P50 and P52 to P54, the threshold level can be specified.	Input port	(SSI01)/(INTP3)/TS19
P51				(SO01)/INTP11/TS20
P52				(SCK01)/(STOPST)/TS21
P53				(SI01)/INTP10/TS22
P54				(TI11)/(TO11)/SSI10/TS27
P55				(TI13)/(TO13)/TS28
P56				(TI15)/(TO15)/(SNZOUT1)/TS29
P57				(TI17)/(TO17)/(SNZOUT0)/TS30
P60	I/O	Port 6 • Input of P62 and P63 can be set to TTL input buffer. Use of an on-chip pull-up resistor can be specified by a software setting. • Output from P60 to P63 can be set to N-ch open-drain output. • For input to P60 to P63, the threshold level can be specified.	Input port	(TO01)/(SCK00)/(SCL00)/CRXD1/TS0
P61				(TO02)/(SI00)/(SDA00)/(RXD0)/ CTXD1/TS1
P62				(TO03)/(SO00)/(TXD0)/SCLA0/TS2
P63				(TO07)/(SSI00)/SDAA0/TS3
P64				(TI14)/(TO14)/(SNZOUT3)/TS4
P65				(TI16)/(TO16)/(SNZOUT2)/TS5
P66				(TI00)/(TO00)/TS6
P67				(TI02)/(TO02)/TS7

Remark Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIORx). 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 • Input of P70, P71, and P73 can be set to TTL input buffer. • P70 to P74 can be set to analog input. • Use of an on-chip pull-up resistor can be specified by a software setting. • Output from P70 to P72 can be set to N-ch open-drain output. • For input to P70, P71, P73, and P75 to P77, the threshold level can be specified.	Analog input port	ANI24/KR0/TI15/TO15/INTP8/SI11/SDA11/(LTXD2)/SNZOUT4/TS15
P71				ANI25/KR1/TI17/TO17/INTP6/SCK11/SCL11/(LRXD2)/SNZOUT5/TS14
P72				ANI26/KR2/(CTXD0)/SO11/SNZOUT6/TS13
P73				ANI27/KR3/(CRXD0)/SSI11/SNZOUT7/TS12
P74				ANI28/KR4/(SO10)/(TXD1)/TS11
P75			Input port	KR5/(SI10)/(RXD1)/TS10
P76				KR6/(SCK10)/TS9
P77				KR7/(SSI10)/INTP12/TS8
P80	I/O	Port 8 • P80 to P87 can be set to analog input.	Analog input port	ANI0/ANO0
P81				ANI1
P82				ANI2/IVCMP00
P83				ANI3/IVCMP01
P84				ANI4/IVCMP02
P85				ANI5/IVCMP03/IVREF0
P86				ANI6
P87				ANI7
P90	I/O	Port 9 • P90 to P97 can be set to analog input.	Analog input port	ANI8
P91				ANI9
P92				ANI10
P93				ANI11
P94				ANI12
P95				ANI13
P96				ANI14
P97				ANI15
P100	I/O	Port 10 • P100 to P105 can be set to analog input. • For P106 and P107, use of an on-chip pull-up resistor can be specified by a software setting. • For input to P107, the threshold level can be specified.	Analog input port	ANI16
P101				ANI17
P102				ANI18
P103				ANI19
P104				ANI20
P105				ANI21
P106			Input port	(LTXD1)
P107				(LRXD1)
P120	I/O	Port 12 • Input of P125 can be set to TTL input buffer. • P120 and P125 can be set to analog input. • For P120 and P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting. • Output from P120 can be set to N-ch open-drain output. • For input to P120 and P125, the threshold level can be specified.	Analog input port	ANI23/TI07/TO07/(TO13)/TRDIOD0/SO01/(SCK10)/(LTXD1)/INTP4
P121	Input		Input port	X1
P122				X2/EXCLK
P123				XT1
P124				XT2/EXCLKS
P125	I/O		Analog input port	ANI22/TI03/TO03/TRDIOB0/SSI01/(LRXD1)/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 Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TRD1RES/PCLBUZ0
P150	I/O	Port 15 • Use of an on-chip pull-up resistor can be specified by a software setting. • For input to P150, P152, P153, and P154, the threshold level can be specified.	Input port	(SSI11)
P151				(SO11)
P152				(SI11)
P153				(SCK11)
P154				LRXD2/(SNZOUT7)
P155				LTXD2/(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 (PIORx).

2.1.2 RL78/F22 48-pin Products

(1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0 • Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	(TI05)/(TO05)/INTP9
P10	I/O	Port 1 • Input of P10, P11, P13, P14, P16, and P17 can be set to TTL input buffer. • Use of an on-chip pull-up resistor can be specified by a software setting. • Output from P10 to P17 can be set to N-ch open-drain output. • For input to P10, P11, P13, P14, P16, and P17, the threshold level can be specified.	Input port	TI13/TO13/TRJO0/SCK10/SCL10/TS11
P11				TI12/TO12/(TRDIOB0)/SI10/SDA10/RXD1
P12				TI11/TO11/(TRDIOD0)/INTP5/SO10/TXD1/SNZOUT3/TSCAP
P13				TI04/TO04/TRDIOA0/TRDCLK0/SI01/SDA01/LTXD0/TS10
P14				TI06/TO06/TRDIOC0/SCK01/SCL01/LRXD0/TS9
P15				TI05/TO05/TRDIOA1/(TRDIOA0)/(TRDCLK0)/SO00/TXD0/TOOLTXD/RTC1HZ/TS8
P16				TI02/TO02/TRDIOC1/SI00/SDA00/RXD0/TOOLRXD/TS7
P17				TI00/TO00/TRDIOB1/SCK00/SCL00/INTP3/TS6
P30	I/O	Port 3 • Input of P30 can be set to TTL input buffer. • Output from P32 can be set to N-ch open-drain output. • For input to P30 to P32, use of an on-chip pull-up resistor can be specified by a software setting. • For input to P30, the threshold level can be specified.	Input port	TI01/TO01/TRDIOD1/SSI00/INTP2/SNZOUT0
P31				STOPST/(INTP2)
P32				(SO11)/INTP7
P40	I/O	Port 4 • Use of an on-chip pull-up resistor can be specified by a software setting. • For input to P41, the threshold level can be specified.	Input port	TOOL0
P41				TI10/TO10/TRJIO0/TRD0RES/(SI10)/(RXD1)/SNZOUT2
P60	I/O	Port 6 • Input of P62 and P63 can be set to TTL input buffer. • Use of an on-chip pull-up resistor can be specified by a software setting. • Output from P60 to P63 can be set to N-ch open-drain output. • For input to P60 to P63, the threshold level can be specified.	Input port	(TO01)/(SCK00)/(SCL00)/TS0
P61				(TO02)/(SI00)/(SDA00)/(RXD0))/TS1
P62				(TO03)/(SO00)/(TXD0)/SCLA0/TS2
P63				(TO07)/(SSI00)/SDAA0/TS3
P70	I/O	Port 7 • Input of P70, P71, and P73 can be set to TTL input buffer. • P70 to P73 can be set to analog input. • Use of an on-chip pull-up resistor can be specified by a software setting. • Output from P70 to P72 can be set to N-ch open-drain output. • For input to P70, P71, and P73, the threshold level can be specified.	Analog input port	ANI24/KR0/INTP8/SI11/SDA11/SNZOUT4
P71				ANI25/KR1/INTP6/SCK11/SCL11/SNZOUT5
P72				ANI26/KR2/SO11/SNZOUT6/TS5
P73				ANI27/KR3/SSI11/SNZOUT7/TS4
P80	I/O	Port 8 • P80 to P87 can be set to analog input.	Analog input port	ANI0
P81				ANI1
P82				ANI2
P83				ANI3
P84				ANI4
P85				ANI5
P86				ANI6
P87				ANI7
P90	I/O	Port 9 • P90 to P92 can be set to analog input.	Analog input port	ANI8
P91				ANI9
P92				ANI10

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

(2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P120	I/O	Port 12 • Input of P125 can be set to TTL input buffer. • P120 and P125 can be set to analog input. • For P120 and P125, use of an on-chip pull-up resistor can be specified by a software setting. • Output from P120 can be set to N-ch open-drain output. • For input to P120 and P125, the threshold level can be specified.	Analog input port	ANI23/TI07/TO07/(TO13)/TRDIOD0/ SO01/(SCK10)/INTP4
P121	Input		Input port	X1
P122				X2/EXCLK
P123				XT1
P124				XT2/EXCLKS
P125	I/O		Analog input port	ANI22/TI03/TO03/TRDIOB0/ SSI01/INTP1/SNZOUT1
P130	Output	Port 13	Output port	RESOUT
P137	Input		Input port	INTP0
P140	I/O	Port 14 • Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TRD1RES/PCLBUZ0

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

2.1.3 Pins for Each Product (Pins Other Than Port Pins)

This subchapter shows the pins other than the ports shown in **Table 2-2** and **Table 2-3** 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/F25 Pins Other than Port Pins (1/6)

Pin Function	I/O	Function	Pin Count			
			100-pin	80-pin	64-pin	48-pin
ANI0	Input	A/D converter analog input (high-speed)	√	√	√	√
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	A/D converter analog input (normal-speed)	√	–	–	–
ANI17	Input		√	–	–	–
ANI18	Input		√	–	–	–
ANI19	Input		√	–	–	–
ANI20	Input		√	–	–	–
ANI21	Input		√	–	–	–
ANI22	Input		√	√	√	√
ANI23	Input		√	√	√	√
ANI24	Input		√	√	√	√
ANI25	Input		√	√	√	√
ANI26	Input		√	√	√	√
ANI27	Input		√	√	√	√
ANI28	Input		√	√	√	–
IVCMP00	Input	Comparator analog voltage input	√	√	√	√
IVCMP01	Input		√	√	√	√
IVCMP02	Input		√	√	√	√
IVCMP03	Input		√	√	√	√
IVREF0	Input	Comparator reference voltage input	√	√	√	√

Table 2-2. List of RL78/F25 Pins Other than Port Pins (2/6)

Pin Function	I/O	Function	Pin Count			
			100-pin	80-pin	64-pin	48-pin
TS0	I/O	Electrostatic capacitance measurement pins (touch sensor)	√	√	√	√
TS1	I/O		√	√	√	√
TS2	I/O		√	√	√	√
TS3	I/O		√	√	√	√
TS4	I/O		√	√	–	–
TS5	I/O		√	√	–	–
TS6	I/O		√	√	–	–
TS7	I/O		√	√	–	–
TS8	I/O		√	√	√	–
TS9	I/O		√	√	√	–
TS10	I/O		√	√	√	–
TS11	I/O		√	√	√	–
TS12	I/O		√	√	√	√
TS13	I/O		√	√	√	√
TS14	I/O		√	√	√	√
TS15	I/O		√	√	√	√
TS16	I/O		√	√	√	√
TS17	I/O		√	√	√	√
TS18	I/O		√	√	√	√
TS19	I/O		√	√	√	–
TS20	I/O		√	√	√	–
TS21	I/O		√	√	√	–
TS22	I/O		√	√	√	–
TS23	I/O		√	√	√	√
TS24	I/O		√	√	√	√
TS25	I/O		√	√	√	√
TS26	I/O		√	√	√	√
TS27	I/O		√	√	–	–
TS28	I/O		√	√	–	–
TS29	I/O		√	√	–	–
TS30	I/O		√	√	–	–
TSCAP	–	Pin for connecting a power supply stabilization capacitor for the touch sensor interface. Connect this pin to V_{SS} via a capacitor (10 nF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.	√	√	√	√

Table 2-2. List of RL78/F25 Pins Other than Port Pins (3/6)

Pin Function	I/O	Function	Pin Count			
			100-pin	80-pin	64-pin	48-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	√	√	√	√
TI15	Input	16-bit timer 15 input	√	√	√	√
TI16	Input	16-bit timer 16 input	√	√	√	√
TI17	Input	16-bit timer 17 input	√	√	√	√
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	√	√	√	√
TO15	Output	16-bit timer 15 output	√	√	√	√
TO16	Output	16-bit timer 16 output	√	√	√	√
TO17	Output	16-bit timer 17 output	√	√	√	√

Table 2-2. List of RL78/F25 Pins Other than Port Pins (4/6)

Pin Function	I/O	Function	Pin Count			
			100-pin	80-pin	64-pin	48-pin
TRJIO0	I/O	Timer RJ input/output	√	√	√	√
TRJO0	Output	Timer RJ output	√	√	√	√
TRDCLK0	Input	Timer RDe external clock input	√	√	√	√
TRDIOA0	I/O	Timer RDe0 input/output	√	√	√	√
TRDIOB0	I/O		√	√	√	√
TRDIOC0	I/O		√	√	√	√
TRDIOD0	I/O		√	√	√	√
TRDIOA1	I/O	Timer RDe1 input/output	√	√	√	√
TRDIOB1	I/O		√	√	√	√
TRDIOC1	I/O		√	√	√	√
TRDIOD1	I/O		√	√	√	√
TRD0RES	Input	Timer RDe0 external timer counter clear trigger input	√	√	√	√
TRD1RES	Input	Timer RDe1 external timer counter clear trigger input	√	√	√	√
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		√	√	√	√
SCK00	I/O	Clock input/output for CSI00	√	√	√	√
SCK01	I/O	Clock input/output for CSI01	√	√	√	√
SCK10	I/O	Clock input/output for CSI10	√	√	√	√
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	√	√	√	√
SSI00	Input	Slave select input to CSI00 (SPI00)	√	√	√	√
SSI01	Input	Slave select input to CSI01 (SPI01)	√	√	√	√
SSI10	Input	Slave select input to CSI10 (SPI10)	√	√	√	–
SSI11	Input	Slave select input to CSI11 (SPI11)	√	√	√	√

Table 2-2. List of RL78/F25 Pins Other than Port Pins (5/6)

Pin Function	I/O	Function	Pin Count			
			100-pin	80-pin	64-pin	48-pin
CRXD0	Input	Serial data input to CAN	√	√	√	√
CRXD1	Input		√	√	√	√
CTXD0	Output	Serial data output from CAN	√	√	√	√
CTXD1	Output		√	√	√	√
LRXD0	Input	Serial data input to LIN	√	√	√	√
LRXD1	Input		√	√	√	√
LRXD2	Input		√	√	√	√
LTXD0	Output	Serial data output from LIN	√	√	√	√
LTXD1	Output		√	√	√	√
LTXD2	Output		√	√	√	√
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		√	√	√	–
INTP13	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	√	√	√	√

Table 2-2. List of RL78/F25 Pins Other than Port Pins (6/6)

Pin Function	I/O	Function	Pin Count			
			100-pin	80-pin	64-pin	48-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	–		√	√	√	√
XT1	–	Resonator connection for subsystem clock	√	√	√	√
XT2	–		√	√	√	√
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 P121 to P124, P137, and RESET pins	√	√	√	√
AV_{DD}	–	Positive power supply for the A/D converter, D/A converter, comparator, P80 to P87, P90 to P97, and P100 to P105	√	√	√	√
EV_{DD0}	–	Positive power supply for the pins that are not connected to V_{DD} and AV_{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 P121 to P124, P137, and RESET pins	√	√	√	√
AV_{SS}	–	Ground potential for the A/D converter, D/A converter, comparator, P80 to P87, P90 to P97, and P100 to P105	√	√	√	√
EV_{SS0}	–	Ground potential for the pins that are not connected to V_{SS} and AV_{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/F22 Pins Other than Port Pins (1/6)

Pin Function	I/O	Function	Pin Count		
			48-pin	32-pin	24-pin
ANI0	Input	A/D converter analog input (high-speed)	√	√	√
ANI1	Input		√	√	√
ANI2	Input		√	√	√
ANI3	Input		√	√	–
ANI4	Input		√	√	–
ANI5	Input		√	√	–
ANI6	Input		√	–	–
ANI7	Input		√	–	–
ANI8	Input		√	–	–
ANI9	Input		√	–	–
ANI10	Input		√	–	–
ANI22	Input	A/D converter analog input (normal-speed)	√	√	–
ANI23	Input		√	√	√
ANI24	Input		√	–	–
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		√	–	–

Table 2-3. List of RL78/F22 Pins Other than Port Pins (2/6)

Pin Function	I/O	Function	Pin Count		
			48-pin	32-pin	24-pin
TS0	I/O	Electrostatic capacitance measurement pins (touch sensor)	√	√	–
TS1	I/O		√	√	–
TS2	I/O		√	√	√
TS3	I/O		√	√	√
TS4	I/O		√	–	–
TS5	I/O		√	–	–
TS6	I/O		√	√	√
TS7	I/O		√	√	√
TS8	I/O		√	√	√
TS9	I/O		√	√	√
TS10	I/O		√	√	√
TS11	I/O		√	√	–
TSCAP	–	Pin for connecting a power supply stabilization capacitor for the touch sensor interface. Connect this pin to V_{SS} via a capacitor (10 nF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.	√	√	√

Table 2-3. List of RL78/F22 Pins Other than Port Pins (3/6)

Pin Function	I/O	Function	Pin Count		
			48-pin	32-pin	24-pin
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)	√	√	√
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 RDe external clock input	√	√	√
TRDIOA0	I/O	Timer RDe0 input/output	√	√	√
TRDIOB0	I/O		√	√	√
TRDIOC0	I/O		√	√	√
TRDIOD0	I/O		√	√	√
TRDIOA1	I/O	Timer RDe1 input/output	√	√	√
TRDIOB1	I/O		√	√	√
TRDIOC1	I/O		√	√	√
TRDIOD1	I/O		√	√	√
TRD0RES	Input	Timer RDe0 external timer counter clear trigger input	√	√	–
TRD1RES	Input	Timer RDe1 external timer counter clear trigger input	√	–	–

Table 2-3. List of RL78/F22 Pins Other than Port Pins (4/6)

Pin Function	I/O	Function	Pin Count		
			48-pin	32-pin	24-pin
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		√	–	–
SCK00	I/O	Clock input/output for CSI00	√	√	√
SCK01	I/O	Clock input/output for CSI01	√	√	√
SCK10	I/O	Clock input/output for CSI10	√	√	–
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	√	–	–
SSI00	Input	Slave select input to CSI00 (SPI00)	√	√	√
SSI01	Input	Slave select input to CSI01 (SPI01)	√	√	–
SSI11	Input	Slave select input to CSI11 (SPI11)	√	–	–
LRXD0	Input	Serial data input to LIN	√	√	√
LTXD0	Output	Serial data output from LIN	√	√	√

Table 2-3. List of RL78/F22 Pins Other than Port Pins (5/6)

Pin Function	I/O	Function	Pin Count		
			48-pin	32-pin	24-pin
INTP0	Input	External interrupt input	√	√	√
INTP1	Input		√	√	–
INTP2	Input		√	√	√
INTP3	Input		√	√	√
INTP4	Input		√	√	√
INTP5	Input		√	√	√
INTP6	Input		√	–	–
INTP7	Input		√	–	–
INTP8	Input		√	–	–
INTP9	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	√	√	√

Table 2-3. List of RL78/F22 Pins Other than Port Pins (6/6)

Pin Function	I/O	Function	Pin Count		
			48-pin	32-pin	24-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	–		√	√	√
XT1	–	Resonator connection for subsystem clock	√	–	–
XT2	–		√	–	–
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 port pins (except pins connected to AV_{DD}) and RESET pins	√	√	√
AV_{DD}	–	Positive power supply for the A/D converter, P80 to P87 and P90 to P92	√	√	√
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 port pins (except pins connected to AV_{SS}) and RESET pins	√	√	√
AV_{SS}	–	Ground potential for the A/D converter, P80 to P87 and P90 to P92	√	√	√
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	√	√	√

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/F25 and the 48-pin products of RL78/F22 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. P01 to P03 are provided only for RL78/F25 products.

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, CAN serial data I/O, and capacitance measurement.

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).

P10, P11, and P13 to P17 pins can be selected for digital input/output or capacitance measurement by setting touch sensor port mode control register 1 (TSPMC1).

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, CAN serial data I/O, and capacitance measurement.

(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) SCK00, SCK01, 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 output pins of the simplified I^2C serial interface.

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

These are timer I/O pins of timer RDe.

(m) TRDCLK0

This is an external clock input pin of timer RDe.

(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 for RL78/F25.

(p) LRXD0, LRXD1

These are serial data input pins for the LIN. LRXD1 is provided only for RL78/F25.

(q) CTXD0

This is a serial data output pin for the CAN. CTXD0 is provided only for RL78/F25.

(r) CRXD0

This is a serial data input pin for the CAN. CRXD0 is provided only for RL78/F25.

(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.

(v) TS16 to TS18, TS23 to TS26 (RL78/F25), TS6 to TS11 (RL78/F22)

These are capacitance measurement pins.

(w) TSCAP

This is the capacitor connection pin of the power supply for capacitance measurement.

2.2.3 P30 to P32 (Port 3)

P30 to P34 function as an I/O port. These pins also function as external interrupt request input, serial interface slave select input, serial interface data output, 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 output from the P32 pin, CMOS output or N-ch open-drain output can be selected using the port output mode register 3 (POM3).

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 P32 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 P32 function as external interrupt request input, serial interface slave select input, serial interface data output, timer I/O, SNOOZE status output, and STOP status output.

(a) 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.

(b) SSI00

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

(c) TI01, TI14, TI16

These are pins for inputting an external count clock/capture trigger to 16-bit timers. TI14 and TI16 are provided only for RL78/F25 products.

(d) TO01, TO14, TO16

These are timer output pins of 16-bit timers. TO14 and TO16 are provided only for RL78/F25 products.

(e) SNZOUT0

This is a SNOOZE status output pin.

(f) TRDIOD1

This is a timer output pin of timer RDe.

(g) STOPST

This is a STOP status output pin.

(h) SO11

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

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, serial interface data input, timer RDe counter clear trigger input, and LIN serial data I/O. P42 to P47 are provided only for RL78/F25 products.

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

For the P41 and P43 pins, 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, serial interface data input, timer RDe counter clear trigger input, 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 for RL78/F25 products.

(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.

(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.

(j) TRD0RES

This is an external timer counter clear trigger input pin of timer RDe.

(k) RXD1

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

(l) SI10

This is a serial data input pin of the CSI10 serial interface.

Caution After reset release, the relationships between P40/TOOL0 and the operating mode are as follows.

Table 2-4. 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

For details, see 32.5 Serial Programming Method.

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, STOP status output, and capacitance measurement. P50 to P57 are provided only for RL78/F25 products.

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).

P50 to P57 pins can be selected for digital input/output or capacitance measurement by setting touch sensor port mode control register 5 (TSPMC5).

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, STOP status output, and capacitance measurement.

(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) 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.

(i) SNZOUT0, SNZOUT1

These are SNOOZE status output pins.

(j) STOPST

This is a STOP status output pin.

(k) TS19 to TS22, TS27 to TS30

These are capacitance measurement pins.

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, SNOOZE status output, CAN serial data I/O, and capacitance measurement. P64 to P67 are provided only for RL78/F25 products.

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).

P60 to P67 pins can be selected for digital input/output or capacitance measurement by setting touch sensor port mode control register 6 (TSPMC6).

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, SNOOZE status output, CAN serial data I/O, and capacitance measurement.

(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 output 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. These are provided only for RL78/F25 products.

(l) TO00, TO01, TO02, TO03, TO07, TO14, TO16

These are timer output pins of 16-bit timers. TO00, TO14, and TO16 are provided only for RL78/F25 products.

(m) SNZOUT2, SNZOUT3

These are SNOOZE status output pins.

(n) CTXD1

This is a serial data output pin of the CAN. CTXD1 is provided only for RL78/F25 products.

(o) CRXD1

This is a serial data input pin of the CAN. CRXD1 is provided only for RL78/F25 products.

(p) TS0 to TS7 (RL78/F25), TS0 to TS3 (RL78/F22)

These are capacitance measurement 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, CAN serial data I/O, LIN serial data I/O, and capacitance measurement. P74 to P77 are provided only for RL78/F25 products.

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).

P70 to P77 pins can be selected for digital input/output or capacitance measurement by setting touch sensor port mode control register 7 (TSPMC7).

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, CAN serial data I/O, LIN serial data I/O, and capacitance measurement.

(a) ANI24 to ANI28

These are analog input pins (normal-speed) of the A/D converter. For details, see **12.7.13 (iv) (v)**.

(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.

(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 output 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 for RL78/F25 products.

(m) TO15, TO17

These are timer output pins of 16-bit timers. They are provided only for RL78/F25 products.

(n) CTXD0

This is a serial data output pin for the CAN. It is provided only for RL78/F25 products.

(o) CRXD0

This is a serial data input pin for the CAN. It is provided only for RL78/F25 products.

(p) SNZOUT4 to SNZOUT7

These are SNOOZE status output pins.

(q) LTXD2

This is a serial data output pin of the LIN. LTXD2 is provided only for RL78/F25 products.

(r) LRXD2

This is a serial data input pin of the LIN. LRXD2 is provided only for RL78/F25 products.

(s) TS8 to TS15 (RL78/F25), TS4, TS5 (RL78/F22)

These are capacitance measurement 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.

Input to the P80 to P87 pins can be specified as digital input or analog input in 1-bit units, using the port mode control register 8 (PMC8).

(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) ANI0 to ANI7

These are analog input pins (high-speed) of the A/D converter. For details, see **12.7.13 (iv) (v)**.

(b) ANO0

This is a D/A converter output pin. It is provided only for RL78/F25 products.

(c) IVCMP00 to IVCMP03

These are analog voltage input pins of the comparator. They are provided only for RL78/F25 products.

(d) IVREF0

This is a reference voltage input pin of the comparator. It is provided only for RL78/F25 products.

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. P93 to P97 are provided only for RL78/F25 products.

Input to the P90 to P97 pins can be specified as digital input or analog input in 1-bit units, using the port mode control register 9 (PMC9).

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) ANI8 to ANI15

These are analog input pins (high-speed) of the A/D converter. For details, see **12.7.13 (iv) (v)**.

2.2.10 P100 to P107 (Port 10)

P100 to P107 function as an I/O port. These pins are provided only for the 100-pin products of RL78/F25. These pins also function as A/D converter analog input and LIN serial data I/O. P100 to P107 are provided only for RL78/F25 products.

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).

Input to the P100 to P105 pins can be specified as digital input or analog input in 1-bit units, using the port mode control register 10 (PMC10).

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) ANI16 to ANI21

These are analog input pins (normal-speed) of the A/D converter. For details, see **12.7.13 (iv) (v)**.

(b) LTXD1

This is a serial data output pin for the LIN. LTXD1 is provided only for RL78/F25 products.

(c) LRXD1

This is a serial data input pin for the LIN. LRXD1 is provided only for RL78/F25 products.

2.2.11 P120 to P127 (Port 12)

P120 and P125 to P127 function as an I/O port, and P121 to P124 function as an input port. P126 and P127 are provided only for RL78/F25 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, serial clock I/O, timer I/O, LIN serial data 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 P120 and P125 pins, 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, serial clock I/O, timer I/O, LIN serial data I/O, and SNOOZE status output.

(a) ANI22, ANI23

These are analog input pins (normal-speed) of the A/D converter. For details, see **12.7.13 (iv) (v)**.

(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) 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 RDe.

(m) SCK10

This is a serial clock I/O pin of the CSI10 serial interface.

(n) LTXD1

This is a serial data output pin for the LIN. LTXD1 is provided only for RL78/F25 products.

(o) LRXD1

This is a serial data input pin for the LIN. LRXD1 is provided only for RL78/F25 products.

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, and timer RDe counter clear trigger input.

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, and timer RDe counter clear trigger input.

(a) PCLBUZ0

This is a clock/buzzer output pin.

(b) TRD1RES

This is an external timer counter clear trigger input pin of timer RDe.

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, SNOOZE status output, and LIN serial data I/O. P150 to P157 are provided only for RL78/F25 products.

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

For the P150, P152, P153, and P154 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, SNOOZE status output, and LIN serial data I/O.

(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.

(f) LTXD2

This is a serial data output pin of the LIN. LTXD2 is provided only for RL78/F25 products.

(g) LRXD2

This is a serial data input pin of the LIN. LRXD2 is provided only for RL78/F25 products.

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

(1) V_{DD}, AV_{DD}, EV_{DD0}, EV_{DD1}

V_{DD} is a positive power supply pin for the P121 to P124, P137 and the pins other than ports.

AV_{DD} is a positive power supply pin for the P80 to P87, P90 to P97, and P100 to P105.

EV_{DD0} and EV_{DD1} are positive power supply pins for ports other than P80 to P87, P90 to P97, P100 to P105, P121 to P124, and P137. EV_{DD1} is provided only for the 100-pin products of RL78/F25.

Caution For products with 48 pins or less, V_{DD} is the positive power supply pin instead of EV_{DD0} and EV_{DD1}.

(2) V_{SS}, AV_{SS}, EV_{SS0}, EV_{SS1}

V_{SS} is a ground potential pin for P121 to P124, P137 and the pins other than ports.

AV_{SS} is a ground potential pin for the P80 to P87, P90 to P97, and P100 to P105.

EV_{SS0} and EV_{SS1} are ground potential pins for ports other than P80 to P87, P90 to P97, P100 to P105, P121 to P124, and P137. EV_{SS1} is provided only for the 100-pin products of RL78/F25.

Caution For products with 48 pins or less, V_{SS} is the ground potential pin instead of EV_{SS0} and EV_{SS1}.

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}, AV_{DD} to AV_{SS}, EV_{DD0} to EV_{SS0} and EV_{DD1} to EV_{SS1} lines.

2.2.16 RESET

This is an active-low system reset input pin.

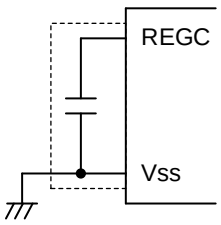
When the external reset pin is not used, connect this pin directly or via a resistor to V_{DD}.

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

Table 2-5 and **Table 2-6** show the recommended connections of unused pins taking the 100-pin products of RL78/F25 and the 48-pin products of RL78/F22 as examples.

Table 2-5. Connection of Unused Pins (100-Pin Products of RL78/F25) (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. • Output: Leave open.
P01/(TI04)/(TO04)		
P02/(TI06)/(TO06)		
P03/(RTC1HZ)		
P10/TI13/TO13/TRJO0/SCK10/SCL10/ LTXD1/CTXD0/TS26		
P11/TI12/TO12/(TRDIOB0)/SI10/SDA10/ RXD1/LRXD1/CRXD0/TS25		
P12/TI11/TO11/(TRDIOD0)/INTP5/SO10/ TXD1/SNZOUT3/TSCAP		
P13/TI04/TO04/TRDIOA0/TRDCLK0/SI01/ SDA01/LTXD0/TS24		
P14/TI06/TO06/TRDIOC0/SCK01/SCL01/ LRXD0/TS23		
P15/TI05/TO05/TRDIOA1/(TRDIOA0)/ (TRDCLK0)/SO00/TXD0/TOOLTXD/ RTC1HZ/TS18		
P16/TI02/TO02/TRDIOC1/SI00/SDA00/ RXD0/TOOLRXD/TS17		
P17/TI00/TO00/TRDIOB1/SCK00/SCL00/ INTP3/TS16		
P30/TI01/TO01/TRDIOD1/SSI00/INTP2/ SNZOUT0		
P31/TI14/TO14/STOPST/(INTP2)		
P32/TI16/TO16/(SO11)/INTP7		
P40/TOOL0		• Input: Independently connect to EV_{DD0} and EV_{DD1} via a resistor or leave open. (Note: For leave open, the condition PU40 = 1.) • Output: Leave open.
P41/TI10/TO10/TRJIO0/TRD0RES/(SI10)/ (RXD1)/VCOUT0/SNZOUT2		• Input: Independently connect to EV_{DD0} and EV_{DD1}, or EV_{SS0} and EV_{SS1} via a resistor. • Output: Leave open.
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 (PIORx).

Table 2-5. Connection of Unused Pins (100-Pin Products of RL78/F25) (2/3)

Pin Name	I/O	Recommended Connection of Unused Pins
P50/(SSI01)/(INTP3)/TS19	I/O	• Input: Independently connect to EV_{DD0} and EV_{DD1}, or EV_{SS0} and EV_{SS1} via a resistor. • Output: Leave open.
P51/(SO01)/INTP11/TS20		
P52/(SCK01)/(STOPST)/TS21		
P53/(SI01)/INTP10/TS22		
P54/(TI11)/(TO11)/SSI10/TS27		
P55/(TI13)/(TO13)/TS28		
P56/(TI15)/(TO15)/(SNZOUT1)/TS29		
P57/(TI17)/(TO17)/(SNZOUT0)/TS30		
P60/(TO01)/(SCK00)/(SCL00)/CRXD1/TS0		
P61/(TO02)/(SI00)/(SDA00)/(RXD0)/CTXD1/TS1		
P62/(TO03)/(SO00)/(TXD0)/SCLA0/TS2		
P63/(TO07)/(SSI00)/SDAA0/TS3		
P64/(TI14)/(TO14)/(SNZOUT3)/TS4		
P65/(TI16)/(TO16)/(SNZOUT2)/TS5		
P66/(TI00)/(TO00)/TS6		
P67/(TI02)/(TO02)/TS7		
P70/ANI24/KR0/TI15/TO15/INTP8/SI11/ SDA11/(LTXD2)/SNZOUT4/TS15		
P71/ANI25/KR1/TI17/TO17/INTP6/SCK11/ SCL11/(LRXD2)/SNZOUT5/TS14		
P72/ANI26/KR2/(CTXD0)/SO11/SNZOUT6/TS13		
P73/ANI27/KR3/(CRXD0)/SSI11/SNZOUT7/TS12		
P74/ANI28/KR4/(SO10)/(TXD1)/TS11		
P75/KR5/(SI10)/(RXD1)/TS10		
P76/KR6/(SCK10)/TS9		
P77/KR7/(SSI10)/INTP12/TS8		
P80/ANI0/ANO0		• Input: Independently connect to AV_{DD} or AV_{SS} via a resistor. • Output: Leave open.
P81/ANI1		
P82/ANI2/IVCMP00		
P83/ANI3/IVCMP01		
P84/ANI4/IVCMP02		
P85/ANI5/IVCMP03/IVREF0		
P86/ANI6		
P87/ANI7		
P90/ANI8		
P91/ANI9		
P92/ANI10		
P93/ANI11		
P94/ANI12		
P95/ANI13		
P96/ANI14		
P97/ANI15		

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

Table 2-5. Connection of Unused Pins (100-Pin Products of RL78/F25) (3/3)

Pin Name	I/O	Recommended Connection of Unused Pins
P100/ANI16	I/O	• Input: Independently connect to AV_{DD} or AV_{SS} via a resistor. • Output: Leave open.
P101/ANI17		
P102/ANI18		
P103/ANI19		
P104/ANI20		
P105/ANI21		
P106/(LTXD1)		• Input: Independently connect to EV_{DD0} and EV_{DD1}, or EV_{SS0} and EV_{SS1} via a resistor. • Output: Leave open.
P107/(LRXD1)		
P120/ANI23/TI07/TO07/(TO13)/TRDIOD0/ SO01/(SCK10)/(LTXD1)/INTP4		
P121/X1	Input	• Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK		
P123/XT1		
P124/XT2/EXCLKS		
P125/ANI22/TI03/TO03/TRDIOB0/ SSI01/(LRXD1)/INTP1/SNZOUT1	I/O	• Input: Independently connect to EV_{DD0} and EV_{DD1}, or EV_{SS0} and EV_{SS1} via a resistor. • 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/TRD1RES/PCLBUZ0	I/O	• Input: Independently connect to EV_{DD0} and EV_{DD1}, or EV_{SS0} and EV_{SS1} via a resistor. • Output: Leave open.
P150/(SSI11)		
P151/(SO11)		
P152/(SI11)		
P153/(SCK11)		
P154/LRXD2/(SNZOUT7)		
P155/LTXD2/(SNZOUT6)		
P156/(SNZOUT5)		
P157/(SNZOUT4)		
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 (PIORx).

Table 2-6. Connection of Unused Pins (48-Pin Products of RL78/F22) (1/3)

Pin Name	I/O	Recommended Connection of Unused Pins
P00/(TI05)/(TO05)/INTP9	I/O	• Input: Independently connect to V_{DD} or V_{SS} via a resistor. • Output: Leave open.
P10/TI13/TO13/TRJO0/SCK10/SCL10/ CTXD0/TS11		
P11/TI12/TO12/(TRDIOB0)/SI10/SDA10/ RXD1/CRXD0		
P12/TI11/TO11/(TRDIOD0)/INTP5/SO10/ TXD1/SNZOUT3/TSCAP		
P13/TI04/TO04/TRDIOA0/TRDCLK0/SI01/ SDA01/LTXD0/TS10		
P14/TI06/TO06/TRDIOC0/SCK01/SCL01/ LRXD0/TS9		
P15/TI05/TO05/TRDIOA1/(TRDIOA0)/ (TRDCLK0)/SO00/TXD0/TOOLTXD/ RTC1HZ/TS8		
P16/TI02/TO02/TRDIOC1/SI00/SDA00/ RXD0/TOOLRXD/TS7		
P17/TI00/TO00/TRDIOB1/SCK00/SCL00/INTP3/ TS6		
P30/TI01/TO01/TRDIOD1/SSI00/INTP2/ SNZOUT0		
P31/STOPST/(INTP2)		
P32/(SO11)/INTP7		
P40/TOOL0		• Input: Independently connect to V_{DD} via a resistor or leave open. (Note: For leave open, the condition PU40 = 1.) • Output: Leave open.
P41/TI10/TO10/TRJIO0/TRD0RES/(SI10)/ (RXD1)/SNZOUT2		• Input: Independently connect to V_{DD} or V_{SS} via a resistor. • Output: Leave open.

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

Table 2-6. Connection of Unused Pins (48-Pin Products of RL78/F22) (2/3)

Pin Name	I/O	Recommended Connection of Unused Pins
P60/(TO01)/(SCK00)/(SCL00)/TS0	I/O	• Input: Independently connect to V_{DD} or V_{SS} via a resistor. • Output: Leave open.
P61/(TO02)/(SI00)/(SDA00)/(RXD0)/TS1		
P62/(TO03)/(SO00)/(TXD0)/SCLA0/TS2		
P63/(TO07)/(SSI00)/SDAA0/TS3		
P70/ANI24/KR0/INTP8/SI11/SDA11/SNZOUT4		
P71/ANI25/KR1/INTP6/SCK11/SCL11/SNZOUT5		
P72/ANI26/KR2/(CTXD0)/SO11/SNZOUT6/TS5		
P73/ANI27/KR3/(CRXD0)/SSI11/SNZOUT7/TS4		
P80/ANI0		• Input: Independently connect to AV_{DD} or AV_{SS} via a resistor. • Output: Leave open.
P81/ANI1		
P82/ANI2		
P83/ANI3		
P84/ANI4		
P85/ANI5		
P86/ANI6		
P87/ANI7		
P90/ANI8		
P91/ANI9		
P92/ANI10		
P120/ANI23/TI07/TO07/(TO13)/TRDIOD0/ SO01/(SCK10)/INTP4		• Input: Independently connect to V_{DD} or V_{SS} via a resistor. • Output: Leave open.

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

Table 2-6. Connection of Unused Pins (48-Pin Products of RL78/F22) (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/ANI22/TI03/TO03/TRDIOB0/SSI01/ INTP1/SNZOUT1	I/O	• Input: Independently connect to V_{DD} or V_{SS} via a resistor. • Output: Leave open.
P130/RESOUT	Output	• Leave open.
P137/INTP0	Input	• Independently connect to V_{DD} or V_{SS} via a resistor.
P140/TRD1RES/PCLBUZ0	I/O	• Input: Independently connect to V_{DD} or V_{SS} via a resistor. • Output: Leave open.
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 (PIORx).

CHAPTER 3 CPU ARCHITECTURE

The RL78/F22, F25 is a microcontroller that has the RL78-S3 CPU core.

The CPU core in the RL78-S3 employs the Harvard architecture which has independent instruction fetch bus, address bus and data bus. In addition, through the adoption of three-stage pipeline control of fetch, decode, and memory access, the operation efficiency is remarkably improved over the conventional CPU core. The CPU core features high performance and highly functional instruction processing, and can be suited for use in various applications that require high speed and highly functional processing.

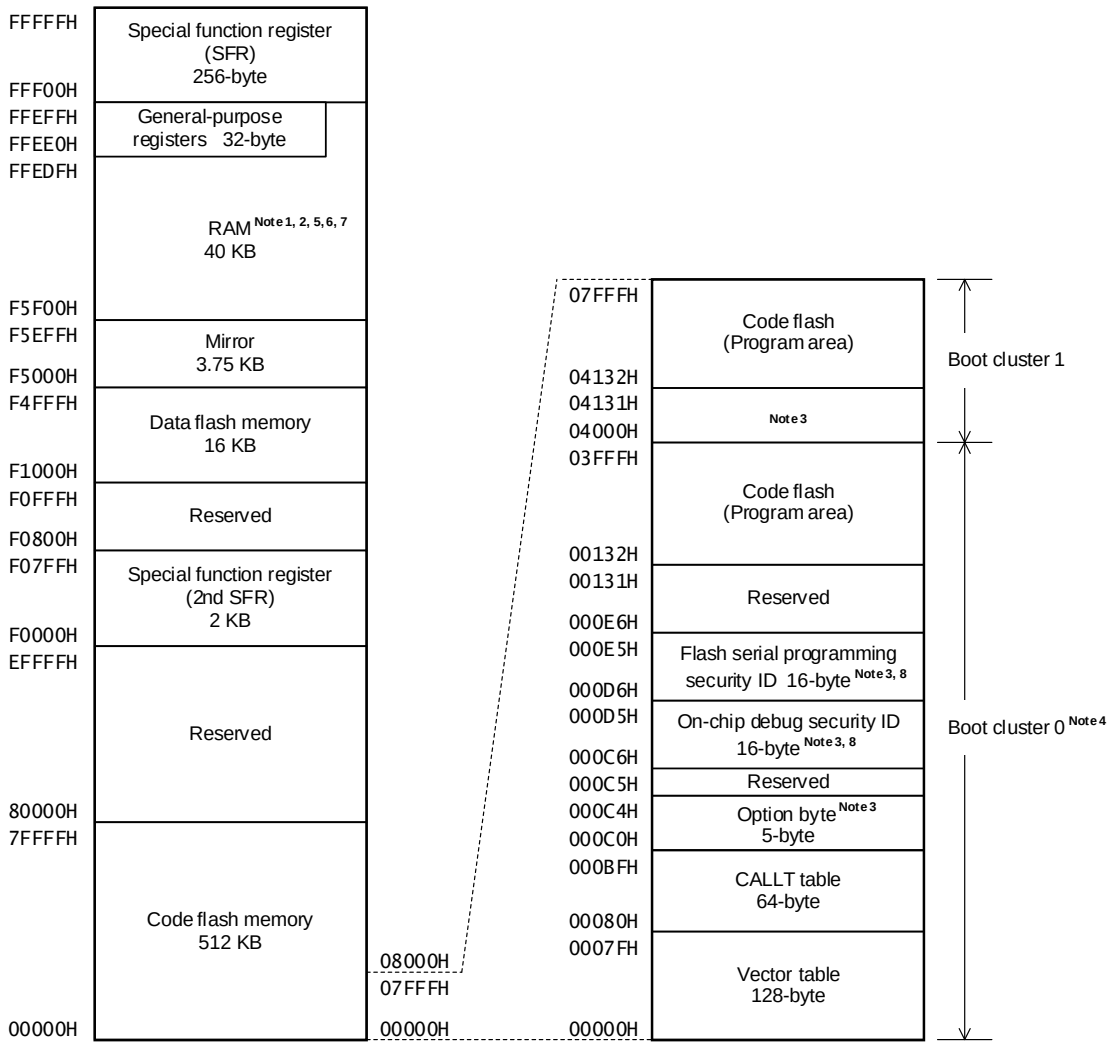
- 3-stage pipeline CISC architecture
- Address space: 1 Mbyte
- Minimum instruction execution time: One instruction per clock cycle
- General-purpose registers: Eight 8-bit registers
- Type of instruction: 81
- Multiply/divide and multiply/accumulate instructions: Supported
- Data allocation: Little endian

The RL78/F22, F25 supports an on-chip debug trace function.

3.1 Memory Space

Products in the RL78/F22 and RL78/F25 can access a 1 MB memory space. **Figure 3-1** and **Figure 3-2** show the memory maps.

Figure 3-1. Memory Map (RL78/F25)



Notes 1. In RAMSAR register the user selects the RAM start address. Use RAM within the range of addresses specified in RAMSAR register.

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 000C4H, the on-chip debug security IDs to 000C6H to 000D5H, and the flash serial programming security IDs to 000D6H to 000E5H.
When boot swap is used: Set the option bytes to 000C0H to 000C4H and 040C0H to 040C4H, the on-chip debug security IDs to 000C6H to 000D5H and 040C6H to 040D5H, and the flash serial programming security IDs to 000D6H to 000E5H and 040D6H to 040E5H.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **33.8 Security Settings**).

5. When using the on-chip trace function of the on-chip debugger, do not use the area from F6300H to F64FFH.

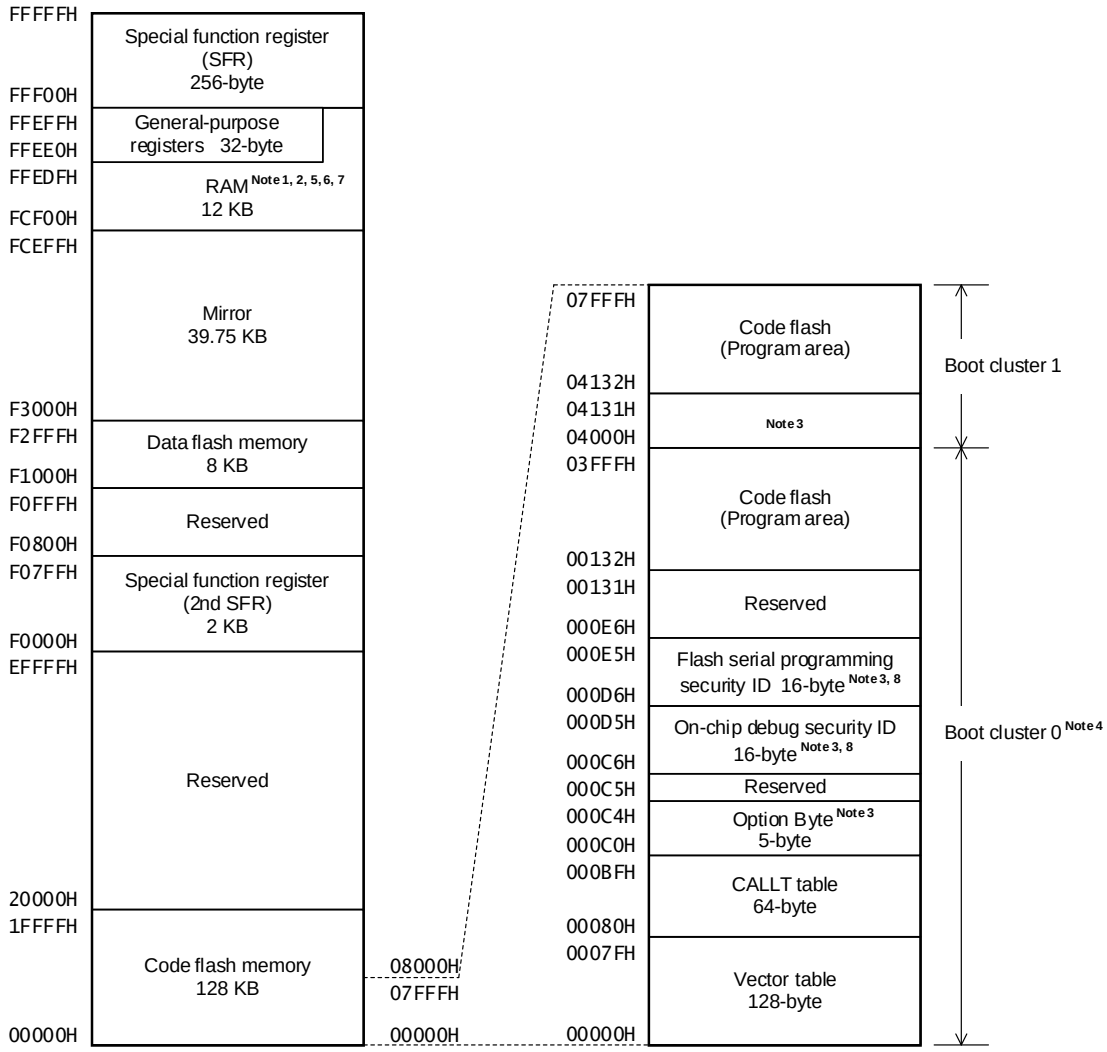
6. When performing self-programming of the on-chip debugger, do not use the area from F5F00H to F5F7FH.

7. When using the hot plug-in function, the real-time RAM monitor (RRM) or dynamic memory modification (DMM) function, do not use the area from F6500H to F652FH.

8. When IDRDEN bit of security option byte is 0, this area is read as 0.

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 (RL78/F22)



Notes 1. In RAMSAR register the user selects the RAM start address. Use RAM within the range of addresses specified in RAMSAR register.

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 000C4H, the on-chip debug security IDs to 000C6H to 000D5H, and the flash serial programming security IDs to 000D6H to 000E5H.
When boot swap is used: Set the option bytes to 000C0H to 000C4H and 040C0H to 040C4H, the on-chip debug security IDs to 000C6H to 000D5H and 040C6H to 040D5H, and the flash serial programming security IDs to 000D6H to 000E5H and 040D6H to 040E5H.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **33.8 Security Settings**).

5. When using the on-chip trace function of the on-chip debugger, do not use the area from FD300H to FD4FFH.

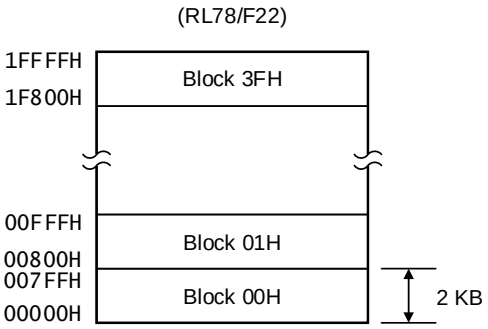
6. When performing self-programming of the on-chip debugger, do not use the area from FCF00H to FCF7FH.

7. When using the hot plug-in function, the real-time RAM monitor (RRM) or dynamic memory modification (DMM) function, do not use the area from FD500H to FD52FH.

8. When IDRDEN bit of security option byte is 0, this area is read as 0.

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 = 2 KB). Correspondence between address values and block numbers, see the **Table 3-1** starting on the next page.



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 007FFH	00H	10000H to 107FFH	20H	20000H to 207FFH	40H	30000H to 307FFH	60H
00800H to 00FFFH	01H	10800H to 10FFFH	21H	20800H to 20FFFH	41H	30800H to 30FFFH	61H
01000H to 017FFH	02H	11000H to 117FFH	22H	21000H to 217FFH	42H	31000H to 317FFH	62H
01800H to 01FFFH	03H	11800H to 11FFFH	23H	21800H to 21FFFH	43H	31800H to 31FFFH	63H
02000H to 027FFH	04H	12000H to 127FFH	24H	22000H to 227FFH	44H	32000H to 327FFH	64H
02800H to 02FFFH	05H	12800H to 12FFFH	25H	22800H to 22FFFH	45H	32800H to 32FFFH	65H
03000H to 037FFH	06H	13000H to 137FFH	26H	23000H to 237FFH	46H	33000H to 337FFH	66H
03800H to 03FFFH	07H	13800H to 13FFFH	27H	23800H to 23FFFH	47H	33800H to 33FFFH	67H
04000H to 047FFH	08H	14000H to 147FFH	28H	24000H to 247FFH	48H	34000H to 347FFH	68H
04800H to 04FFFH	09H	14800H to 14FFFH	29H	24800H to 24FFFH	49H	34800H to 34FFFH	69H
05000H to 057FFH	0AH	15000H to 157FFH	2AH	25000H to 257FFH	4AH	35000H to 357FFH	6AH
05800H to 05FFFH	0BH	15800H to 15FFFH	2BH	25800H to 25FFFH	4BH	35800H to 35FFFH	6BH
06000H to 067FFH	0CH	16000H to 167FFH	2CH	26000H to 267FFH	4CH	36000H to 367FFH	6CH
06800H to 06FFFH	0DH	16800H to 16FFFH	2DH	26800H to 26FFFH	4DH	36800H to 36FFFH	6DH
07000H to 077FFH	0EH	17000H to 177FFH	2EH	27000H to 277FFH	4EH	37000H to 377FFH	6EH
07800H to 07FFFH	0FH	17800H to 17FFFH	2FH	27800H to 27FFFH	4FH	37800H to 37FFFH	6FH
08000H to 087FFH	10H	18000H to 187FFH	30H	28000H to 287FFH	50H	38000H to 387FFH	70H
08800H to 08FFFH	11H	18800H to 18FFFH	31H	28800H to 28FFFH	51H	38800H to 38FFFH	71H
09000H to 097FFH	12H	19000H to 197FFH	32H	29000H to 297FFH	52H	39000H to 397FFH	72H
09800H to 09FFFH	13H	19800H to 19FFFH	33H	29800H to 29FFFH	53H	39800H to 39FFFH	73H
0A000H to 0A7FFH	14H	1A000H to 1A7FFH	34H	2A000H to 2A7FFH	54H	3A000H to 3A7FFH	74H
0A800H to 0AFFFH	15H	1A800H to 1AFFFH	35H	2A800H to 2AFFFH	55H	3A800H to 3AFFFH	75H
0B000H to 0B7FFH	16H	1B000H to 1B7FFH	36H	2B000H to 2B7FFH	56H	3B000H to 3B7FFH	76H
0B800H to 0BFFFH	17H	1B800H to 1BFFFH	37H	2B800H to 2BFFFH	57H	3B800H to 3BFFFH	77H
0C000H to 0C7FFH	18H	1C000H to 1C7FFH	38H	2C000H to 2C7FFH	58H	3C000H to 3C7FFH	78H
0C800H to 0CFFFH	19H	1C800H to 1CFFFH	39H	2C800H to 2CFFFH	59H	3C800H to 3CFFFH	79H
0D000H to 0D7FFH	1AH	1D000H to 1D7FFH	3AH	2D000H to 2D7FFH	5AH	3D000H to 3D7FFH	7AH
0D800H to 0DFFFH	1BH	1D800H to 1DFFFH	3BH	2D800H to 2DFFFH	5BH	3D800H to 3DFFFH	7BH
0E000H to 0E7FFH	1CH	1E000H to 1E7FFH	3CH	2E000H to 2E7FFH	5CH	3E000H to 3E7FFH	7CH
0E800H to 0EFFFH	1DH	1E800H to 1EFFFH	3DH	2E800H to 2EFFFH	5DH	3E800H to 3EFFFH	7DH
0F000H to 0F7FFH	1EH	1F000H to 1F7FFH	3EH	2F000H to 2F7FFH	5EH	3F000H to 3F7FFH	7EH
0F800H to 0FFFFH	1FH	1F800H to 1FFFFH	3FH	2F800H to 2FFFFH	5FH	3F800H to 3FFFFH	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
40000H to 407FFH	80H	50000H to 507FFH	A0H	60000H to 607FFH	C0H	70000H to 707FFH	E0H
40800H to 40FFFH	81H	50800H to 50FFFH	A1H	60800H to 60FFFH	C1H	70800H to 70FFFH	E1H
41000H to 417FFH	82H	51000H to 517FFH	A2H	61000H to 617FFH	C2H	71000H to 717FFH	E2H
41800H to 41FFFH	83H	51800H to 51FFFH	A3H	61800H to 61FFFH	C3H	71800H to 71FFFH	E3H
42000H to 427FFH	84H	52000H to 527FFH	A4H	62000H to 627FFH	C4H	72000H to 727FFH	E4H
42800H to 42FFFH	85H	52800H to 52FFFH	A5H	62800H to 62FFFH	C5H	72800H to 72FFFH	E5H
43000H to 437FFH	86H	53000H to 537FFH	A6H	63000H to 637FFH	C6H	73000H to 737FFH	E6H
43800H to 43FFFH	87H	53800H to 53FFFH	A7H	63800H to 63FFFH	C7H	73800H to 73FFFH	E7H
44000H to 447FFH	88H	54000H to 547FFH	A8H	64000H to 647FFH	C8H	74000H to 747FFH	E8H
44800H to 44FFFH	89H	54800H to 54FFFH	A9H	64800H to 64FFFH	C9H	74800H to 74FFFH	E9H
45000H to 457FFH	8AH	55000H to 557FFH	AAH	65000H to 657FFH	CAH	75000H to 757FFH	EAH
45800H to 45FFFH	8BH	55800H to 55FFFH	ABH	65800H to 65FFFH	CBH	75800H to 75FFFH	EBH
46000H to 467FFH	8CH	56000H to 567FFH	ACH	66000H to 667FFH	CCH	76000H to 767FFH	ECH
46800H to 46FFFH	8DH	56800H to 56FFFH	ADH	66800H to 66FFFH	CDH	76800H to 76FFFH	EDH
47000H to 477FFH	8EH	57000H to 577FFH	AEH	67000H to 677FFH	CEH	77000H to 777FFH	EEH
47800H to 47FFFH	8FH	57800H to 57FFFH	AFH	67800H to 67FFFH	CFH	77800H to 77FFFH	EFH
48000H to 487FFH	90H	58000H to 587FFH	B0H	68000H to 687FFH	D0H	78000H to 787FFH	F0H
48800H to 48FFFH	91H	58800H to 58FFFH	B1H	68800H to 68FFFH	D1H	78800H to 78FFFH	F1H
49000H to 497FFH	92H	59000H to 597FFH	B2H	69000H to 697FFH	D2H	79000H to 797FFH	F2H
49800H to 49FFFH	93H	59800H to 59FFFH	B3H	69800H to 69FFFH	D3H	79800H to 79FFFH	F3H
4A000H to 4A7FFH	94H	5A000H to 5A7FFH	B4H	6A000H to 6A7FFH	D4H	7A000H to 7A7FFH	F4H
4A800H to 4AFFFH	95H	5A800H to 5AFFFH	B5H	6A800H to 6AFFFH	D5H	7A800H to 7AFFFH	F5H
4B000H to 4B7FFH	96H	5B000H to 5B7FFH	B6H	6B000H to 6B7FFH	D6H	7B000H to 7B7FFH	F6H
4B800H to 4BFFFH	97H	5B800H to 5BFFFH	B7H	6B800H to 6BFFFH	D7H	7B800H to 7BFFFH	F7H
4C000H to 4C7FFH	98H	5C000H to 5C7FFH	B8H	6C000H to 6C7FFH	D8H	7C000H to 7C7FFH	F8H
4C800H to 4CFFFH	99H	5C800H to 5CFFFH	B9H	6C800H to 6CFFFH	D9H	7C800H to 7CFFFH	F9H
4D000H to 4D7FFH	9AH	5D000H to 5D7FFH	BAH	6D000H to 6D7FFH	DAH	7D000H to 7D7FFH	FAH
4D800H to 4DFFFH	9BH	5D800H to 5DFFFH	BBH	6D800H to 6DFFFH	DBH	7D800H to 7DFFFH	FBH
4E000H to 4E7FFH	9CH	5E000H to 5E7FFH	BCH	6E000H to 6E7FFH	DCH	7E000H to 7E7FFH	FCH
4E800H to 4EFFFH	9DH	5E800H to 5EFFFH	BDH	6E800H to 6EFFFH	DDH	7E800H to 7EFFFH	FDH
4F000H to 4F7FFH	9EH	5F000H to 5F7FFH	BEH	6F000H to 6F7FFH	DEH	7F000H to 7F7FFH	FEH
4F800H to 4FFFFH	9FH	5F800H to 5FFFFH	BFH	6F800H to 6FFFFH	DFH	7F800H to 7FFFFH	FFH

3.1.1 Internal Program Memory Space

The internal program memory space stores the program and table data.

The RL78/F22 and RL78/F25 products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Product	Internal ROM	
	Structure	Capacity (address)
RL78/F22	Flash memory	128 Kbytes (00000H to 1FFFFH)
RL78/F25		512 Kbytes (00000H to 7FFFFH)

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 04000H to 0407FH.

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 (1/3)

Vector Table Address	Interrupt Source	RL78/F25				RL78/F22		
		100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-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	√	√	√	√	√	√	√

Table 3-3. Vector Table (2/3)

Vector Table Address	Interrupt Source	RL78/F25				RL78/F22		
		100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
0020H	INTRAM	√	√	√	√	√	√	√
0022H	INTCTSUWR	√	√	√	√	√	√	√
0024H	INTCTSURD	√	√	√	√	√	√	√
0026H	INTCTSUFN	√	√	√	√	√	√	√
0028H	INTIICA0	√	√	√	√	√	√	√
002AH	INTP8	√	√	√	√	√	–	–
	INTRTC	√	√	√	√	√	√	√
002CH	INTTM00	√	√	√	√	√	√	√
002EH	INTTM01	√	√	√	√	√	√	√
	INTLIN2TRM	√	√	√	√	–	–	–
0030H	INTTM02	√	√	√	√	√	√	√
	INTLIN2RVC	√	√	√	√	–	–	–
0032H	INTTM03	√	√	√	√	√	√	√
	INTLIN2STA/INTLIN2	√	√	√	√	–	–	–
0034H	INTAD	√	√	√	√	√	√	√
0036H	INTP6	√	√	√	√	√	–	–
	INTLIN2WUP	√	√	√	√	–	–	–
	INTTM11H	√	√	√	√	√	√	√
0038H	INTP7	√	√	√	√	√	–	–
	INTTM13H	√	√	√	√	√	√	√
003AH	INTP9	√	√	√	√	√	–	–
	INTTM01H	√	√	√	√	√	√	√
003CH	INTP10	√	√	√	–	–	–	–
	INTTM03H	√	√	√	√	√	√	√
003EH	INTST1	√	√	√	√	√	√	√
	INTCSI10/INTIIC10	√	√	√	√	√	√	–
0040H	INTSR1	√	√	√	√	√	√	√
	INTCSI11/INTIIC11	√	√	√	√	√	–	–
0042H	INTTM04	√	√	√	√	√	√	√
0044H	INTTM05	√	√	√	√	√	√	√
	INTLIN0TRM	√	√	√	√	√	√	√
0046H	INTTM06	√	√	√	√	√	√	√
	INTLIN0RVC	√	√	√	√	√	√	√
0048H	INTTM07	√	√	√	√	√	√	√
	INTLIN0STA//INTLIN0	√	√	√	√	√	√	√
004AH	INTP11	√	√	√	–	–	–	–
	INTLIN0WUP	√	√	√	√	√	√	√
004CH	INTTM10	√	√	√	√	√	√	√
004EH	INTTM11	√	√	√	√	√	√	√
	INTLIN1TRM	√	√	√	√	–	–	–

Table 3-3. Vector Table (3/3)

Vector Table Address	Interrupt Source	RL78/F25				RL78/F22		
		100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
0050H	INTTM12	√	√	√	√	√	√	√
	INTLIN1RVC	√	√	√	√	–	–	–
0052H	INTTM13	√	√	√	√	√	√	√
	INTLIN1STA/INTLIN1	√	√	√	√	–	–	–
0054H	(Reserved)	√	√	√	√	√	√	√
0056H	(Reserved)	√	√	√	√	√	√	√
0058H	INTADGB	√	√	√	√	√	√	√
005AH	INTROM	√	√	√	√	√	√	√
005CH	INTKR	√	√	√	√	√	√	√
	INTRCAN0GRVC	√	√	√	√	–	–	–
005EH	INTRCAN0ERR	√	√	√	√	–	–	–
0060H	INTRCAN0WUP	√	√	√	√	–	–	–
0062H	(Reserved)	–	–	–	–	–	–	–
0064H	INTRCAN0CFR	√	√	√	√	–	–	–
0066H	INTRCAN0TRM	√	√	√	√	–	–	–
0068H	INTRCAN0GRFR	√	√	√	√	–	–	–
006AH	INTRCAN0GERR	√	√	√	√	–	–	–
006CH	INTP12	√	√	√	–	–	–	–
	INTLIN1WUP	√	√	√	√	–	–	–
006EH	INTRCAN1GRFR	√	√	√	√	–	–	–
0070H	INTRCAN1GERR	√	√	√	√	–	–	–
0072H	INTRCAN1WUP	√	√	√	√	–	–	–
0074H	INTTM14	√	√	√	√	–	–	–
	INTRCAN1GRVC	√	√	√	√	–	–	–
0076H	INTTM15	√	√	√	√	–	–	–
	INTRCAN1ERR	√	√	√	√	–	–	–
0078H	INTTM16	√	√	√	√	–	–	–
	INTRCAN1CFR	√	√	√	√	–	–	–
007AH	INTTM17	√	√	√	√	–	–	–
	INTRCAN1TRM	√	√	√	√	–	–	–
007CH	INTCRAM	√	√	√	√	–	–	–
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 04080H to 040BFH.

(3) Option byte area

A 5-byte area of 000C0H to 000C4H can be used as an option byte area. Set the option byte at 040C0H to 040C4H when the boot swap is used. For details, see **CHAPTER 32 OPTION BYTE**.

(4) On-chip debug security ID setting area

A 16-byte area of 000C6H to 000D5H and 040C6H to 040D5H can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 16 bytes at 000C6H to 000D5H when the boot swap is not used and at 000C6H to 000D5H and at 040C6H to 040D5H when the boot swap is used. For details, see **CHAPTER 34 ON-CHIP DEBUG FUNCTION**.

(5) Flash serial programming security ID setting area

A 16-byte area of 000D6H to 000E5H and 040D6H to 040E5H can be used as a flash serial programming security ID setting area. Set the flash serial programming security ID of 16 bytes at 000D6H to 000E5H when the boot swap is not used and at 000D6H to 000E5H and at 040D6H to 040E5H when the boot swap is used. For details, see **CHAPTER 33 FLASH MEMORY**.

3.1.2 Mirror Area

The RL78/F22 and RL78/F25 mirror the code flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, to F0000H to FFFFFH. Which area is mirrored is set by the processor mode control register (PMC).

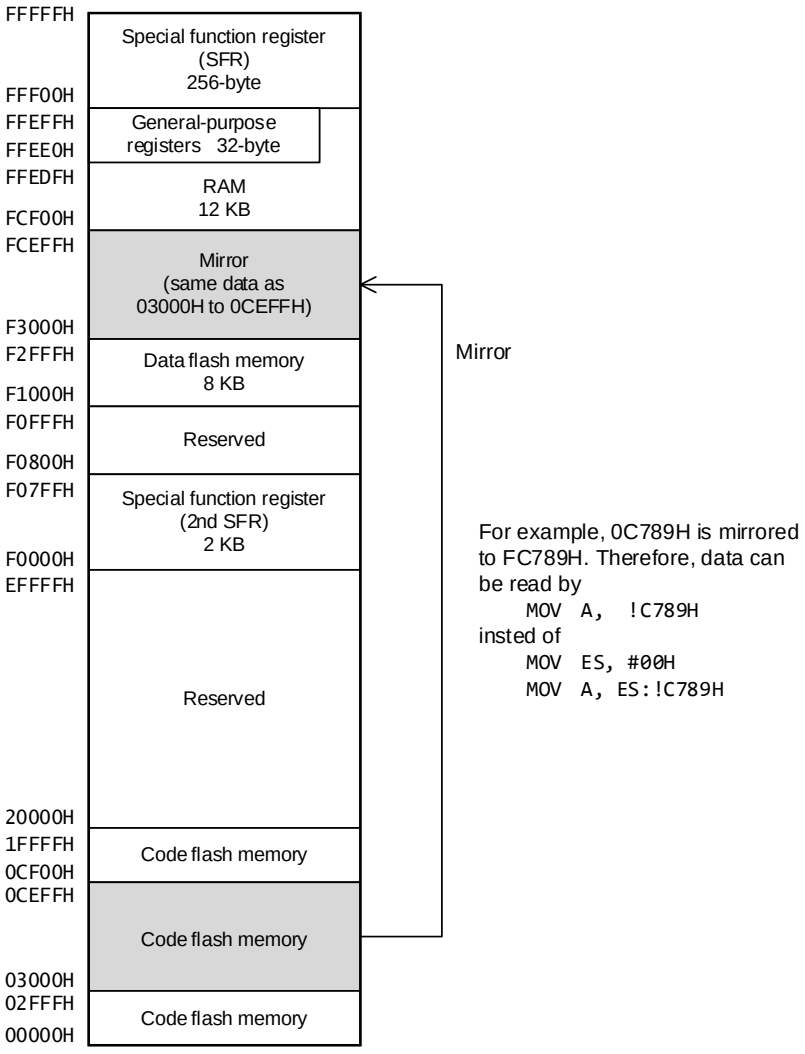
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the SFR, extended SFR, RAM, data flash memory, and use prohibited 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 for Mirror Area in RL78/F22 (Flash memory: 128 KB, RAM: 12 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 3-3. 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

Cautions 1. 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.

2. 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/F22 and RL78/F25 products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Product	Internal RAM
RL78/F22	12 Kbytes (FCF00H to FFEFFH)
RL78/F25	40 Kbytes (F5F00H 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. In RAMSAR register the user selects the RAM start address. Use RAM within the range of addresses specified in RAMSAR register.

3. When using the on-chip trace function of the on-chip debugger, the use of the following RAM area is prohibited.

RL78/F25: F6300H to F64FFH

RL78/F22: FD300H to FD4FFH

4. When performing self-programming of the on-chip debugger, the use of the following RAM area is prohibited.

RL78/F25: F5F00H to F5F7FH

RL78/F22: FCF00H to FCF7FH

5. When using the hot plug-in function, the real-time RAM monitor (RRM) or dynamic memory modification (DMM) function, the use of the RAM area of the following products is prohibited.

RL78/F25: F6500H to F652FH

RL78/F22: FD500H to FD52FH

6. After the reset (external or internal reset) is released, the following RAM areas are used in the reset sequence and are undefined.

RL78/F25: FEF00H to FFEFFH

RL78/F22: FEF00H to FFEFFH

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** 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** 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.

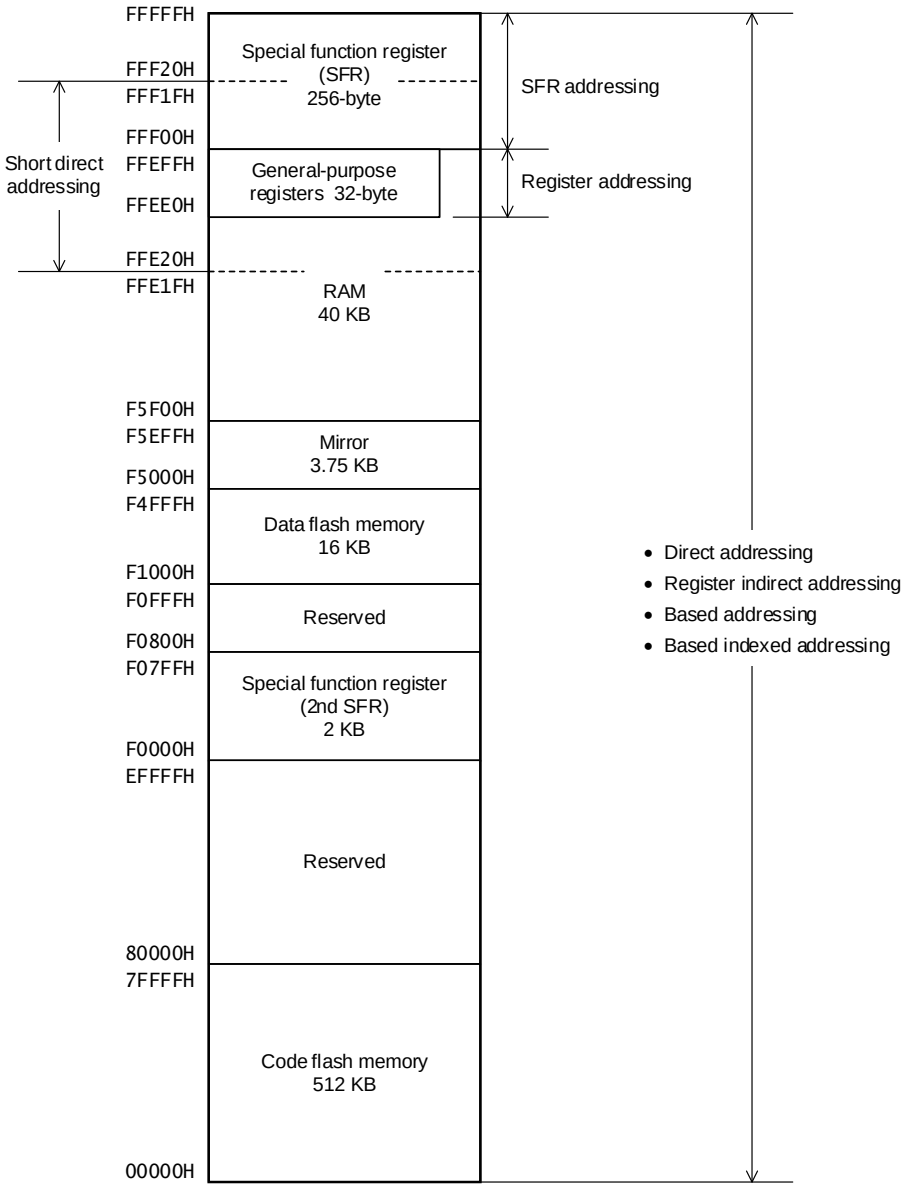
Caution Do not access addresses to which extended SFRs are not assigned.

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.

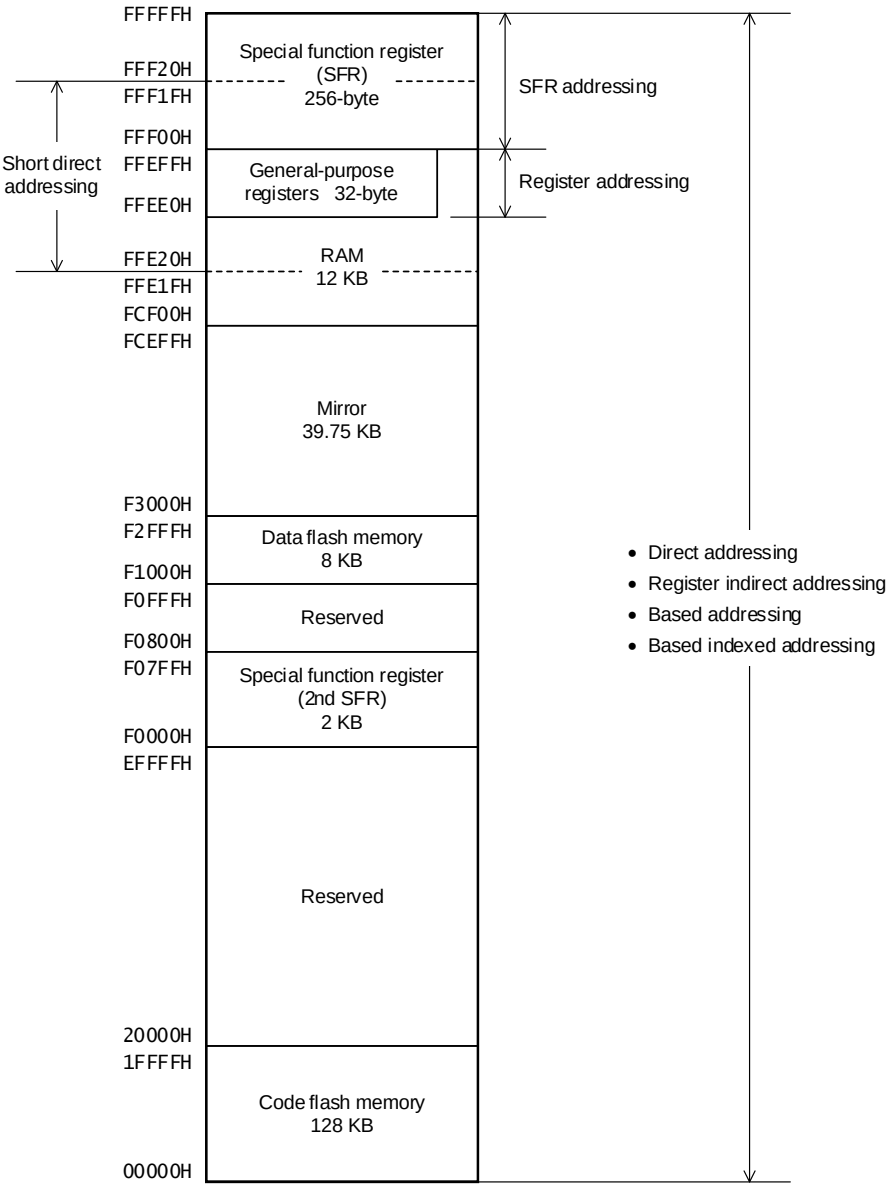
Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the RL78/F22 and RL78/F25, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. **Figure 3-4** and **Figure 3-5** show correspondence between data memory and addressing. For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3-4. Correspondence between Data Memory and Addressing (RL78/F25)



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. Correspondence between Data Memory and Addressing (RL78/F22)



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/F22 and RL78/F25 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)

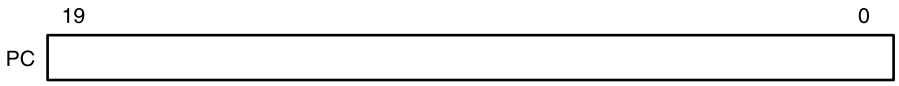
The program counter is a 20-bit register that holds the address information of the next program to be executed.

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 3-6. 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 3-7. 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, PRn3H) (see **22.3.3 Priority specification flag registers (PRxxL, PRxxH)**) 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 3-8. 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 3-9**.

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 tracing function of on-chip debugging.

RL78/F25: F6300H to F64FFH

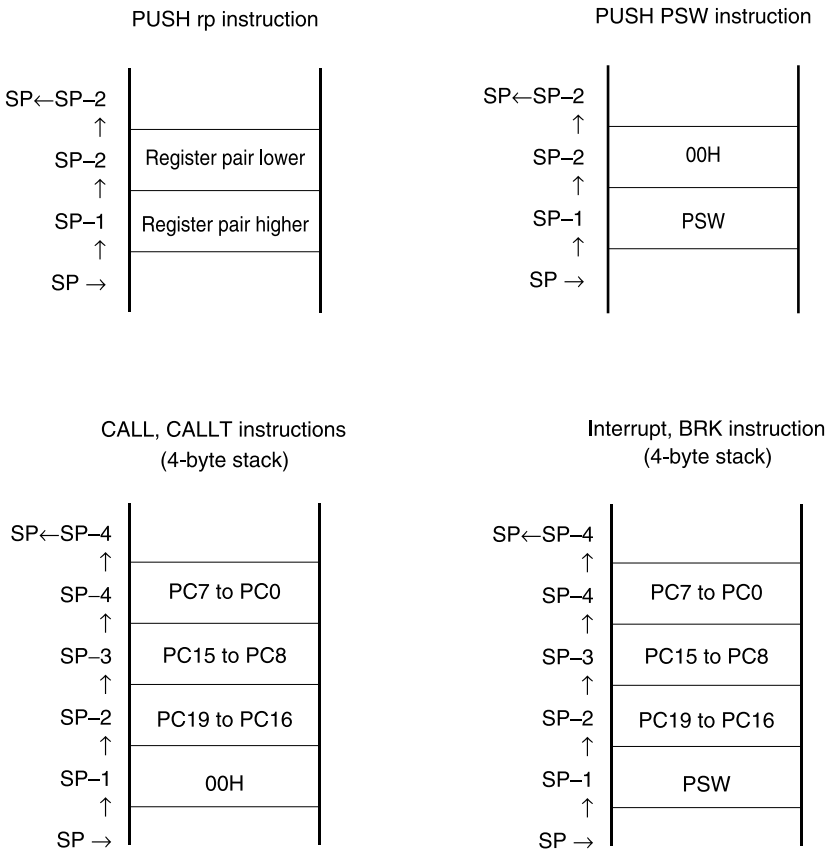
RL78/F22: FD300H to FD4FFH

4. 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.

RL78/F25: F6500H to F652FH

RL78/F22: FD500H to FD52FH

Figure 3-9. 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 consist 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).

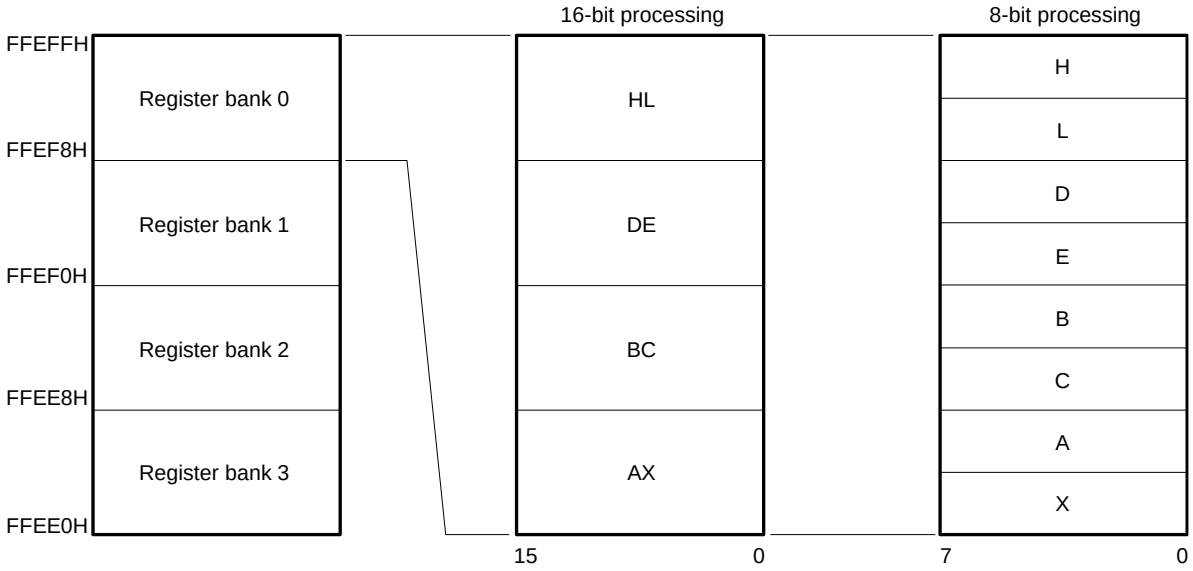
These registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

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.

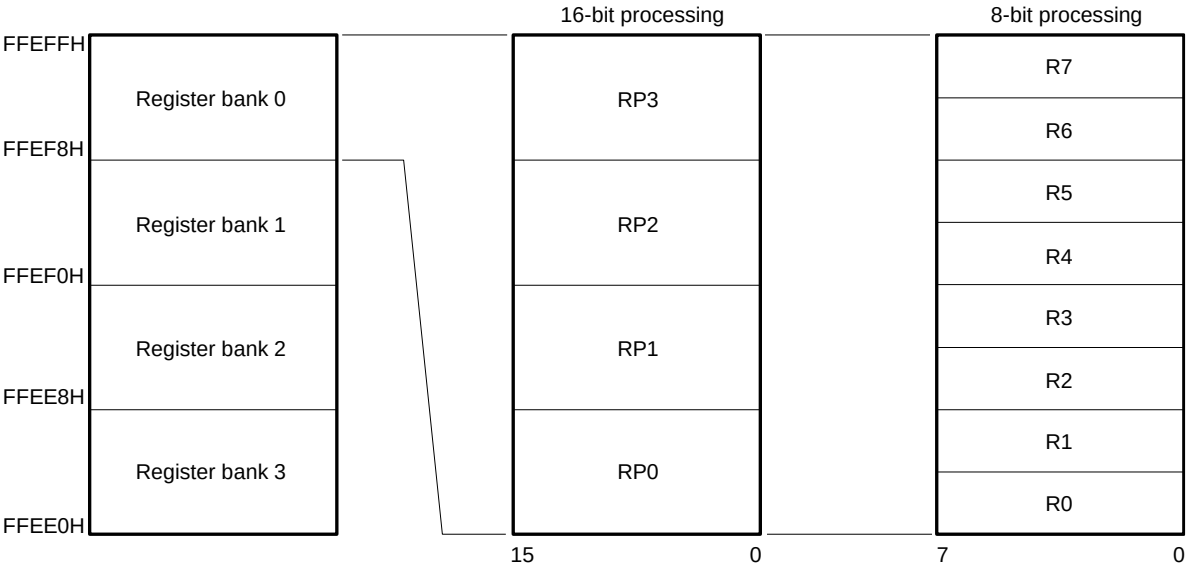
Caution It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

Figure 3-10. 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 3-11. 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	CS2	CS1	CS0

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 lists the SFRs.

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

Address	Special Function Register (SFR Symbol)	Reference Chapter
FFF00H to FFF0FH	Port registers (P0, P1, P3 to P10, P12 to P15)	CHAPTER 4
FFF10H to FFF13H	Serial data registers (SDR00, SDR01)	CHAPTER 16
FFF18H to FFF1BH	Timer data registers (TDR00, TDR01)	CHAPTER 6
FFF20H to FFF2FH	Port mode registers (PM0, PM1, PM3 to PM10, PM12, PM14, PM15)	CHAPTER 4
FFF30H	A/D converter access window register (ADWINR)	CHAPTER 12
FFF32H	Application accelerator unit access window register (AAUWINR)	CHAPTER 28
FFF34H	D/A conversion value setting registers (DACS0)	CHAPTER 13
FFF36H	D/A converter mode register (DAM)	
FFF37H	Key return mode register (KRM)	CHAPTER 23
FFF38H to FFF3BH	External interrupt edge enable registers (EGP0, EGN0, EGP1, EGN1)	CHAPTER 22
FFF48H to FFF4BH	Serial data registers (SDR10, SDR11)	CHAPTER 16
FFF50H to FFF52H	IICA related registers (IICA0, IICS0, IICF0)	CHAPTER 17
FFF54H, FFF55H	16-bit watch error correction register (SUBCUDW)	CHAPTER 9
FFF58H to FFF6FH	Timer RDe related registers (TRDGRC0, …)	CHAPTER 8
FFF74H to FFF8FH	Timer data registers (TDR02 to TDR07, TDR10 to TDR17)	CHAPTER 6
FFF92H to FFF9EH	RTC related registers (SEC, …)	CHAPTER 9
FFFA0H to FFFA6H	Clock generator related registers (CMC, …)	CHAPTER 5, 10
FFFA8H	Reset control flag register (RESF)	CHAPTER 25
FFFA9H, FFFAAH	LVD related registers (LVIM, LVIS)	CHAPTER 27
FFFABH	Watchdog timer enable register (WDTE)	CHAPTER 11
FFFACH	CRC input register (CRCIN)	CHAPTER 29
FFFB0H to FFFC6H	Flash related registers (FLSEC, …)	CHAPTER 33
FFFD0H to FFFEFH	Interrupt related registers (IF2L, …)	CHAPTER 22
FFFF0H to FFFF3H	Multiply and accumulation registers (MACRL, MACRH)	CHAPTER 36
FFFFEH	Processor mode control register (PMC)	CHAPTER 3

Note For the access size and reset value of each registers, refer to the corresponding chapters.

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 (!addr16.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 (!addr16). 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 (!addr16s). When specifying an address, describe an even address.

Table 3-6 lists the extended SFRs.

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 (1/2)

Address	Special Function Register (SFR Symbol)	Reference Chapter
F0016H to F006CH	Port related registers (PIORx, PITHLx, PUx, PIMx, POMx, PMCx)	CHAPTER 4
F0070H to F0072H	Noise filter registers (NFEN0, NFEN1, NFEN2)	CHAPTER 6, 16
F0073H to F0075H	Input switch registers (ISC, TIS0, TIS1)	CHAPTER 6, 22
F0076H	RAM start address setting register (RAMSAR)	CHAPTER 3
F0077H	Port mode select register (PMS)	CHAPTER 4
F0078H	Invalid memory access detection control register (IAWCTL)	CHAPTER 29
F0079H	Interrupt source determination flag register (INTFLG0)	CHAPTER 22
F007AH	Timer input select register 2 (TIS2)	CHAPTER 6
F007BH	LIN channel select register (LCHSEL)	CHAPTER 18
F007CH	Interrupt mask register (INTMSK)	CHAPTER 22
F007DH	Interrupt source determination flag register (INTFLG1)	
F0090H	Data flash control register (DFLCTL)	CHAPTER 33
F0098H to F009EH	Code flash bit error detection function related registers (CFERRCTLR, …)	CHAPTER 29
F00A0H, F00A8H	High-speed on-chip oscillator related registers (HIOTRM, HOCODIV)	CHAPTER 5
F00C0H to F00CFH	Flash related registers (FLPMC, …)	CHAPTER 33
F00D8H to F00DDH	SPM related registers (SPMCTLR, SPOFR, SPUFR)	CHAPTER 29
F00F3H	Operation speed mode control register (OSMC)	CHAPTER 5
F00F9H	Memory power reduction control register	CHAPTER 24
F00FEH	BCD correction result register (BCDADJ)	CHAPTER 35
F0100H to F0123H	SAU related registers (SSR00/SSR10, …)	CHAPTER 16
F0140H to F017FH	TAU related registers (TCR00/TCR10, …)	CHAPTER 6
F0180H to F01EDH, F01F8H	Capacitive sensing unit related registers (CTSUCRAL, ...)	CHAPTER 15
F01F0H	Unit select register (UTSEL)	CHAPTER 6, 16
F01F2H, F01F3H	Interrupt source determination flag register (INTFLG2, INTFLG3)	CHAPTER 22
F01F6H	SAU I^2S function control register (I2SER)	CHAPTER 16
F01F7H	RS-CANFD lite window access control register (CFDWINR)	CHAPTER 19
F0200H to F0207H	RAM ECC function related registers (ERADR, …)	CHAPTER 29
F0211H to F0225H	Port related registers (TSPMCx, PSRSEL, PSNZCNTx)	CHAPTER 4
F0227H	D/A converter mode register 2 (DAM2)	CHAPTER 13
F0228H to F022DH	PWM output delay control registers (PWMDLY0, PWMDLY1, PWMDLY2)	CHAPTER 6, 8
F0230H to F0234H	IICA related registers (IICCTL00, …)	CHAPTER 17
F0240H to F0243H	Timer RJ related registers (TRJCR0, …)	CHAPTER 7
F0248H to F024CH	PWMOPA related registers (OPCTL0, …)	CHAPTER 8
F024EH, F024FH	Decimation module control registers (TRDMBKCTL, TRDMBKCMP)	
F0250H to F029BH	Timer RDe related registers (TRDELC, …)	
F02A0H to F02A2H	CMP related registers (CMPCTL, CMPSEL, CMPMON)	CHAPTER 14
F02B0H to F02BFH	AAU related registers (ADTREG0, …) (page.0 to 2)	CHAPTER 28
F02C0H to F02C4H	Clock generator related registers (PER0 to 2, CANCKSEL, LINCKSEL)	CHAPTER 5
F02C5H	RTC clock select register (RTCCL)	CHAPTER 9
F02C6H	STOP status output control register (STPSTC)	CHAPTER 24
F02C7H	A/D conversion clock control register (ADCKS)	CHAPTER 12

Note For the access size and reset value of each registers, refer to the corresponding chapters.

Table 3-6. Extended SFR (2nd SFR) List (2/2)

Address	Special Function Register (SFR Symbol)	Reference Chapter
F02C8H to F02CBH	Clock generator related registers (PLLCTL, PLLSTS, CKSEL, MDIV)	CHAPTER 5
F02CCH	POR/CLM reset confirmation register (POCRES)	CHAPTER 25
F02CEH	Clock monitor test register (CLMTES)	CHAPTER 29
F02D0H to F02EEH	DTC related registers (HDTCCR0, …)	CHAPTER 20
F02F0H to F02F3H	Flash memory CRC related registers (CRC0CTL, PGCRCL)	CHAPTER 29
F02F9H to F02FBH	General-purpose CRC related registers (CRCMD, CRCD)	
F0300H to F0413H	RS-CANFD lite related registers (C0NCFGL/C1NCFGL, …)	CHAPTER 19
F0420H to F067FH	RS-CANFD lite related global RAM window registers (page.0 to 3)	
F06A0H to F06AFH	A/D mirror registers (ADDR0M to ADDR7M)	CHAPTER 12
F06B0H to F06BFH	12-bit A/D converter related window registers (page.0 to 14)	
F06C1H to F06EEH	RLIN3 related registers (LWBR0/1/2, …)	CHAPTER 18
F06F0H, F06F1H	Timer RJ counter register 0 (TRJ0)	CHAPTER 7
F0780H to F0799H	Event output destination select registers (ELSELRx)	CHAPTER 21
F07C0H to F07D1H	CAN RAM-ECC function related registers (CFDECCTL0/1, …)	CHAPTER 29

Note For the access size and reset value of each registers, refer to the corresponding chapters.

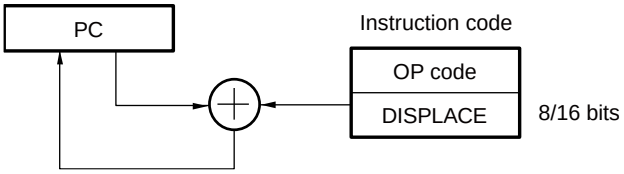
3.3 Instruction Address Addressing

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 3-12. 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 3-13. Example of CALL !!addr20/BR !!addr20

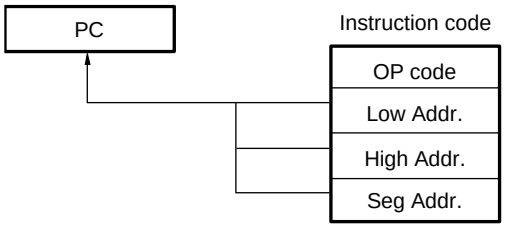
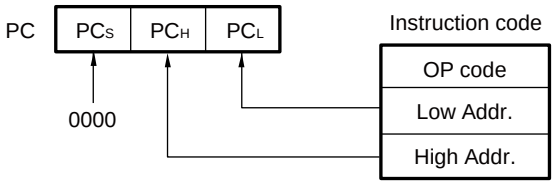


Figure 3-14. Example of CALL !addr16/BR !addr16

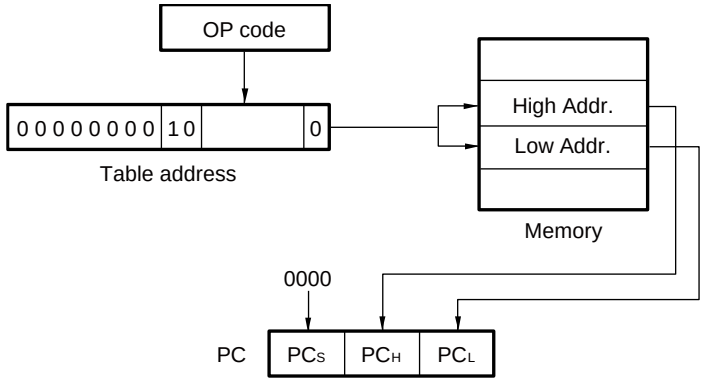


3.3.3 Table Indirect Addressing

[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 3-15. Outline of Table Indirect Addressing

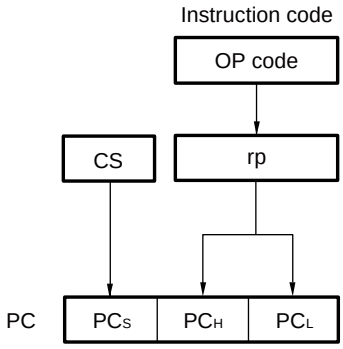


3.3.4 Register Indirect Addressing

[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 indirect addressing can be applied only to the CALL AX, BC, DE, HL, and BR AX instructions.

Figure 3-16. Outline of Register Indirect Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied Addressing

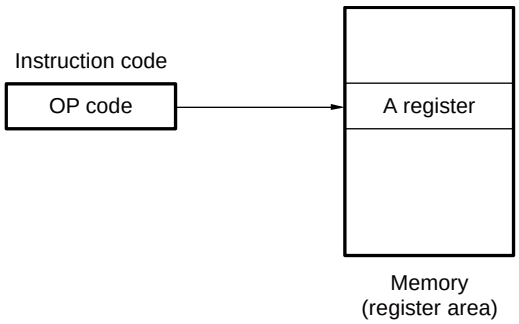
[Function]

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 3-17. Outline of Implied Addressing



3.4.2 Register Addressing

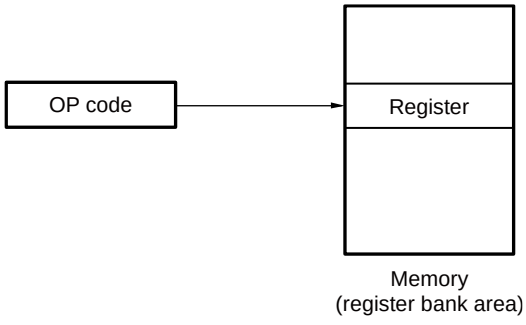
[Function]

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 3-18. Outline of Register Addressing



3.4.3 Direct Addressing

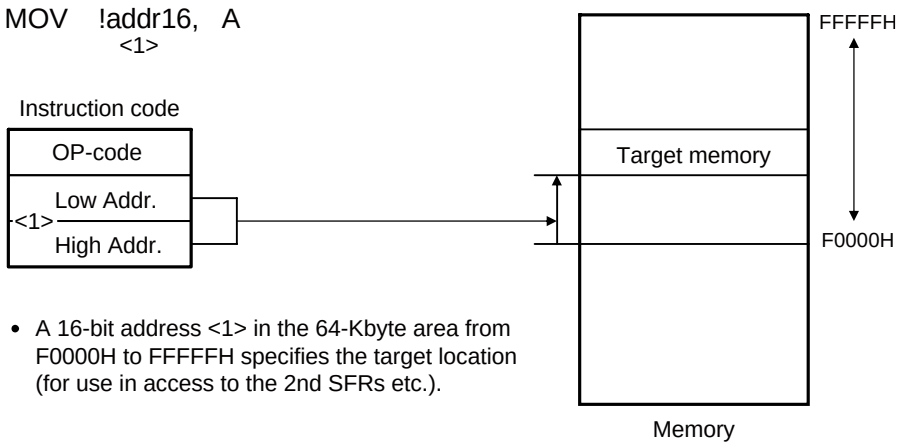
[Function]

Direct addressing uses immediate data in the instruction word as an operand address to directly specify the target address.

[Operand format]

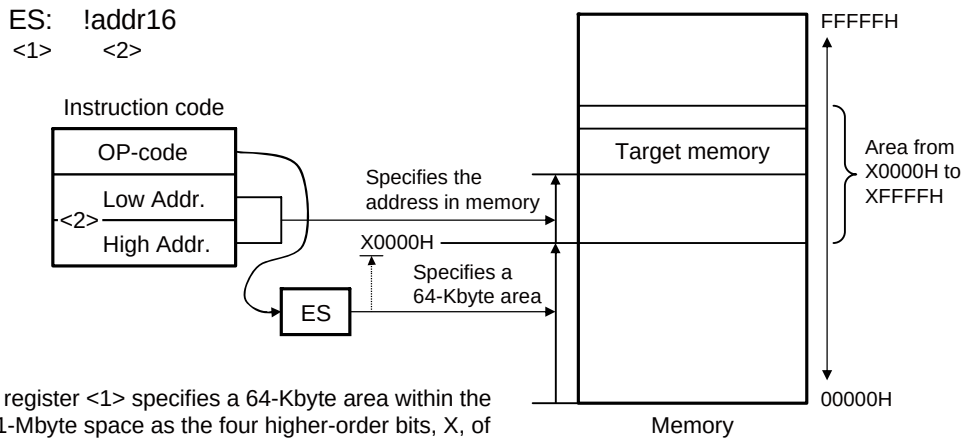
Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3-19. 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 3-20. 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

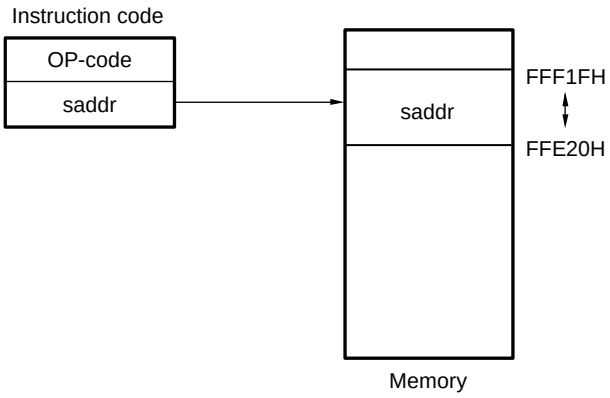
[Function]

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 (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) (only the space from FFE20H to FFF1FH is specifiable)

Figure 3-21. 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

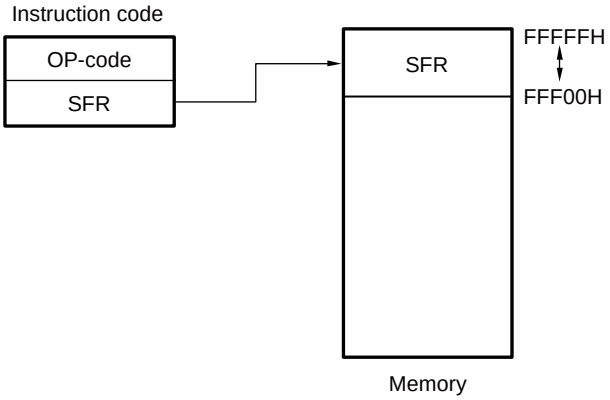
[Function]

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 3-22. Outline of SFR Addressing



3.4.6 Register Indirect Addressing

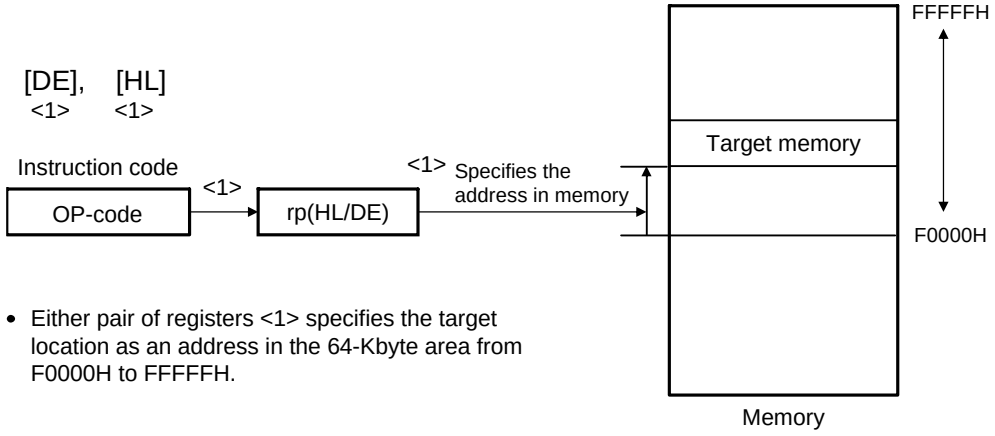
[Function]

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]

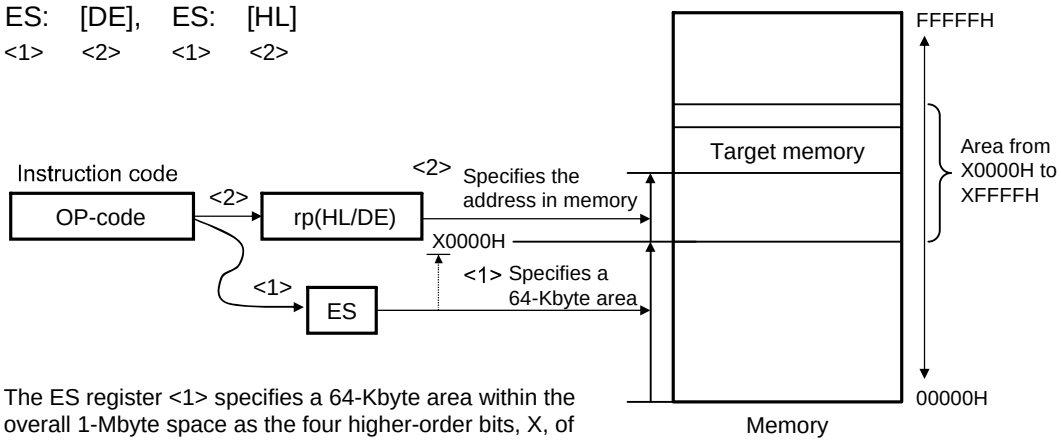
Identifier	Description
–	[DE], [HL] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

Figure 3-23. 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 3-24. 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

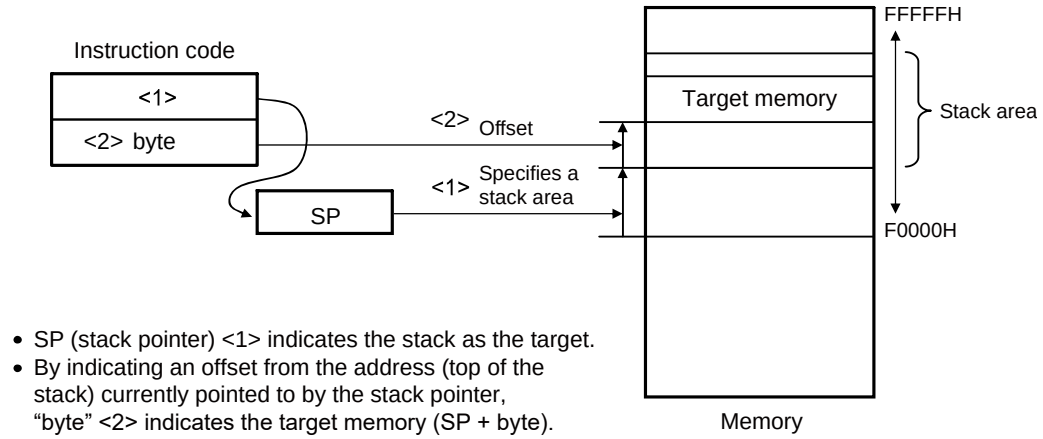
[Function]

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]

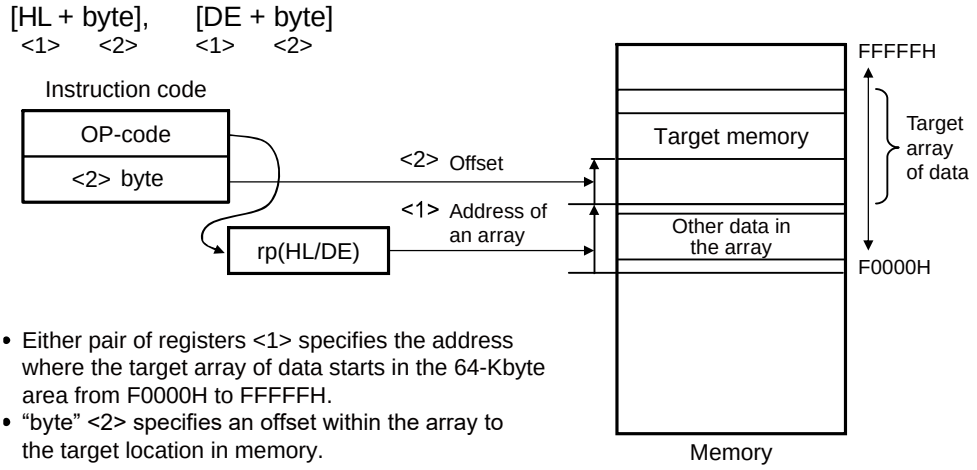
Identifier	Description
–	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
–	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
–	word[BC] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL + byte], ES:[DE + byte] (higher 4-bit 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 3-25. Example of [SP+byte]



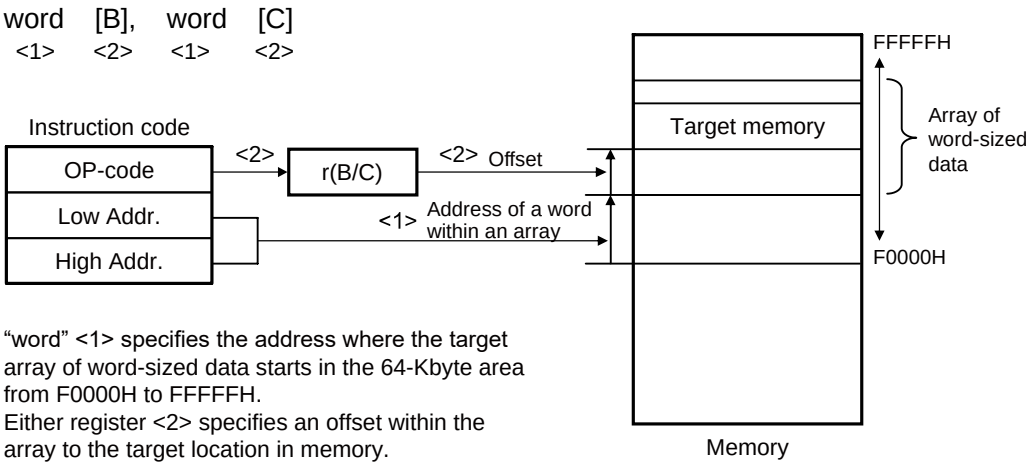
- SP (stack pointer) <1> indicates the stack as the target.
- By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, “byte” <2> indicates the target memory (SP + byte).

Figure 3-26. Example of [HL + byte], [DE + byte]



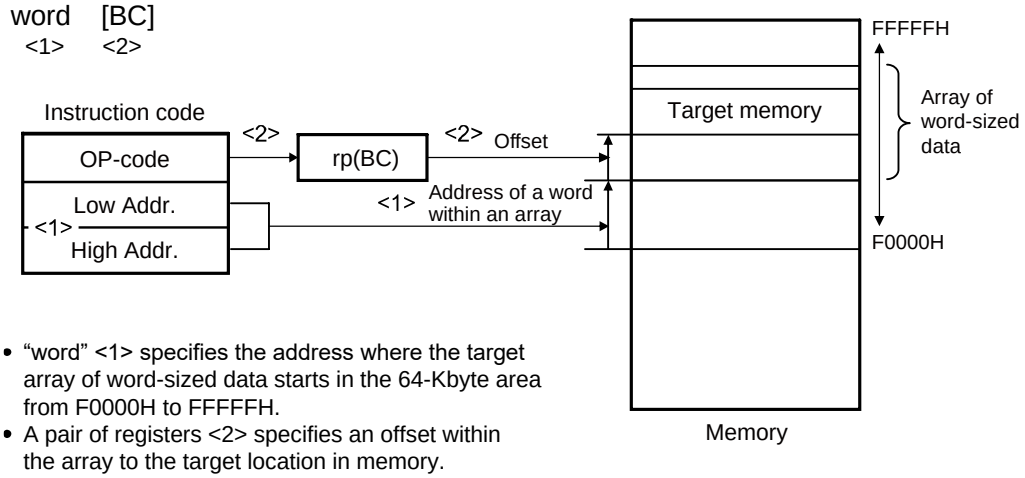
- Either pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- "byte" <2> specifies an offset within the array to the target location in memory.

Figure 3-27. Example of word[B], word[C]



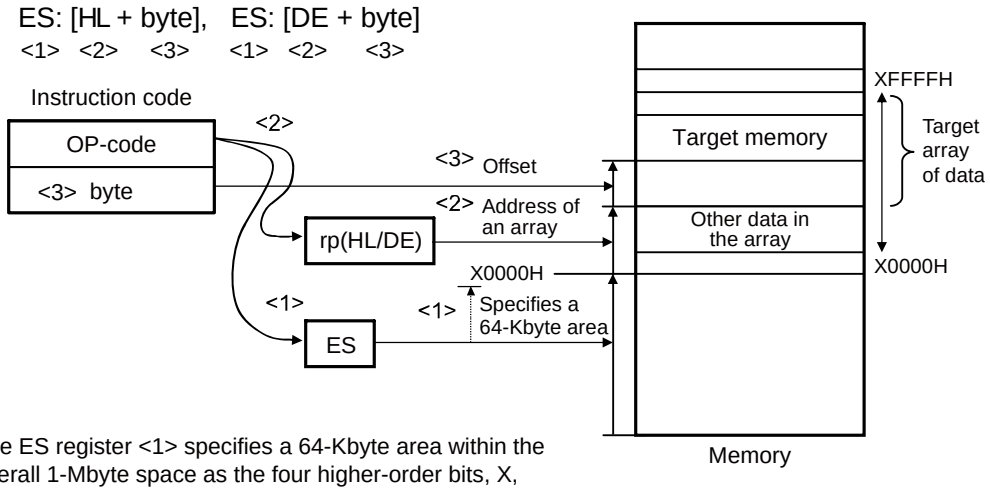
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3-28. Example of word[BC]



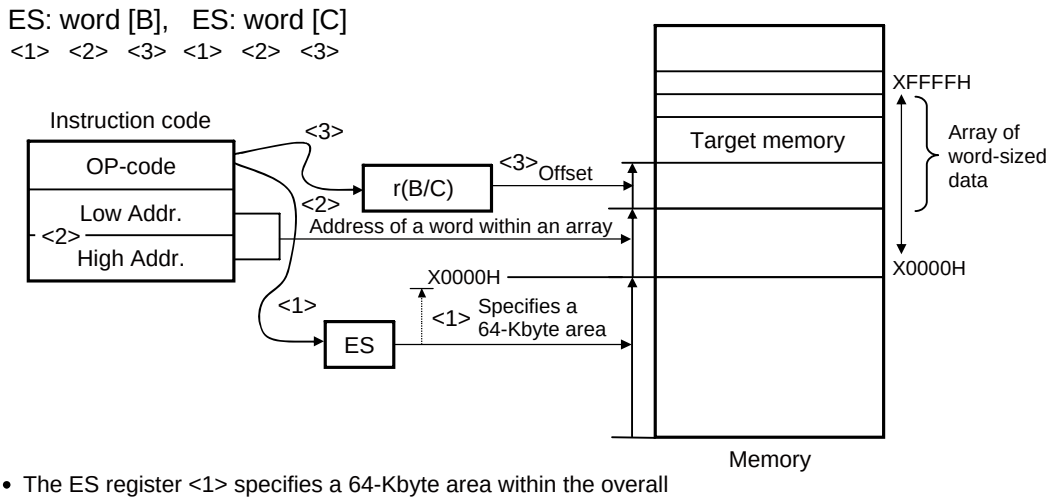
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

Figure 3-29. Example of ES:[HL + byte], ES:[DE + byte]



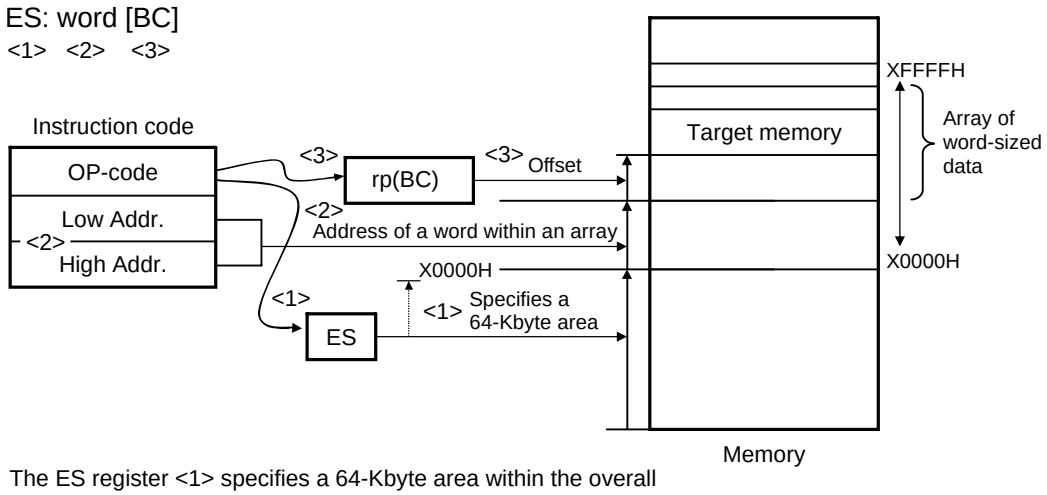
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- “byte” <3> specifies an offset within the array to the target location in memory.

Figure 3-30. 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 3-31. 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

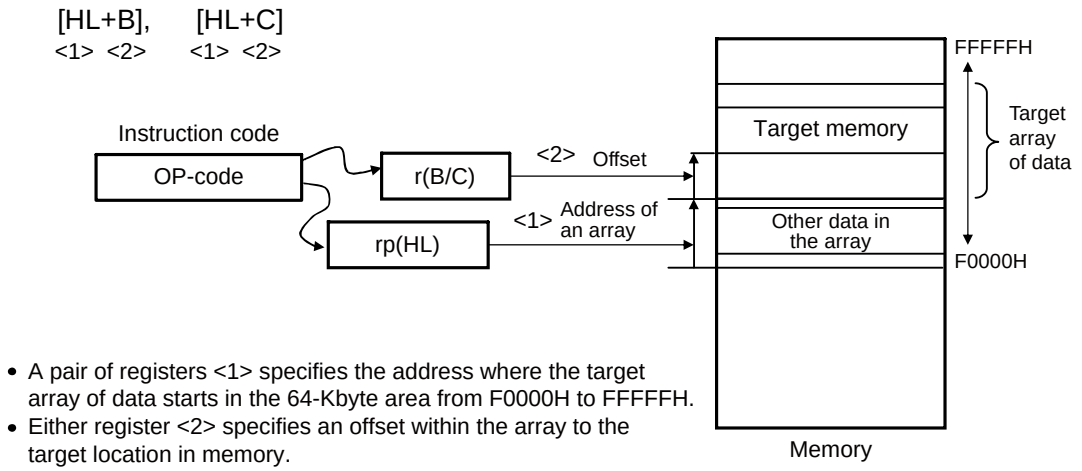
[Function]

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]

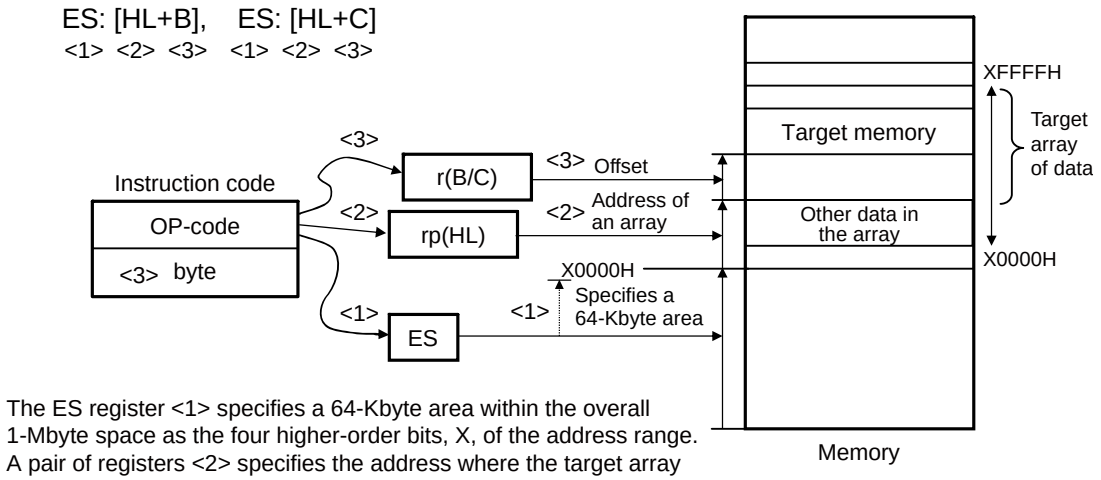
Identifier	Description
–	[HL + B], [HL + C] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL + B], ES:[HL + C] (higher 4-bit addresses are specified by the ES register)

Figure 3-32. 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 3-33. 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.

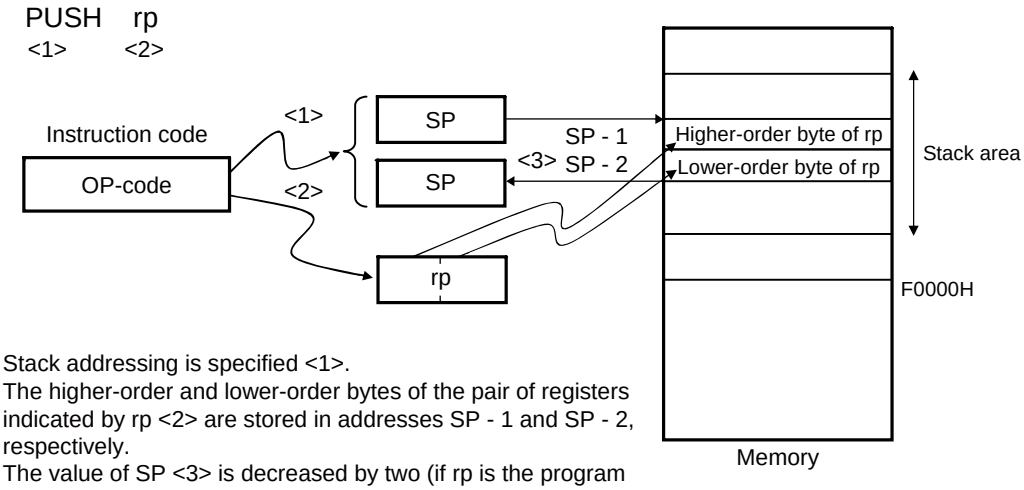
Only the internal RAM area can be set as the stack area.

[Operand format]

Identifier	Description
–	PUSH PSW AX/BC/DE/HL POP PSW AX/BC/DE/HL CALL/CALLT RET BRK RETB (Interrupt request generated) RETI

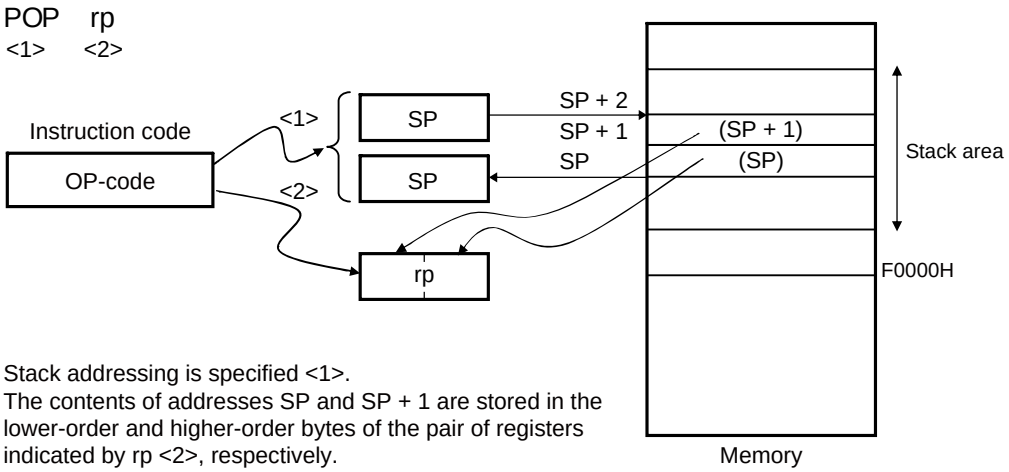
Each stack operation saves or restores data as shown in **Figure 3-34** to **Figure 3-39**.

Figure 3-34. Example of PUSH rp



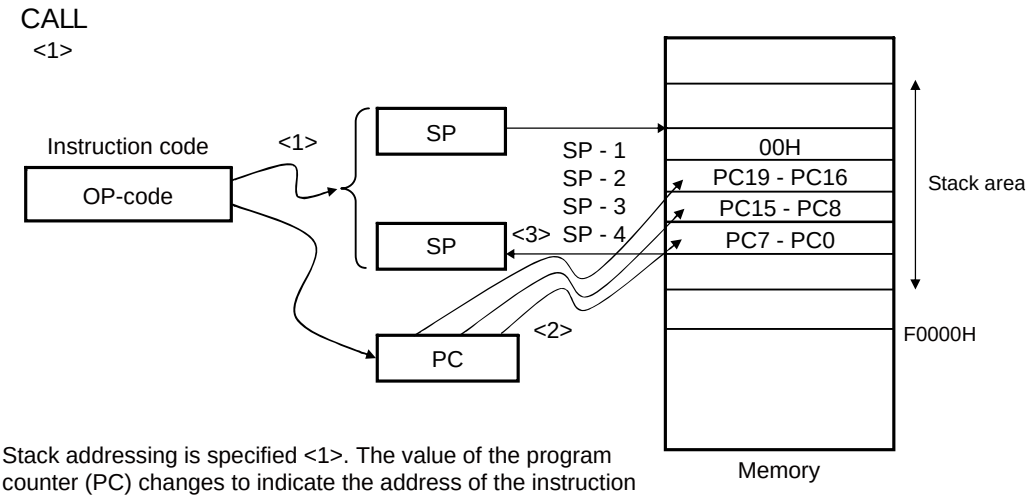
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP - 1 and SP - 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP - 1 and 0 is stored in SP -2).

Figure 3-35. Example of POP



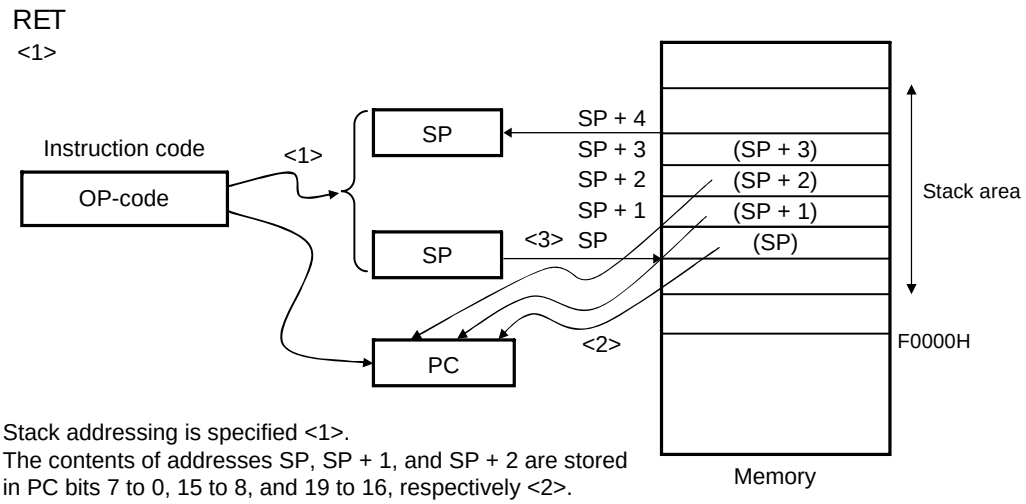
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP + 1 is stored in the PSW).

Figure 3-36. Example of CALL, CALLT



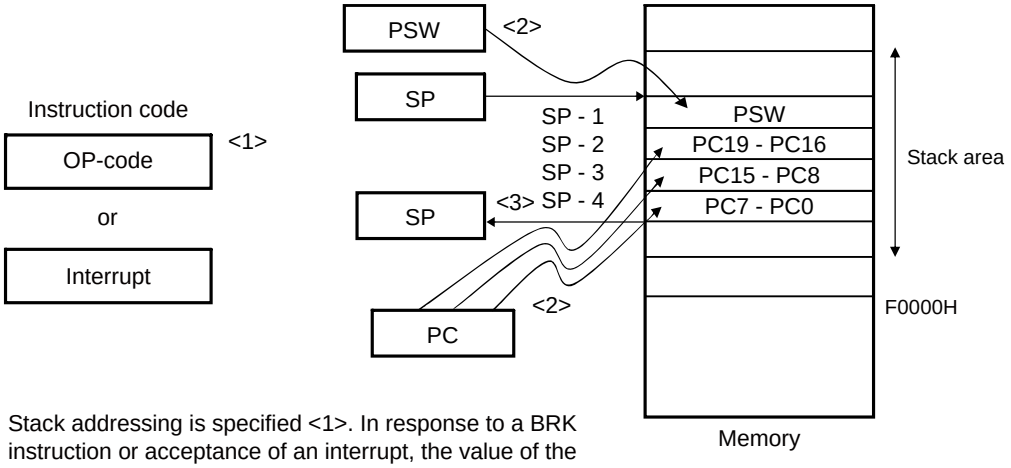
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP - 1, SP - 2, SP - 3, and SP - 4, respectively <2>.
- The value of the SP <3> is decreased by 4.

Figure 3-37. Example of RET



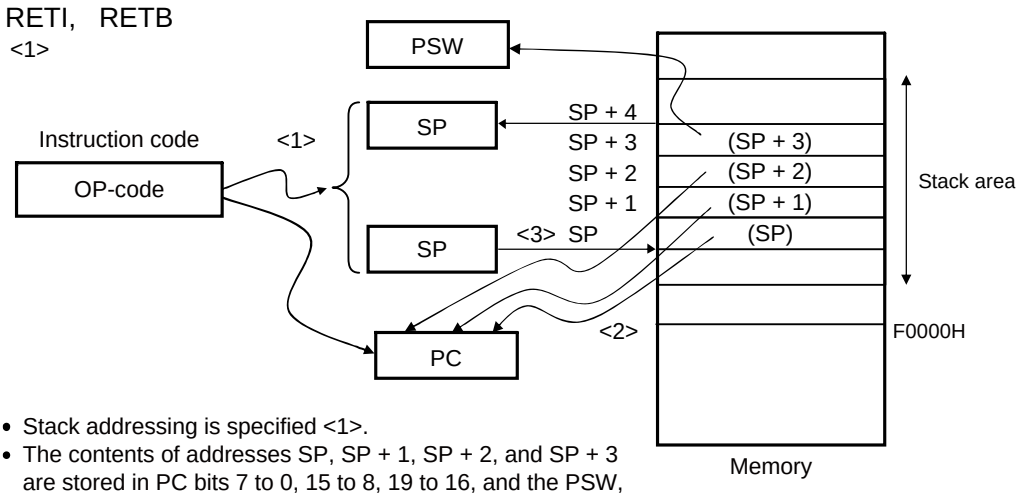
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3-38. 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 3-39. 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.

3.5 Register Descriptions

Table 3-7. Register Configuration

Address	Register Name	Symbol	After reset	Access Size
F0076H	RAM start address setting register	RAMSAR	EFH	8
F0078H	Invalid memory access detection control register	IAWCTL	00H	8
FFFFEH	Processor mode control register	PMC	00H	1, 8

Remark See **3.1.2 Mirror Area** for the processor mode control register (PMC).

3.5.1 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 3-40. 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	Invalid memory access detection control bit
0	Disable the detection of invalid memory access.
1	Enable the detection of invalid memory access.

GRAM1	GRAM0	RAM Guard Space Note
0	0	RAM guard function is disabled.
0	1	The 128 bytes of space starting at the start address in the RAM specified with RAMSAR register.
1	0	The 256 bytes of space starting at the start address in the RAM specified with RAMSAR register.
1	1	The 512 bytes of space starting at the start address in the RAM specified with RAMSAR register.

Note The RAM start address differs depending on the size of the RAM provided with the product. And It can be set with RAMSAR register. When setting GRAM* bits in IAWCTL register, set the RAMSAR register before that.

(2) RAM Start Address Setting Register (RAMSAR)

Change to allow the user to select RAM start address. The RAMSAR registers to be added are shown below.
The RAMSAR register can be set by an 8-bit memory manipulation instruction.

Figure 3-41. Format of RAM Start Address Setting Register (RAMSAR) Note

Address: F0076H After reset: EFH R/W

Symbol	7	6	5	4	3	2	1	0
RAMSAR	RAMSAR[7:0]							

Bit	Bit name	Description
7-0	–	RAM start address setting bits. (Setting range: 5FH to FDH) These bits set bit 15 to bit 8 of the RAM start address. For example, when 9FH is set: RAM guard start address = F9F00H

Note RAMSAR register can be written only once after reset release. If the value is set, access the RAM after an interval of 2 clocks or more.

Table 3-8. RAM Guard Function Setting Example

RAMSAR Register Note	IAWCTL.GRAM[1:0] Bits	Valid RAM Area	RAM Guard Space
5FH	01B (128 bytes)	F5F00H-FFEFFH (40 KB)	F5F00H to F5F7FH
	10B (256 bytes)		F5F00H to F5FFFH
	11B (512 bytes)		F5F00H to F60FFH
9FH	01B (128 bytes)	F9F00H to FFEFFH (24KB)	F9F00H to F9F7FH
	10B (256 bytes)		F9F00H to F9FFFH
	11B (512 bytes)		F9F00H to FA0FFH
AFH	01B (128 bytes)	FAF00H to FFEFFH (20KB)	FAF00H to FAF7FH
	10B (256 bytes)		FAF00H to FAFFFH
	11B (512 bytes)		FAF00H to FB0FFH
CFH	01B (128 bytes)	FCF00H to FFEFFH (12KB)	FCF00H to FCF7FH
	10B (256 bytes)		FCF00H to FCFFFH
	11B (512 bytes)		FCF00H to FD0FFH
DFH	01B (128 bytes)	FDF00H to FFEFFH (8KB)	FDF00H to FDF7FH
	10B (256 bytes)		FDF00H to FDFFFH
	11B (512 bytes)		FDF00H to FE0FFH
EFH	01B (128 bytes)	FEF00H to FFEFFH (4KB)	FEF00H to FEF7FH
	10B (256 bytes)		FEF00H to FEFFFH
	11B (512 bytes)		FEF00H to FF0FFH

Note Be sure to set it within the memory range of the product to be used.

3.5.2 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 3-42. 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	Invalid memory access detection control bit
0	Disable the detection of invalid memory access.
1	Enable the detection of invalid memory access.

GPORT	Port function SFR guard control bit
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. [Guarded SFR] PMxx, PUxx, PIMxx, POMxx, PMCxx, TSPMCxx, PITHLxx, PIORx Note

GINT	Interrupt function SFR guard control bit
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. [Guarded SFR] IFxx, MKxx, PRxx, EGPx, EGNx

GCSC	Clock control function SFR guard control bit
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. [Guarded SFR] CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS, CANCKSEL, LINCKSEL, CKSEL, PLLCTL, MDIV, RTCCL, POCRES, STPSTC, CLMTES, ADCKS, PSMCR

Note Pxx (Port register) is not guarded.

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} represents EV_{DD0} and EV_{DD1}.

Table 4-1. Pin I/O Buffer Power Supplies

(1) 24-pin products

Power Supply	Corresponding Pins
V_{DD}	• Port pins other than P80 to P82
AV_{DD}	• P80 to P82

(2) 32-pin products

Power Supply	Corresponding Pins
V_{DD}	• Port pins other than P80 to P85
AV_{DD}	• P80 to P85

(3) 48-pin products

Power Supply	Corresponding Pins
V_{DD}	• Port pins other than P80 to P87 and P90 to P92
AV_{DD}	• P80 to P87, P90 to P92

(4) 64-pin products

Power Supply	Corresponding Pins
EV_{DD0}	• Port pins other than P80 to P87, P90 to P96, P121 to P124, and P137
V_{DD}	• P121 to P124, P137 • Pins other than port pins
AV_{DD}	• P80 to P87, P90 to P96

(5) 80-pin products

Power Supply	Corresponding Pins
EV_{DD0}	• Port pins other than P80 to P87, P90 to P97, P121 to P124, and P137
V_{DD}	• P121 to P124, P137 • Pins other than port pins
AV_{DD}	• P80 to P87, P90 to P97

(6) 100-pin products

Power Supply	Corresponding Pins
EV_{DD0}, EV_{DD1}	• Port pins other than P80 to P87, P90 to P97, P100 to P105, P121 to P124, and P137
V_{DD}	• P121 to P124, P137 • Pins other than port pins
AV_{DD}	• P80 to P87, P90 to P97, P100 to P105

The RL78/F22 and RL78/F25 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) • Port registers (P0, P1, P3 to P10, P12 to P15) • Pull-up resistor option registers (PU0, PU1, PU3 to PU7, PU10, PU12, PU14, PU15) • Port input mode registers (PIM1, PIM3, PIM5 to PIM7, PIM12) • Port output mode registers (POM1, POM3, POM6, POM7, POM12) • Port mode control registers (PMC7 to PMC10, PMC12) • Touch sensor port mode control registers (TSPMC1, TSMPC5 to TSPMC7) • Peripheral I/O redirection registers (PIOR0 to PIOR9) • Port input threshold control registers (PITHL1, PITHL3 to PITHL7, PITHL10, PITHL12, PITHL15) • Port output slew rate select register (PSRSEL) • SNOOZE status output control registers 0 to 3 (PSNZCNT0 to PSNZCNT3) • Port mode select register (PMS)
Port	• 24-pin products Total: 18 (CMOS I/O: 15, CMOS input: 3) • 32-pin products Total: 26 (CMOS I/O: 23, CMOS input: 3) • 48-pin products Total: 42 (CMOS I/O: 36, CMOS input: 5, CMOS output: 1) • 64-pin products Total: 56 (CMOS I/O: 50, CMOS input: 5, CMOS output: 1) • 80-pin products Total: 72 (CMOS I/O: 66, CMOS input: 5, CMOS output: 1) • 100-pin products Total: 90 (CMOS I/O: 84, CMOS input: 5, CMOS output: 1)

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
Name	I/O		
P00	Input	1	×
	Output	0	(TO05 output = 0) Note 1
P01	Input	1	×
	Output	0	(TO04 output = 0) Note 1
P02	Input	1	×
	Output	0	(TO06 output = 0) Note 1
P03	Input	1	×
	Output	0	(RTC1HZ output = 0) Note 2

Important To use the port 0 as a general-purpose port, set the alternate pin function output to the level indicated by the "Alternate Function Setting When Using Pin as Port".

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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR1, 3, 9).

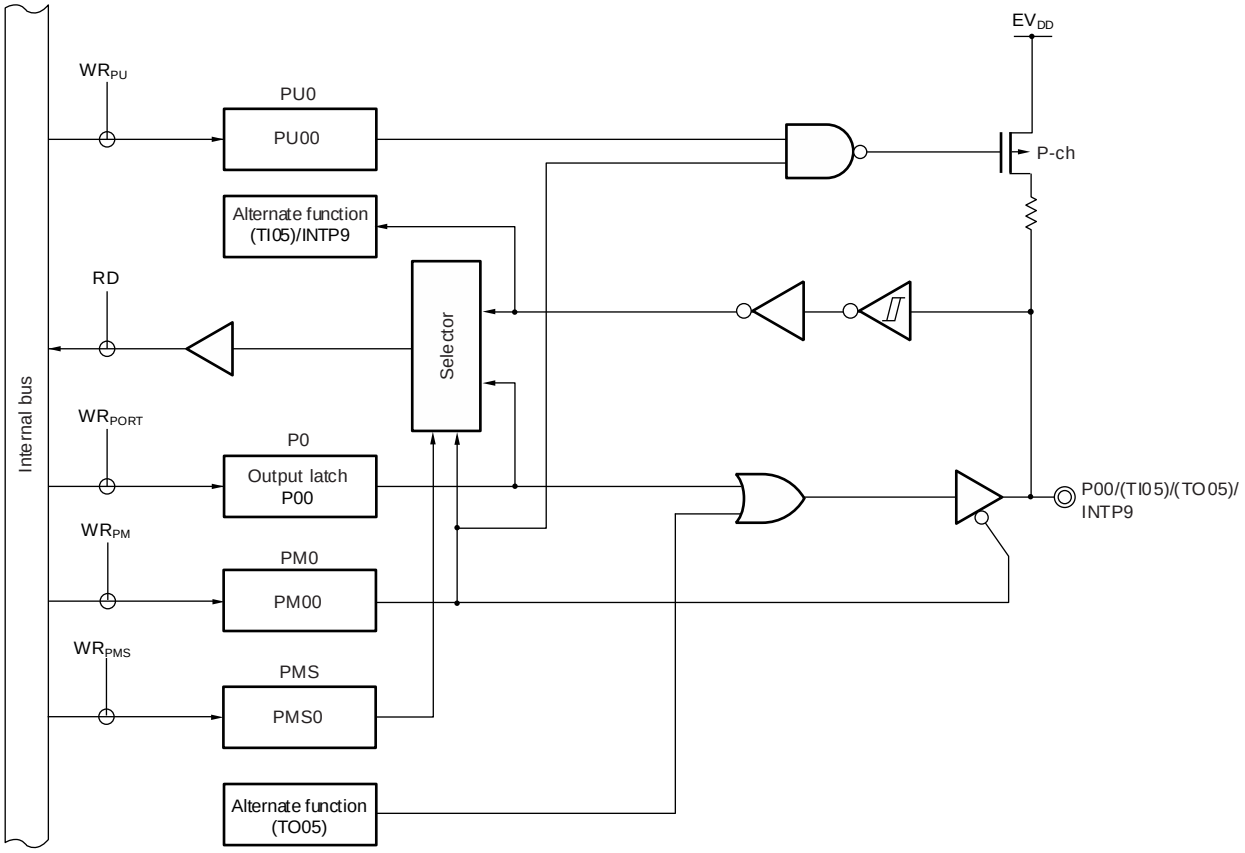
2. When a pin sharing the output (1 Hz) function of the RTC1HZ pin is to be used as a general-purpose port pin, the RCLOE1 bit of the real-time clock control register 0 (RTCC0) must have the same setting as its initial value. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR8).

Remarks ×: Don't care

PM0x: Bit of port mode register 0

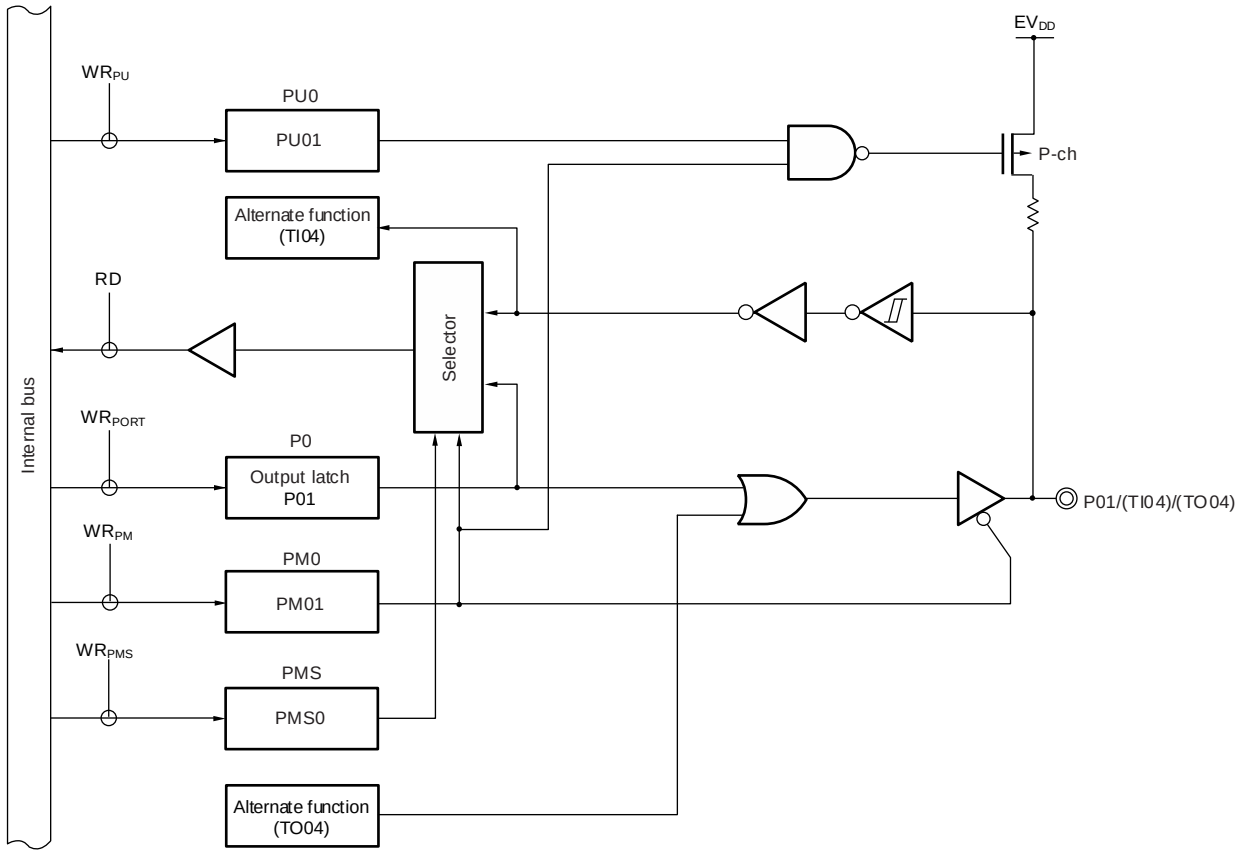
For example, **Figure 4-1** to **Figure 4-4** show block diagrams of port 0 for 100-pin products.

Figure 4-1. Block Diagram of P00



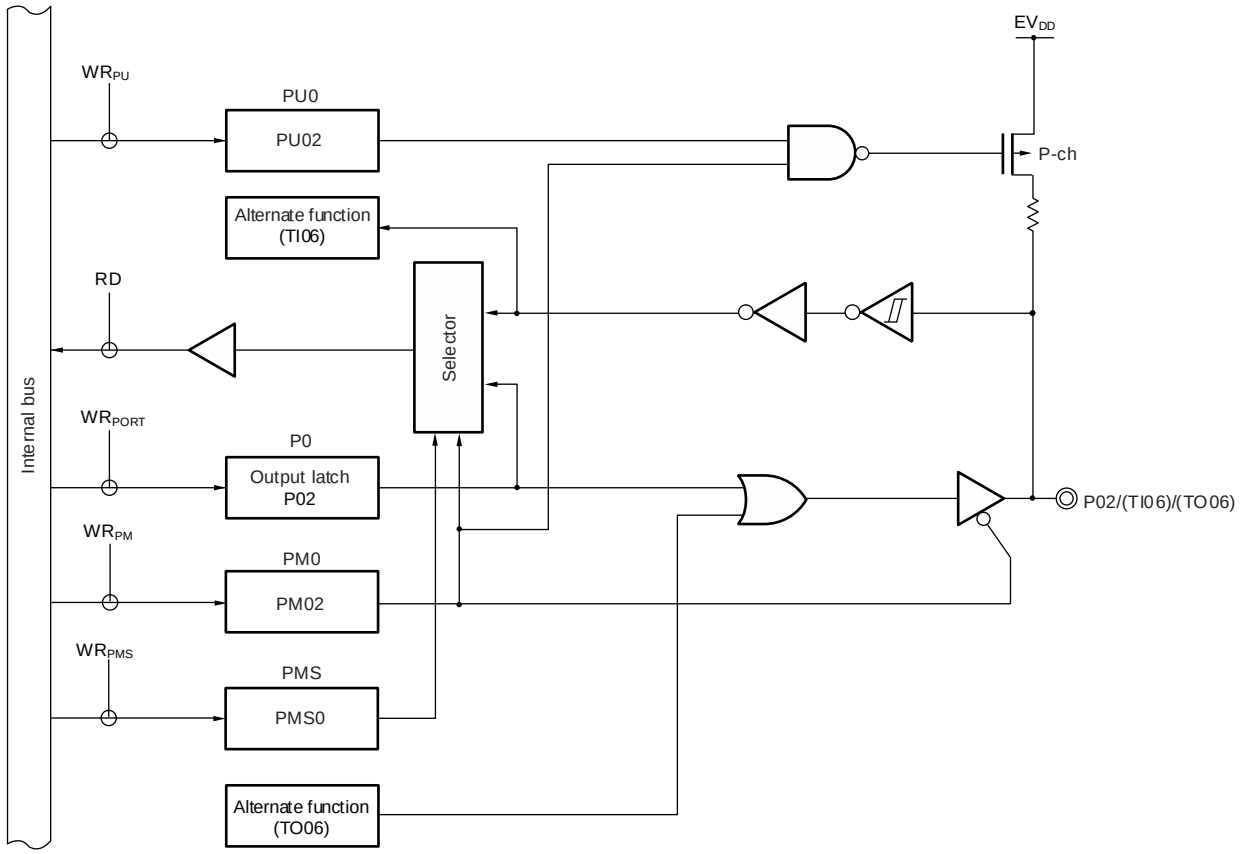
P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

Figure 4-2. Block Diagram of P01



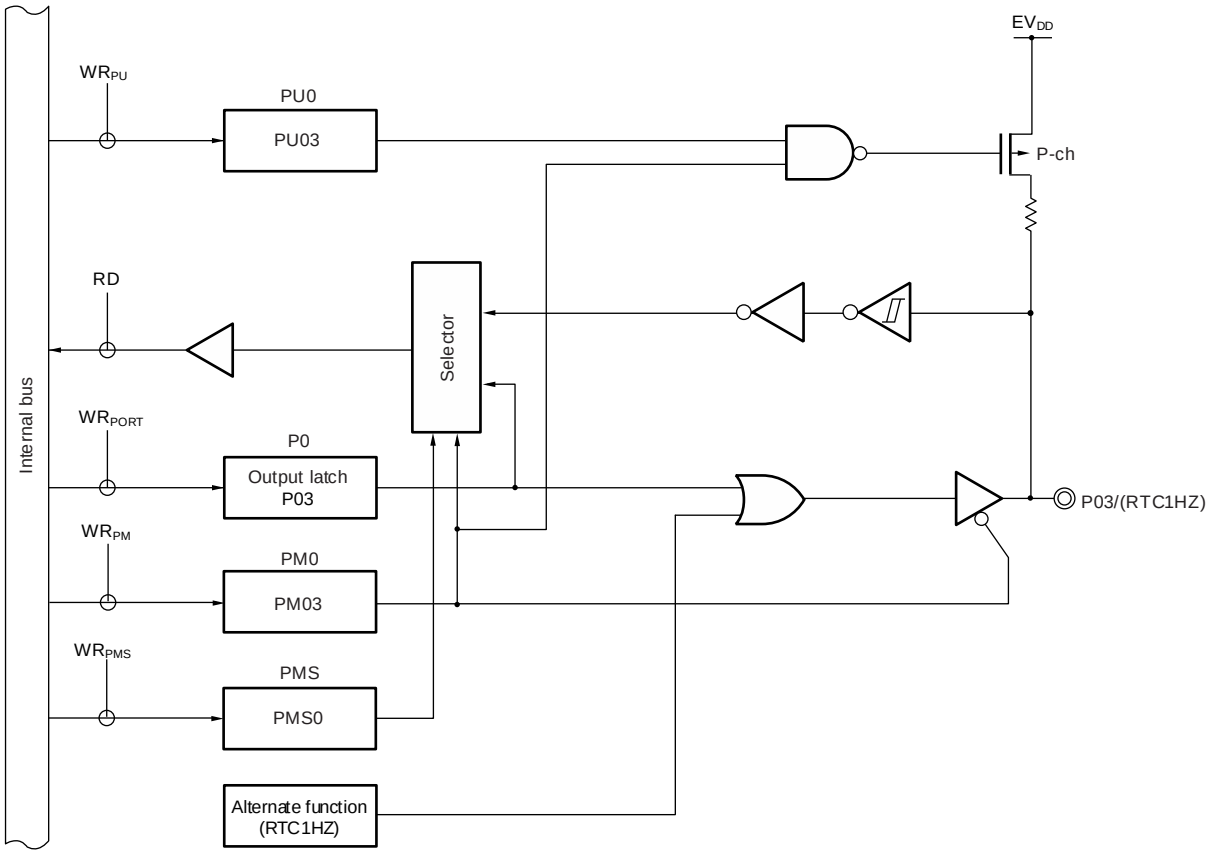
P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

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, SNOOZE status output, and capacitance measurement.

Reset signal generation sets this port to input mode.

Table 4-4. Settings of Registers When Using Port 1 (1/2)

Pin name		PM1x	PIM1x	POM1x	TSPMC1x	PITHL1x	Alternate Function Setting	Remarks
Name	I/O							
P10	Input	1	0	×	0	0	×	CMOS input (Schmitt1 input)
						1		CMOS input (Schmitt3 input)
			1	×	0	0	×	TTL input
	Output	0	×	0	0	×	SCK10/SCL10 output = 1 Note 1 TO13 output = 0 Note 2 TRJO0 output = 0 Note 3 LTXD1 output = 1 Note 8 CTXD0 output = 1 Note 9	CMOS output
			×	1	0	×		N-ch O.D output
P11	Input	1	0	×	0	0	×	CMOS input (Schmitt1 input)
						1		CMOS input (Schmitt3 input)
			1	×	0	0	×	TTL input
	Output	0	×	0	0	×	SDA10 output = 1 Note 1 TO12 output = 0 Note 2 (TRDIOB0 output = 0) Note 5	CMOS output
			×	1	0	×		N-ch O.D output
P12	Input	1	–	×	0	–	×	
	Output	0	–	0	0	–	TO11 output = 0 Note 2 SO10/TXD1 output = 1 Note 1 SNZOUT3 output = 0 Note 7 (TRDIOD0 output = 0) Note 5	CMOS output
			–	1	0	–		N-ch O.D output

(**Notes** and **Remarks** are listed on the next page of **Table 4-4 (2/2)**.)

Table 4-4. Settings of Registers When Using Port 1 (2/2)

Pin name		PM1x	PIM1x	POM1x	TSPMC1x	PITHL1x	Alternate Function Setting	Remarks
Name	I/O							
P13	Input	1	0	×	0	0	×	CMOS input (Schmitt1 input)
						1		CMOS input (Schmitt3 input)
			1	×	0	0	×	TTL input
	Output	0	×	0	0	×	TRDIOA0 output = 0 Note 5 SDA01 output = 1 Note 1 TO04 output = 0 Note 2 LTXD0 output = 1 Note 8	CMOS output
			×	1	0	×		N-ch O.D output
P14	Input	1	0	×	0	0	×	CMOS input (Schmitt1 input)
						1		CMOS input (Schmitt3 input)
			1	×	0	0	×	TTL input
	Output	0	×	0	0	×	TRDIOC0 output = 0 Notes 4, 5 SCK01/SCL01 output = 1 Note 1 TO06 output = 0 Note 2	CMOS output
			×	1	0	×		N-ch O.D output
P15	Input	1	–	×	0	–	×	
	Output	0	–	0	0	–	TRDIOA1 output = 0 Note 5 TO05 output = 0 Note 2 SO00/TXD0 output = 1 Note 1 RTC1HZ output = 0 Note 6 (TRDIOA0 output = 0) Note 5	CMOS output
			–	1	0	–		N-ch O.D output
P16	Input	1	0	×	0	0	×	CMOS input (Schmitt1 input)
						1		CMOS input (Schmitt3 input)
			1	×	0	0	×	TTL input
	Output	0	×	0	0	×	SDA00 output = 1 Note 1 TRDIOC1 output = 0 Note 5 TO02 output = 0 Note 2	CMOS output
			×	1	0	×		N-ch O.D output
P17	Input	1	0	×	0	0	×	CMOS input (Schmitt1 input)
						1		CMOS input (Schmitt3 input)
		1	1	×	0	0	×	TTL input
	Output	0	×	0	0	×	TRDIOB1 output = 0 Note 5 SCK00/SCL00 output = 1 Note 1 TO00 output = 0 Note 2	CMOS output
			×	1	0	×		N-ch O.D output

Important To use the port 1 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

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

Notes

1. When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn and CKOmn bits 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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).
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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR1, 3, 9).
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 the SNOOZE status output is in use, output from TRDIOC0 is stopped.
5. When a pin sharing a timer RDe function is to be used as a general-purpose port pin, the target bit for TRDIOij pin output control in the timer RDe output master enable register 1 (TRDOER1) must have the same setting as its initial value (i = A, B, C, D, j = 0, 1). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).
6. 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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR8).
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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR6).
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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).
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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR4).

Remarks

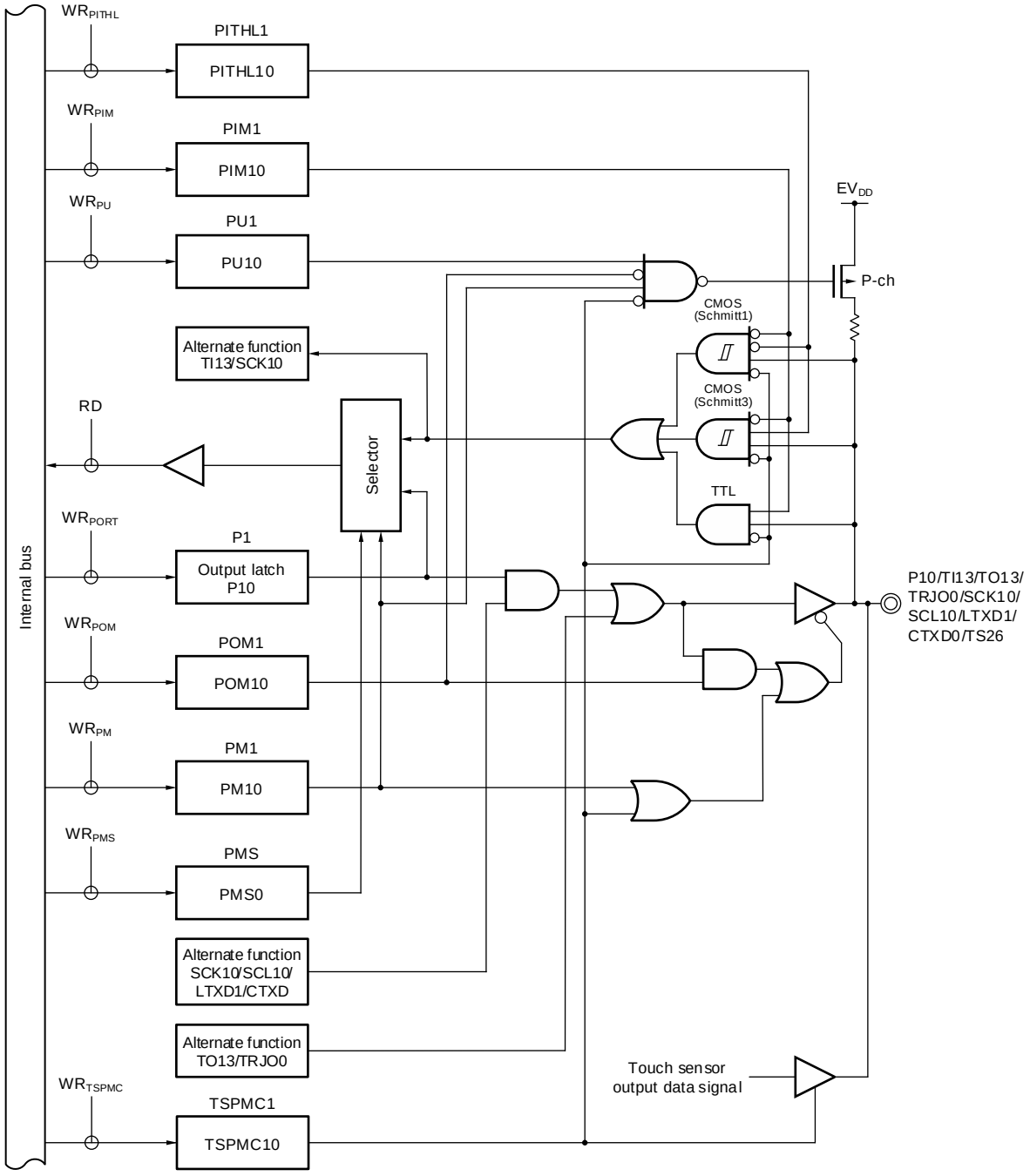
×: Don't care
PM1x: Bit of port mode register 1
PIM1x: Bit of port input mode register 1
POM1x: Bit of port output mode register 1
PITHL1x: Bit of port input threshold control register 1
TSPMC1x: Bit of touch sensor port mode control register 1

Table 4-5. P10/TS26 to P17/TS16 Pin Function Setting

TSPMC1 Register	PM1 Register	P10/TS26 to P17/TS16 Pins
Digital input/output selection	Input mode	Digital input
	Output mode	Digital output
Capacitance measurement function selection	Input mode	Capacitance measurement function
	Output mode	Setting prohibited

Figure 4-5 to **Figure 4-12** show block diagrams of port 1 for 100-pin products.

Figure 4-5. Block Diagram of P10



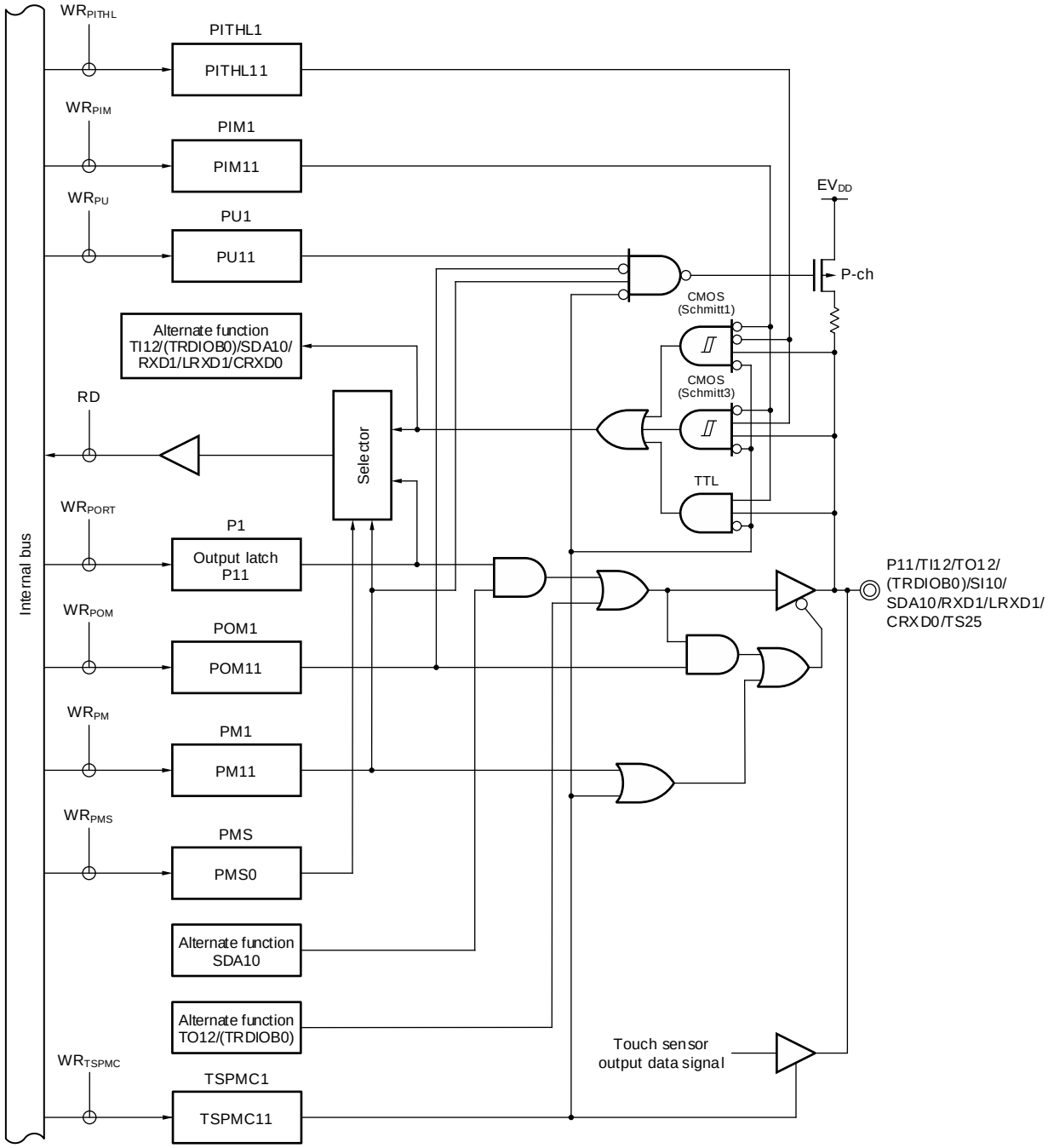
P1:	Port register 1	PITHL1:	Port input threshold control register 1
PU1:	Pull-up resistor option register 1	TSPMC1:	Touch sensor port mode control register 1
PM1:	Port mode register 1	RD:	Read signal
PIM1:	Port input mode register 1	WRxx:	Write signal
POM1:	Port output mode register 1		
PMS:	Port mode select register		

(**Cautions** are listed on the next page.)

Cautions 1. The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

2. When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.

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

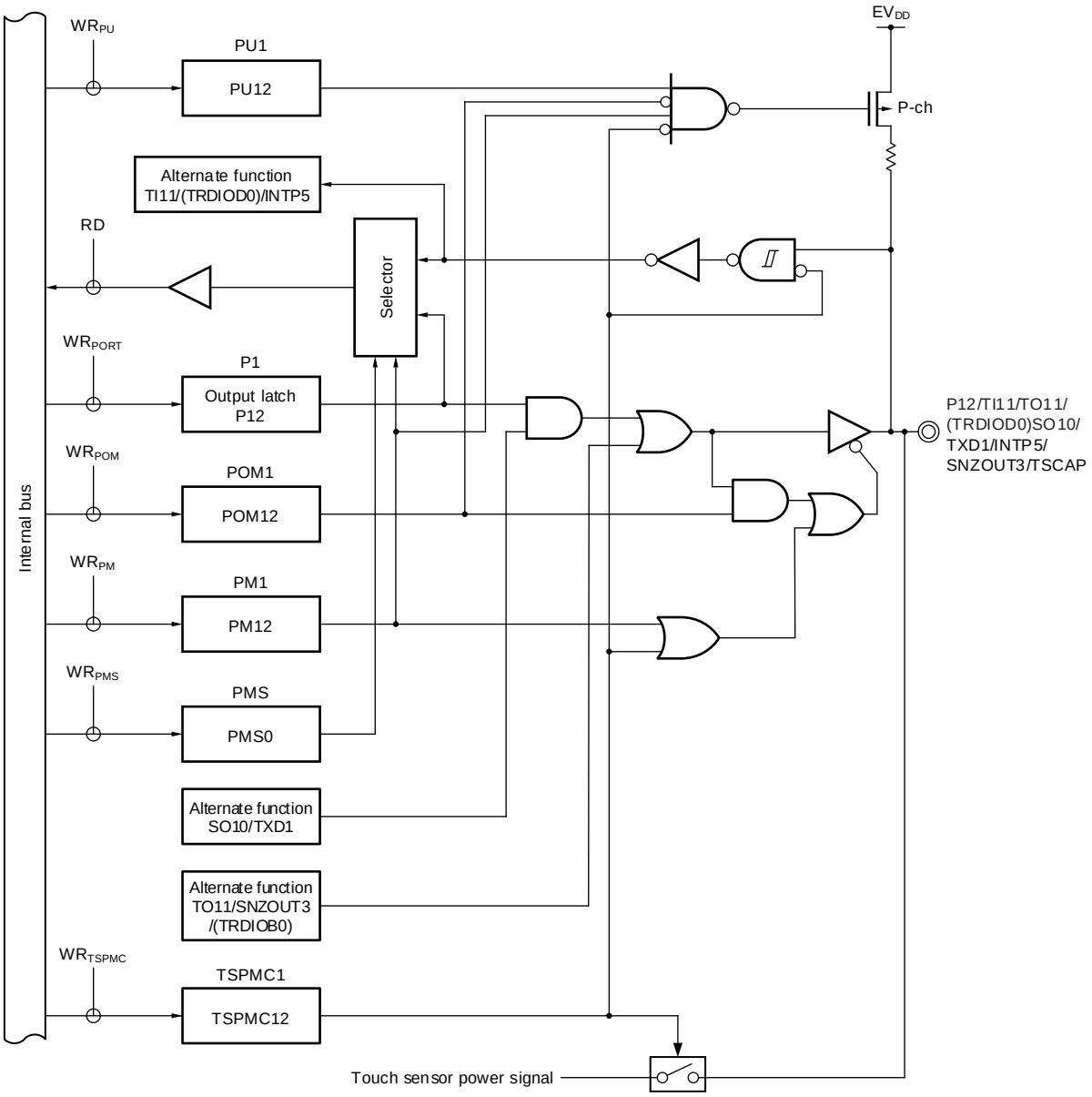
PITHL1: Port input threshold control register 1
TSPMC1: Touch sensor port mode control register 1
RD: Read signal
WRxx: Write signal

(**Cautions** are listed on the next page.)

Cautions 1. The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

2. When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.

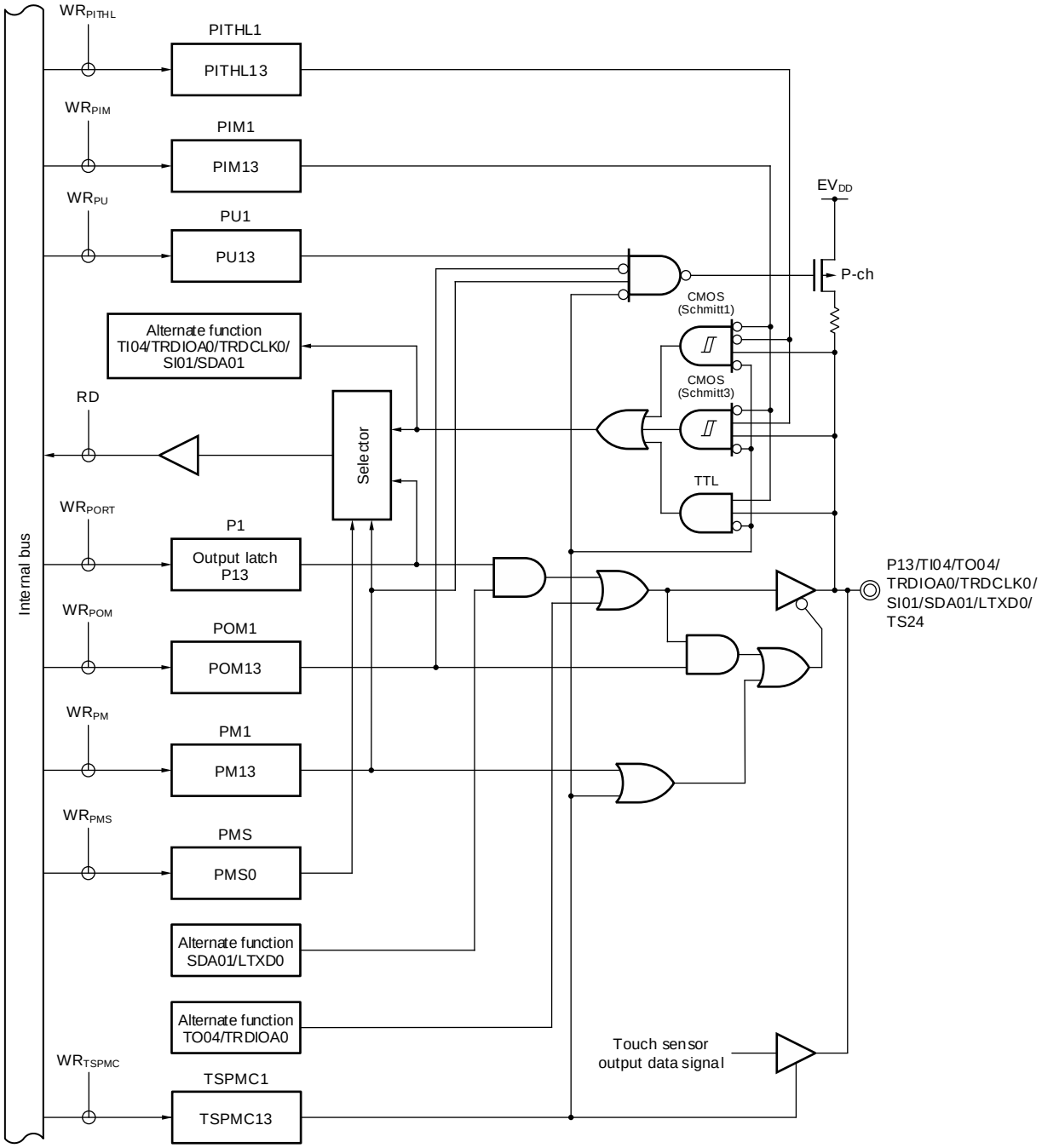
Figure 4-7. Block Diagram of P12



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
TSPMC1: Touch sensor port mode control register 1
RD: Read signal
WRxx: Write signal

Caution The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

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

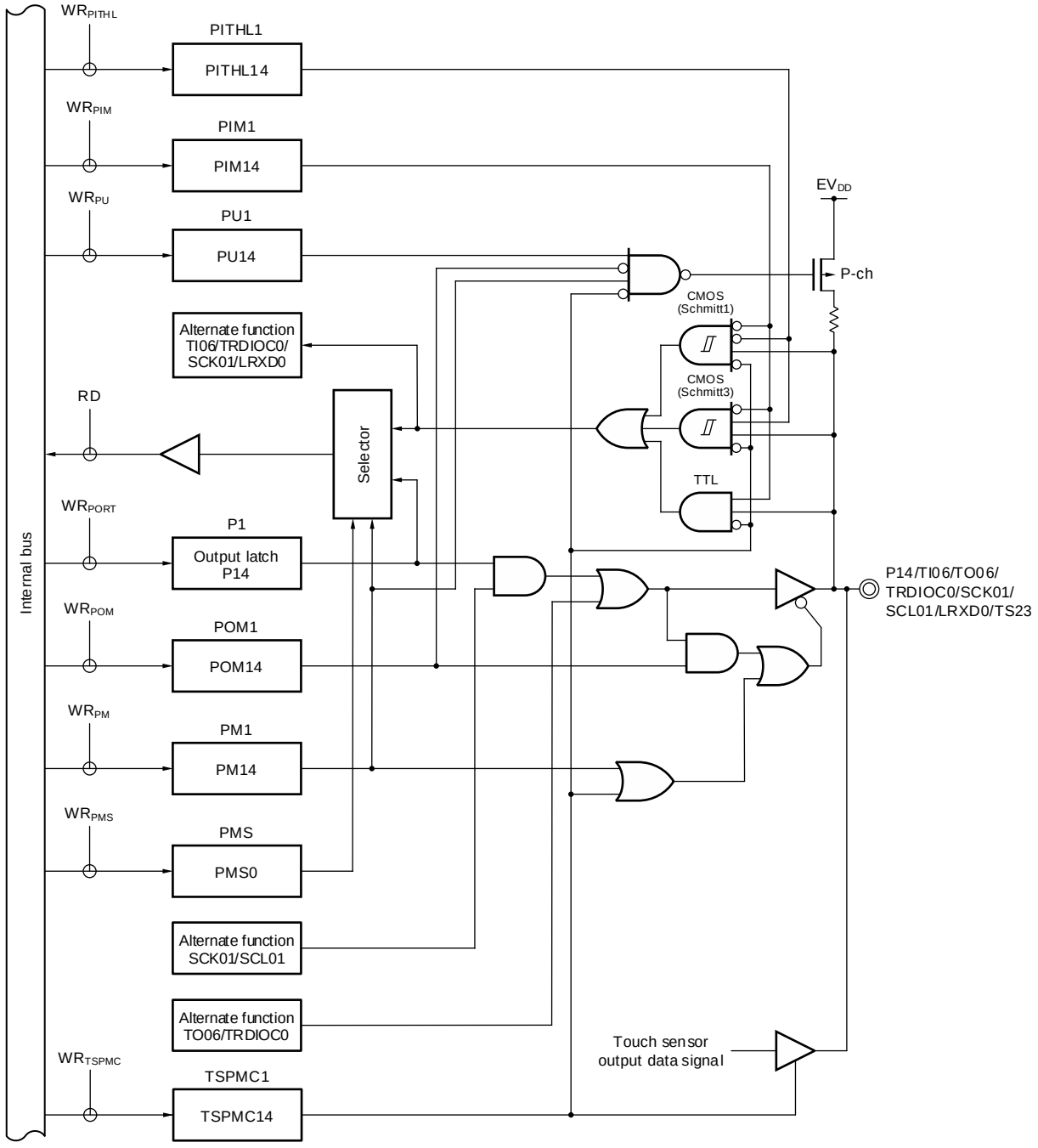
PITHL1: Port input threshold control register 1
TSPMC1: Touch sensor port mode control register 1
RD: Read signal
WRxx: Write signal

(**Cautions** are listed on the next page.)

Cautions

1. **The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.**
2. **When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.**

Figure 4-9. Block Diagram of P14



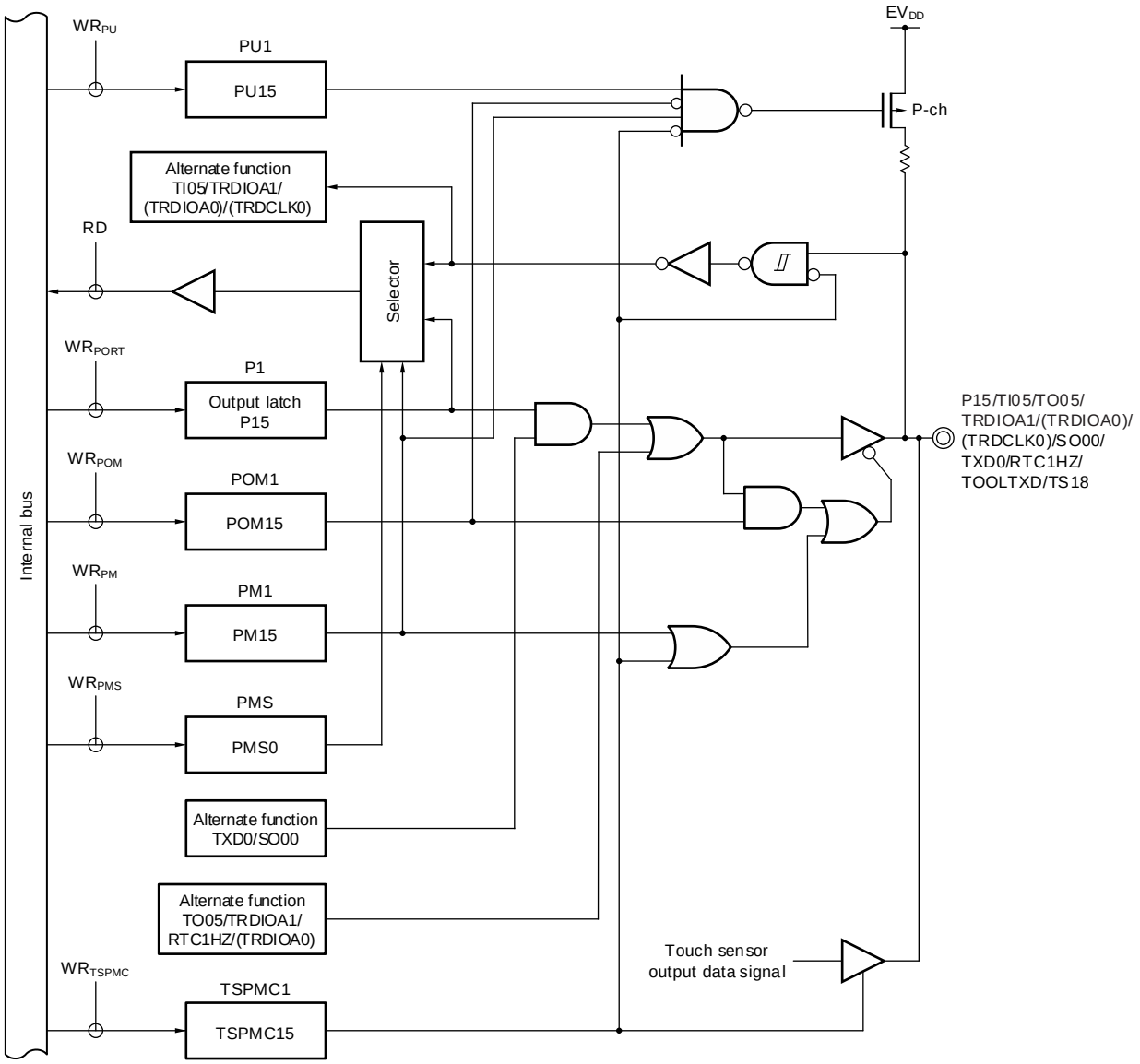
P1:	Port register 1	PITHL1:	Port input threshold control register 1
PU1:	Pull-up resistor option register 1	TSPMC1:	Touch sensor port mode control register 1
PM1:	Port mode register 1	RD:	Read signal
PIM1:	Port input mode register 1	WRxx:	Write signal
POM1:	Port output mode register 1		
PMS:	Port mode select register		

(**Cautions** are listed on the next page.)

Cautions 1. The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

2. When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.

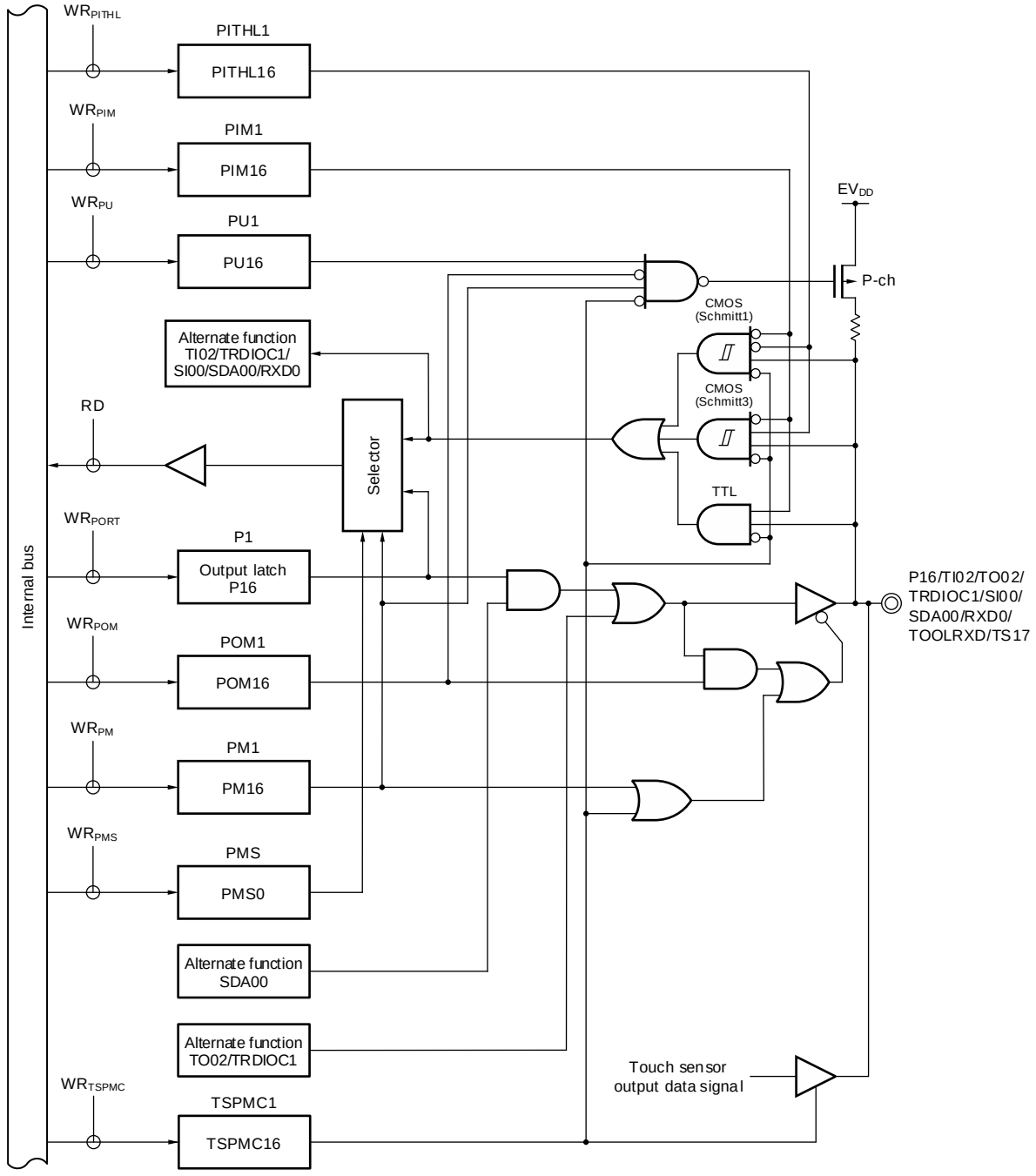
Figure 4-10. Block Diagram of P15



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
TSPMC1: Touch sensor port mode control register 1
RD: Read signal
WRxx: Write signal

Caution The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

Figure 4-11. Block Diagram of P16



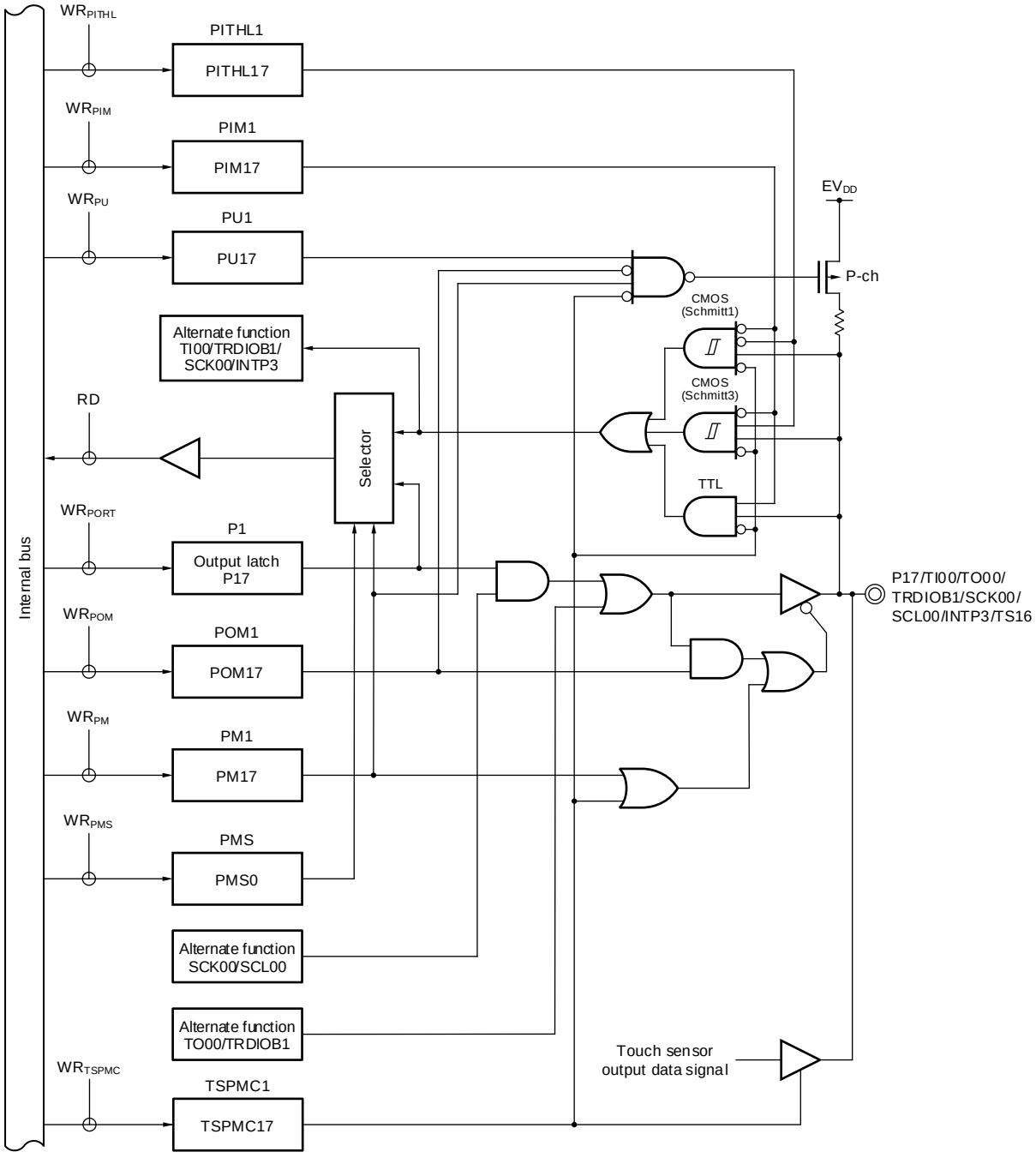
P1:	Port register 1	PITHL1:	Port input threshold control register 1
PU1:	Pull-up resistor option register 1	TSPMC1:	Touch sensor port mode control register 1
PM1:	Port mode register 1	RD:	Read signal
PIM1:	Port input mode register 1	WRxx:	Write signal
POM1:	Port output mode register 1		
PMS:	Port mode select register		

(**Cautions** are listed on the next page.)

Cautions 1. The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

2. When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.

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
TSPMC1: Touch sensor port mode control register 1
RD: Read signal
WRxx: Write signal

(**Cautions** are listed on the next page.)

Cautions 1. The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

2. When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.

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).

Output from the P32 pin can be specified as N-ch open-drain output (EV_{DD} tolerance) in 1-bit units using port output mode register 3 (POM3).

This port can also be used for external interrupt request input, timer I/O, serial interface slave select input, data output for serial interfaces (CSI), SNOOZE status output, and STOP status output.

Table 4-6. Settings of Registers When Using Port 3 (P30 to P32)

Pin name		PM3x	PIM3x	POM3x	PITHL3x	Alternate Function Setting	Remarks
Name	I/O						
P30	Input	1	0	–	0	×	CMOS input (Schmitt1 input)
				–	1		CMOS input (Schmitt3 input)
			1	–	0	×	TTL input
	Output	0	×	–	×	TRDIOD1 output = 0 Note 1 TO01 output = 0 Note 2 SNZOUT0 output = 0 Note 3	
P31	Input	1	–	–	–	×	
	Output	0	–	–	–	TO14 output = 0 Note 2 STOPST output = 0 Note 4	
P32	Input	1	–	×	–	×	
	Output	0	–	0	–	TO16 output = 0 Note 2 (SO11 output = 1) Note 5	CMOS output
				1			N-ch O.D output

Important To use the port 3 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

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

Notes 1. When a pin sharing a timer RDe function is to be used as a general-purpose port pin, the target bit for TRDIOij pin output control in the timer RDe output master enable register 1 (TRDOER1) must have the same setting as its initial value (i = A, B, C, D, j = 0, 1). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).

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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR1, 3, 9).

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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR6).

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. Alternatively, assign the corresponding function to another pin by STPSEL bit in the STPSTC register.

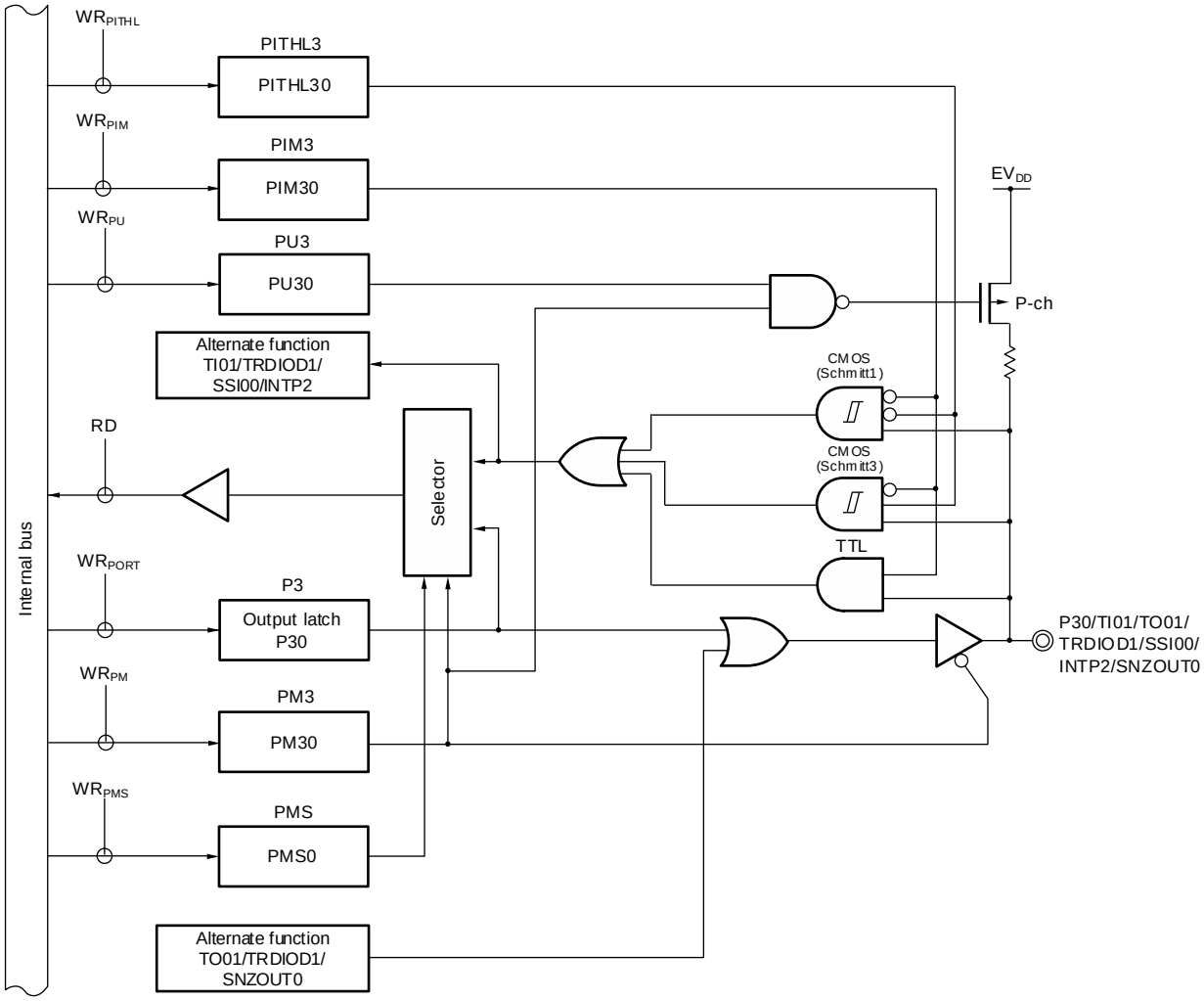
5. When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn and CKOmn bits of the serial output register m (SOm), the SOEmn bit of the serial output enable register m (SOEm), and the SEmn bit of the serial channel enable status register m (SEm) corresponding to the target unit and channel must have the same setting as its initial value (m = 0, 1, n = 0, 1). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).

Remarks

×:	Don't care
PM3x:	Bit of port mode register 3
PIM3x:	Bit of port input mode register 3
POM3x:	Bit of port output mode register 3
PITHL3x:	Bit of port input threshold control register 3

Figure 4-13 to **Figure 4-15** show block diagrams of port 3 for 100-pin products.

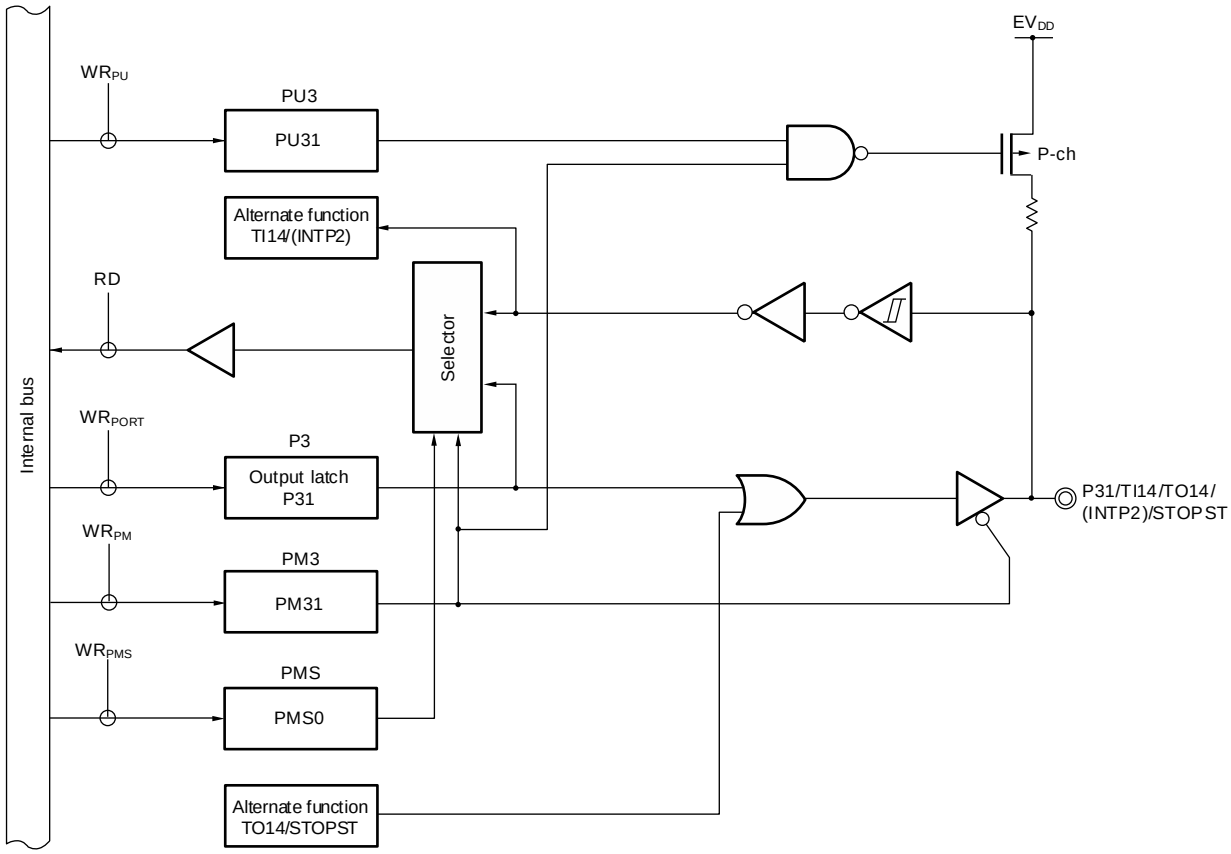
Figure 4-13. Block Diagram of P30



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

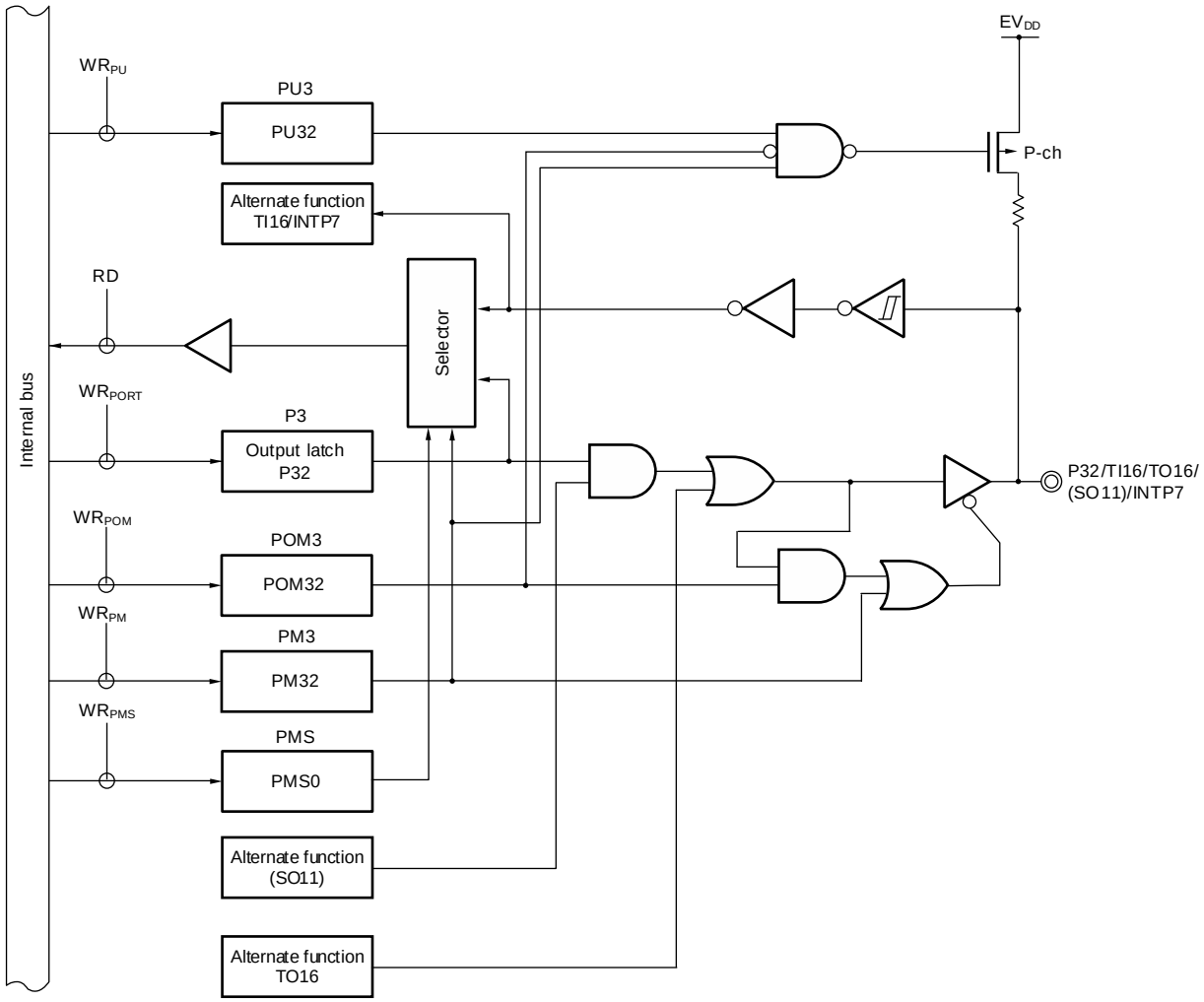
Caution When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.

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
POM3: Port output mode register 3
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

Caution The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

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 P41 and 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, serial interface data input, timer RDe counter clear trigger input, and data I/O for a flash memory programmer/debugger.

Reset signal generation sets this port to input mode.

Table 4-7. Settings of Registers When Using Port 4

Pin name		PM4x	PITHL4x	Alternate Function Setting	Remarks
Name	I/O				
P40	Input	1	–	×	
	Output	0	–	×	
P41	Input	1	0	×	CMOS input (Schmitt1 input)
			1		CMOS input (Schmitt3 input)
	Output	0	×	TRJIO0 output = 0 Note 1 TO10 output = 0 Note 2 VCOUT0 output = 0 Note 3 SNZOUT2 output = 0 Note 4	
P42	Input	1	–	×	
	Output	0	–	(LTXD0 output = 1) Note 5	
P43	Input	1	0	×	CMOS input (Schmitt1 input)
			1	×	CMOS input (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	–	×	

Important **To use the port 4 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.**

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

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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR1, 3, 9).

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.

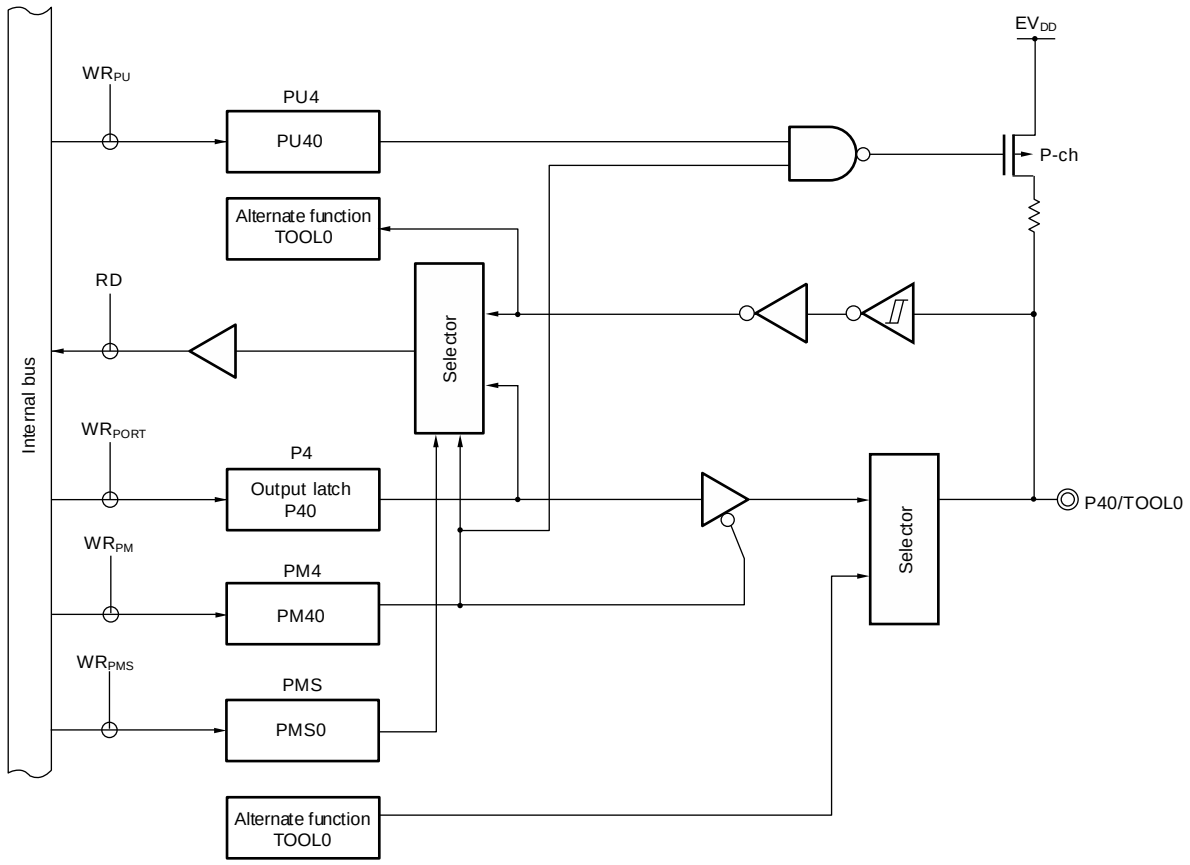
4. When a pin sharing the SNOOZE status output function is to be used as a general-purpose port pin, the OUTEN0 to OUTEN7 bits of the SNOOZE status output control registers 0, 1, 2, 3 (PSNZCNT0, 1, 2, 3) must have the same setting as its initial value. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR6).

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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR4).

Remarks ×: Don't care
PM4x: Bit of port mode register 4
PITHL4x: Bit of port input threshold control register 4

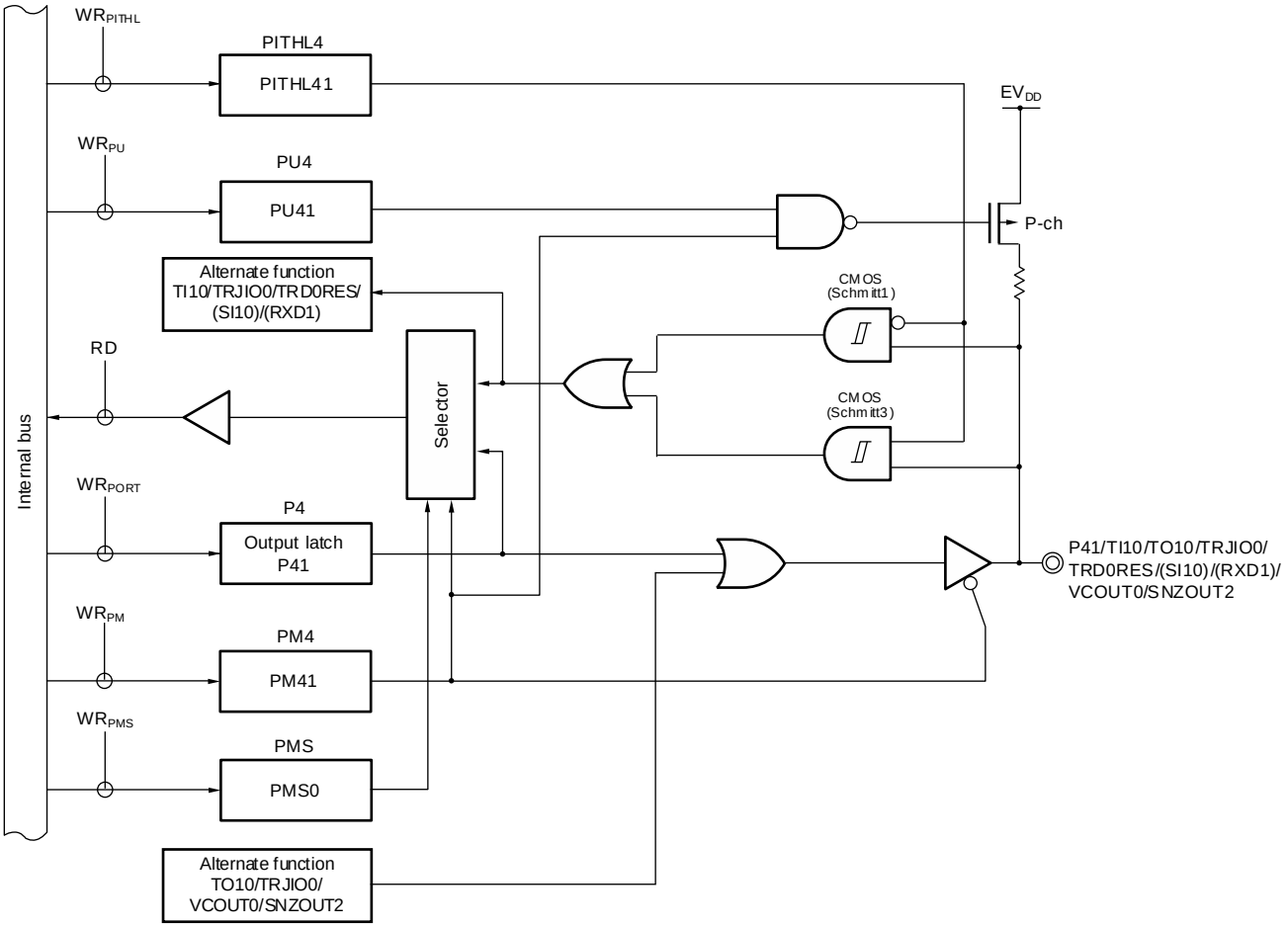
Figure 4-16 to **Figure 4-23** show block diagrams of port 4 for 100-pin products.

Figure 4-16. Block Diagram of P40



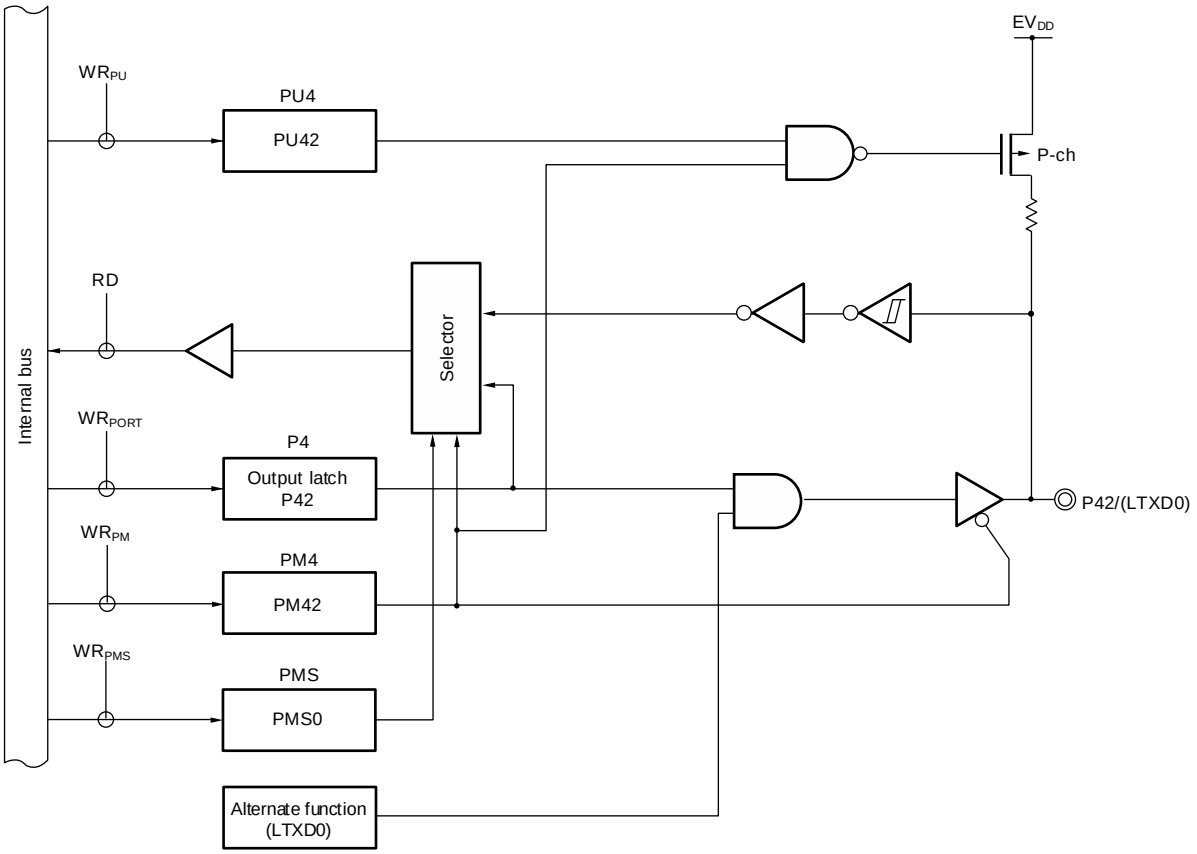
P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

Figure 4-17. Block Diagram of P41



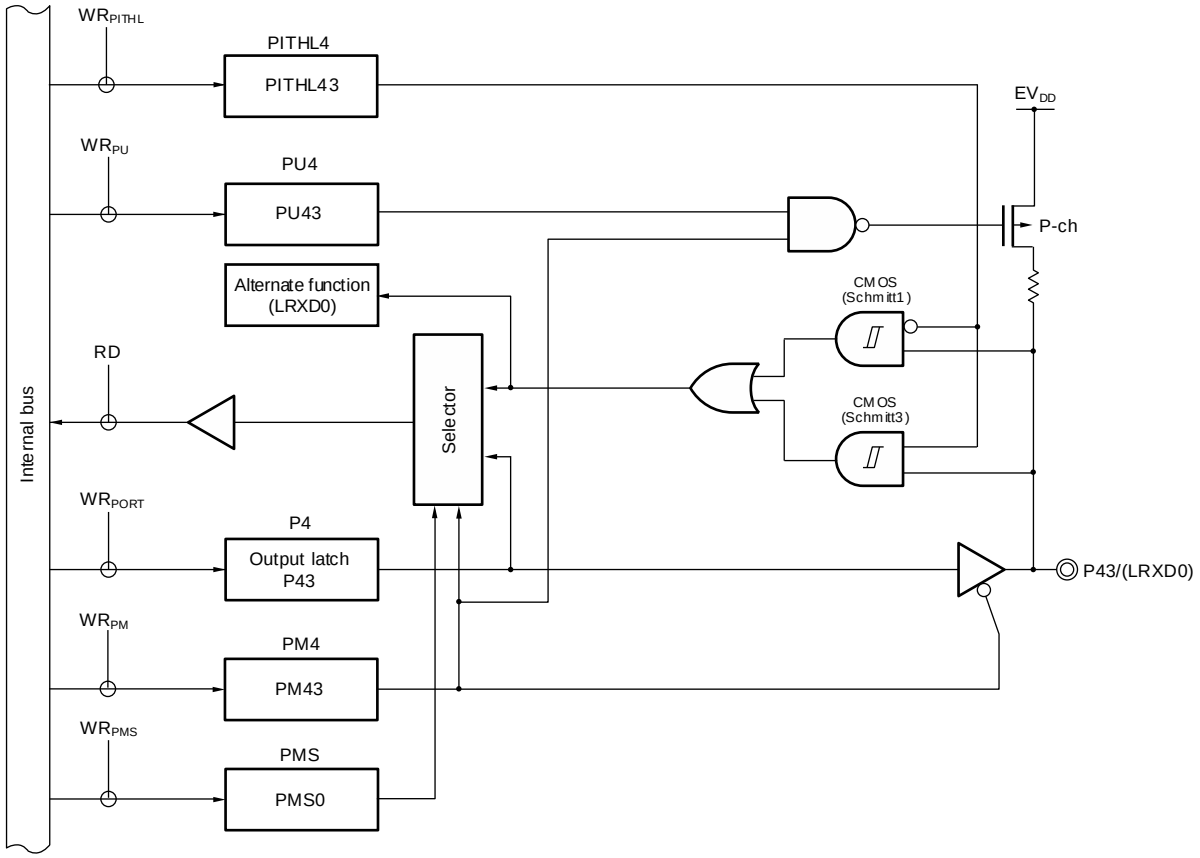
P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
PITHL4: Port input threshold control register 4
RD: Read signal
WRxx: Write signal

Figure 4-18. Block Diagram of P42



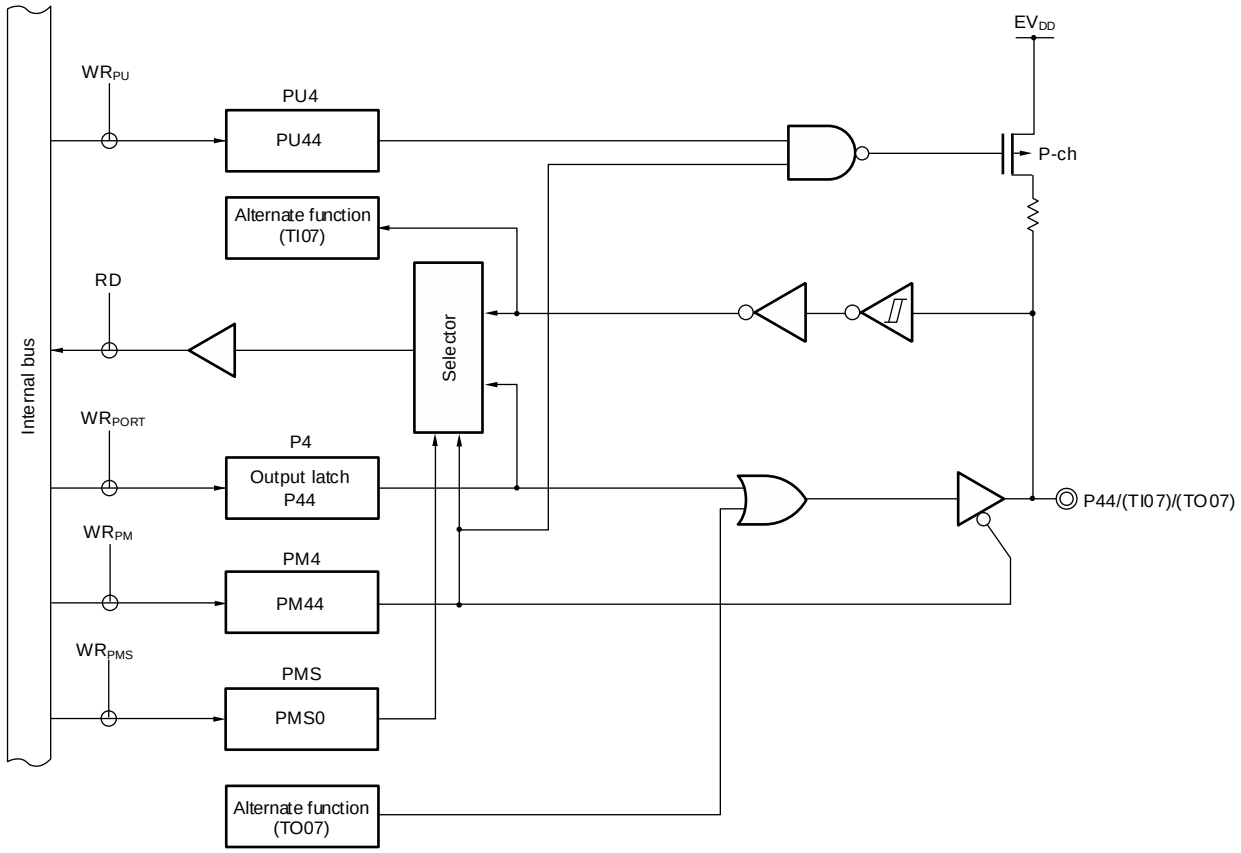
P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

Figure 4-19. Block Diagram of P43



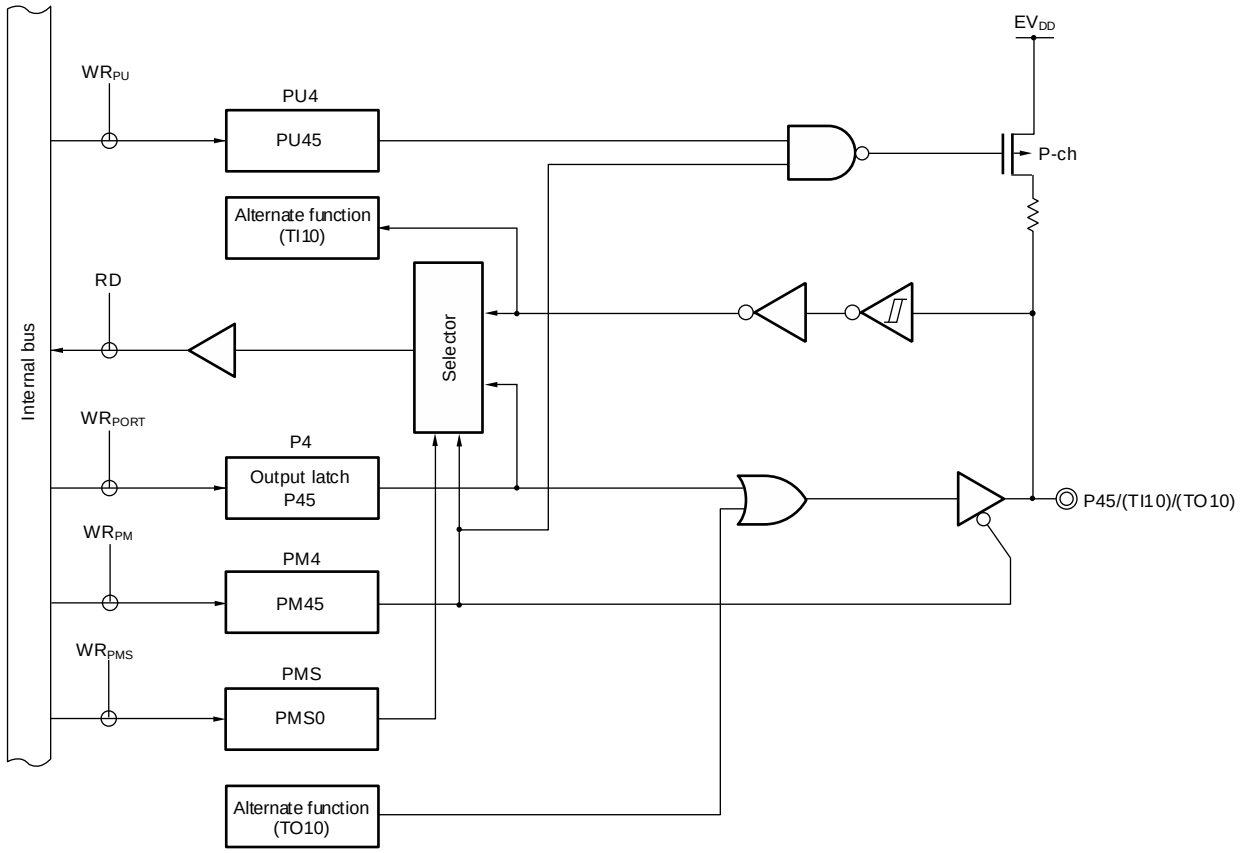
P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
PITHL4: Port input threshold control register 4
RD: Read signal
WRxx: Write signal

Figure 4-20. Block Diagram of P44



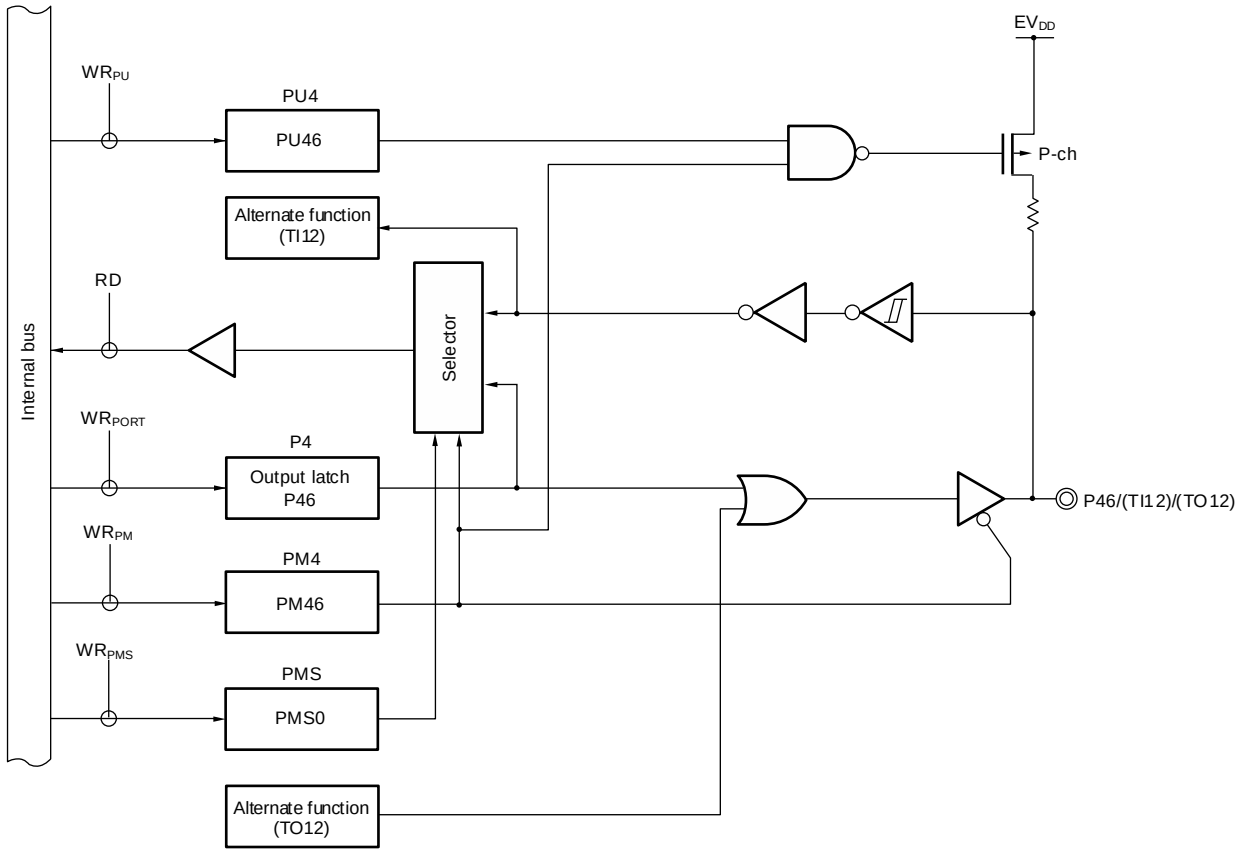
P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

Figure 4-21. Block Diagram of P45



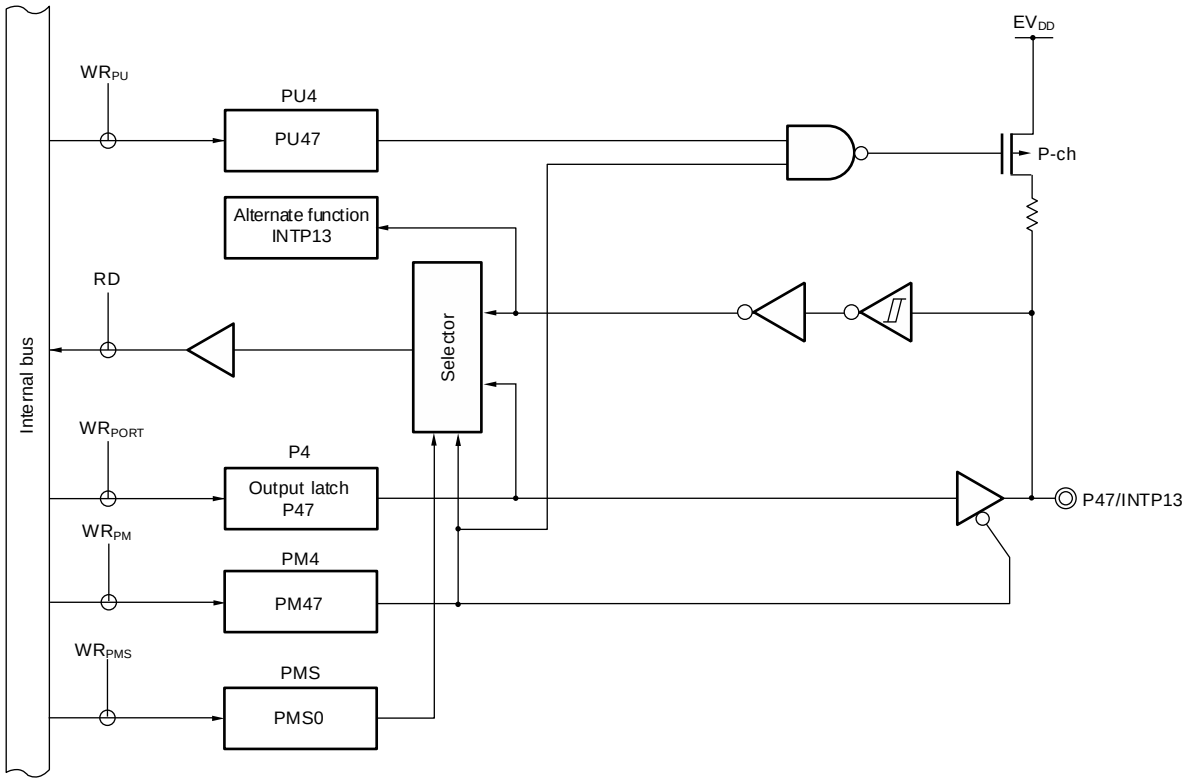
P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

Figure 4-22. 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 4-23. 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, SNOOZE status output, and capacitance measurement.

Reset signal generation sets this port to input mode.

Table 4-8. Settings of Registers When Using Port 5 (1/2)

Pin Name		PM5x	PIM5x	TSPMC5x	PITHL5x	Alternate Function Setting	Remarks
Name	I/O						
P50	Input	1	–	0	0	×	CMOS input (Schmitt1 input)
					1		CMOS input (Schmitt3 input)
	Output	0	–	0	×	×	
P51	Input	1	–	0	–	×	
	Output	0	–	0	–	(SO01 output = 1) Note 1	
P52	Input	1	–	0	0	×	CMOS input (Schmitt1 input)
					1		CMOS input (Schmitt3 input)
	Output	0	–	0	×	(SCK01 output = 1) Note 1 (STOPST output = 0) Note 2	
P53	Input	1	–	0	0	×	CMOS input (Schmitt1 input)
					1		CMOS input (Schmitt3 input)
	Output	0	–	0	×	×	
P54	Input	1	0	0	0	×	CMOS input (Schmitt1 input)
					1		CMOS input (Schmitt3 input)
			1	0	0	×	TTL input
	Output	0	×	0	×	(TO11 output = 0) Note 3	

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

Table 4-8. Settings of Registers When Using Port 5 (2/2)

Pin Name		PM5x	PIM5x	TSPMC5x	PITHL5x	Alternate Function Setting	Remarks
Name	I/O						
P55	Input	1	–	0	–	×	
	Output	0	–	0	–	(TO13 output = 0) Note 3	
P56	Input	1	–	0	–	×	
	Output	0	–	0	–	(TO15 output = 0) Note 3 (SNZOUT1 output = 0) Note 4	
P57	Input	1	–	0	–	×	
	Output	0	–	0	–	(TO17 output = 0) Note 3 (SNZOUT0 output = 0) Note 4	

Important To use the port 5 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

Notes 1. When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn and CKOmn bits 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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).

2. 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. Alternatively, assign the corresponding function to another pin by STPSEL bit in the STPSTC register.

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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR1, 3, 9).

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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR6).

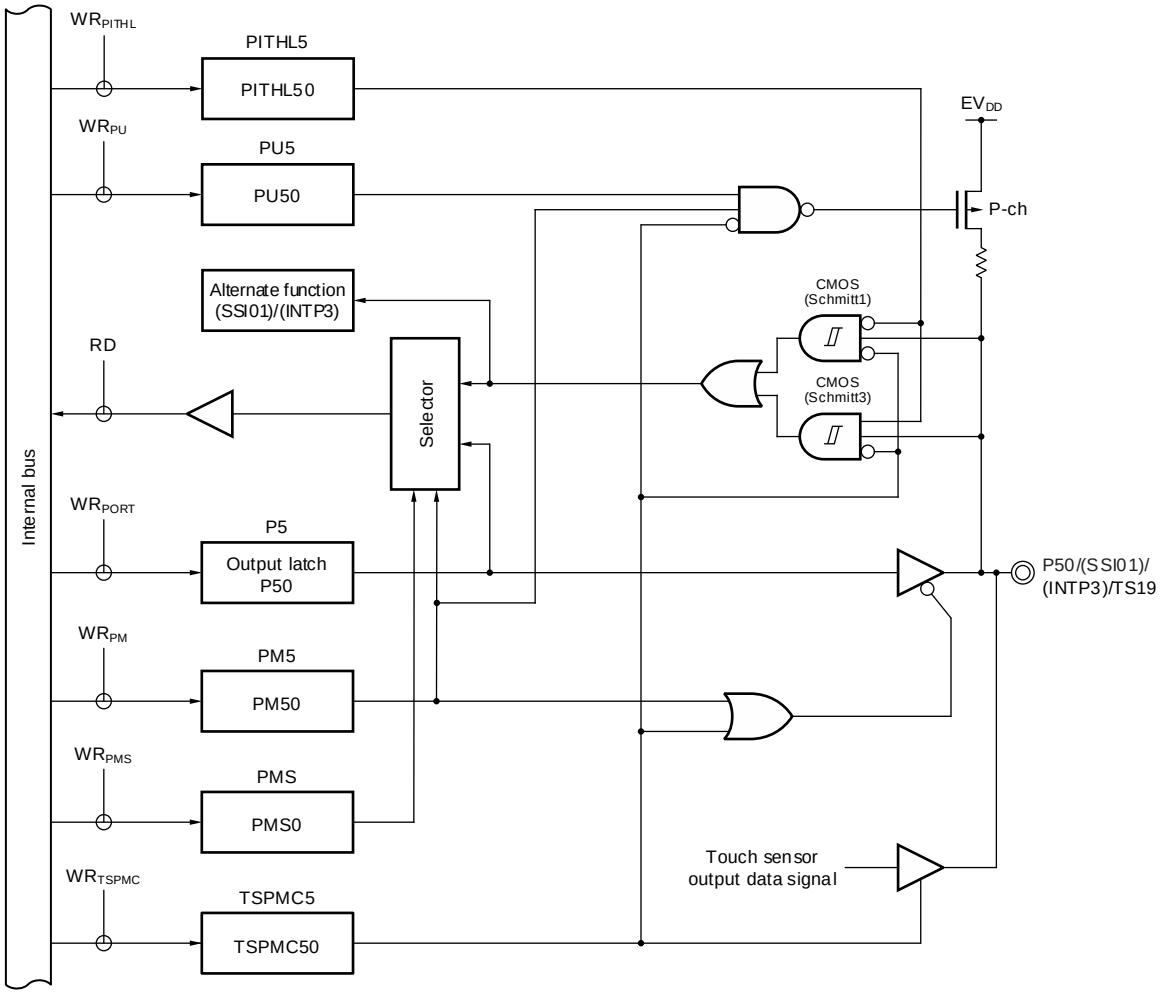
Remarks ×: Don't care
PM5x: Bit of port mode register 5
PIM5x: Bit of port input mode register 5
PITHL5x: Bit of port input threshold control register 5
TSPMC5x: Bit of touch sensor port mode control register 5

Table 4-9. P50/TS19 to P57/TS30 Pin Function Setting

TSPMC5 Register	PM5 Register	P50/TS19 to P57/TS30 Pins
Digital input/output selection	Input mode	Digital input
	Output mode	Digital output
Capacitance measurement function selection	Input mode	Capacitance measurement function
	Output mode	Setting prohibited

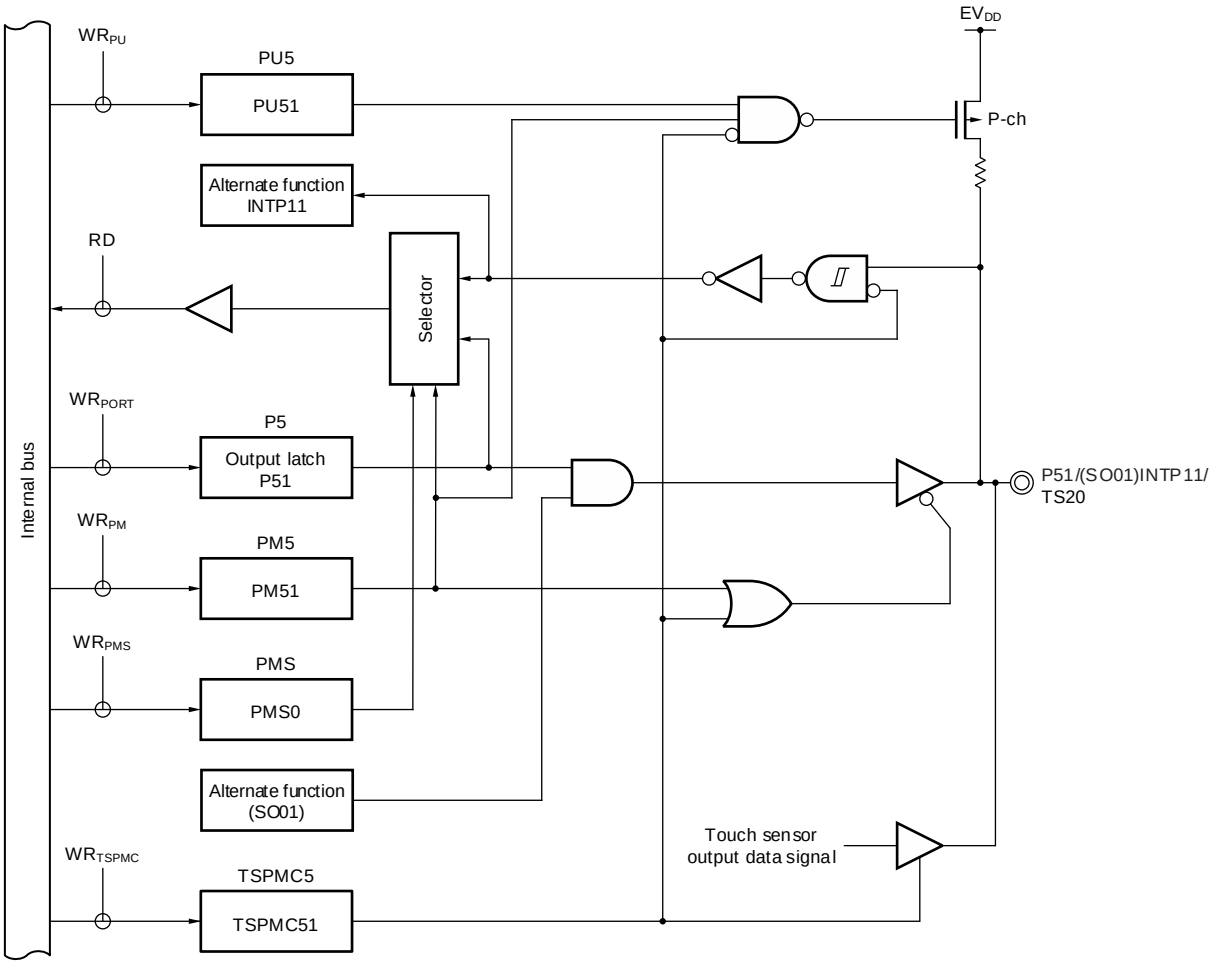
Figure 4-24 to **Figure 4-31** show block diagrams of port 5 for 100-pin products.

Figure 4-24. Block Diagram of P50



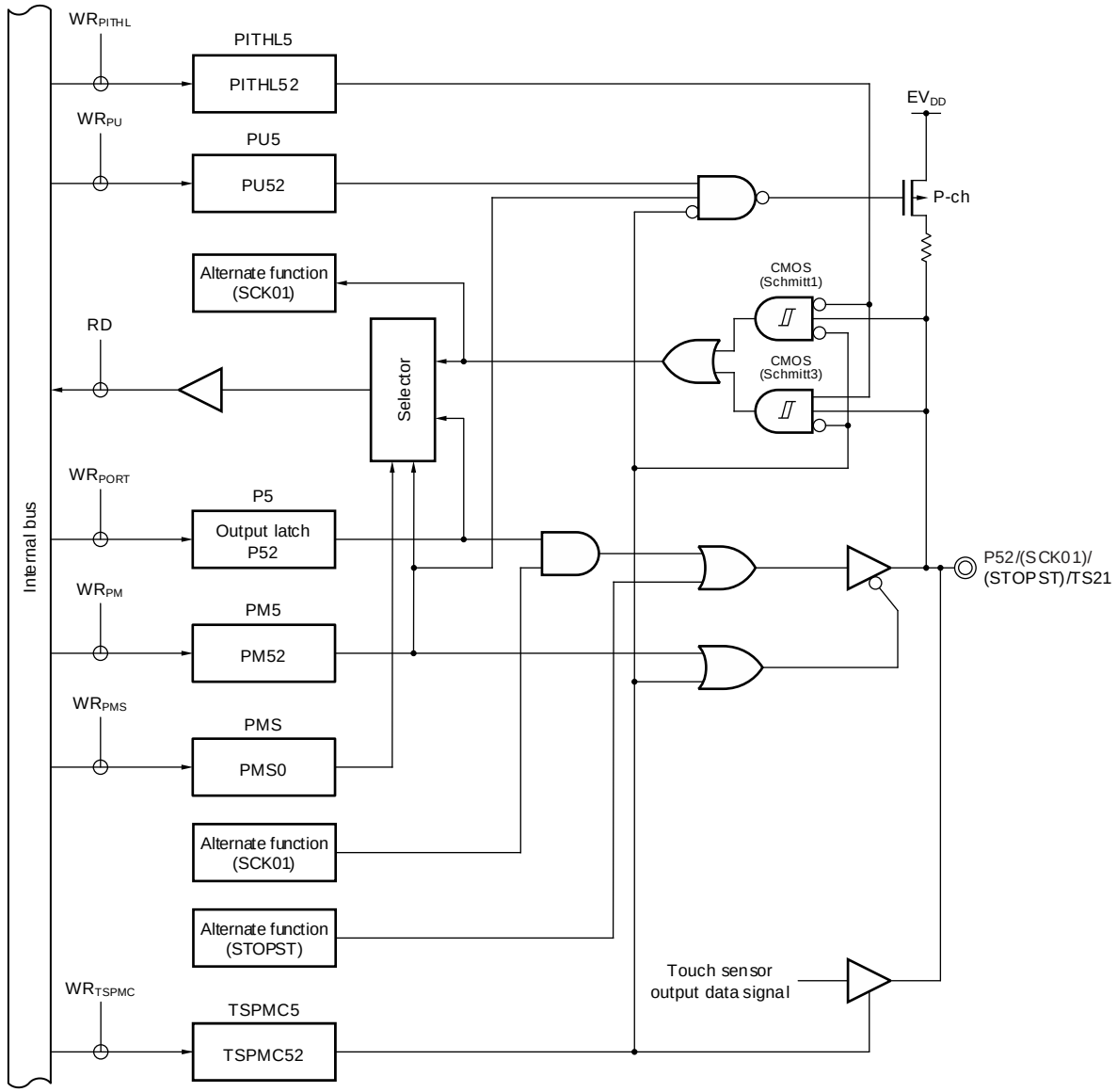
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PMS: Port mode select register
PITHL5: Port input threshold control register 5
TSPMC5: Touch sensor port mode control register 5
RD: Read signal
WRxx: Write signal

Figure 4-25. Block Diagram of P51



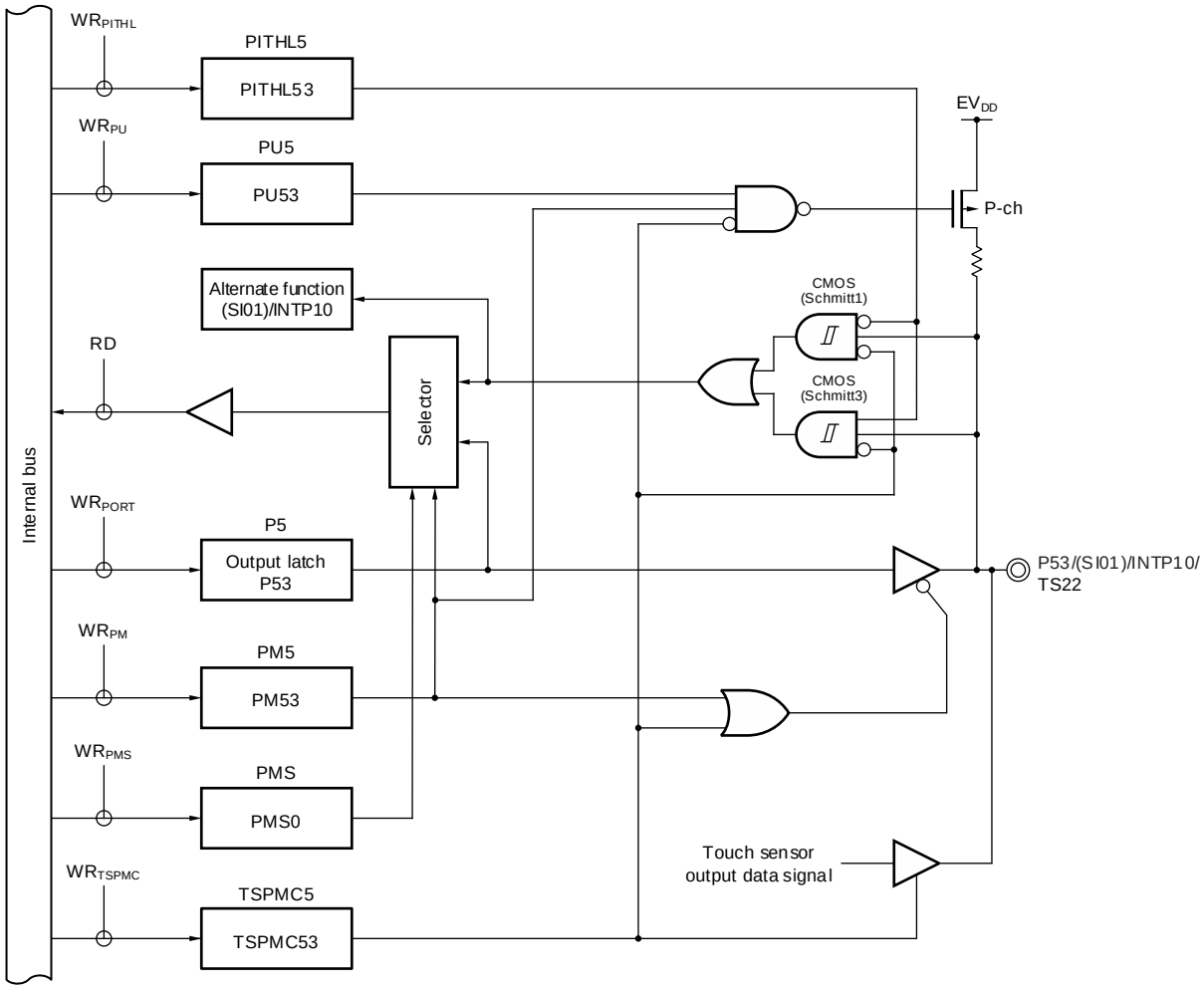
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PMS: Port mode select register
TSPMC5: Touch sensor port mode control register 5
RD: Read signal
WRxx: Write signal

Figure 4-26. Block Diagram of P52



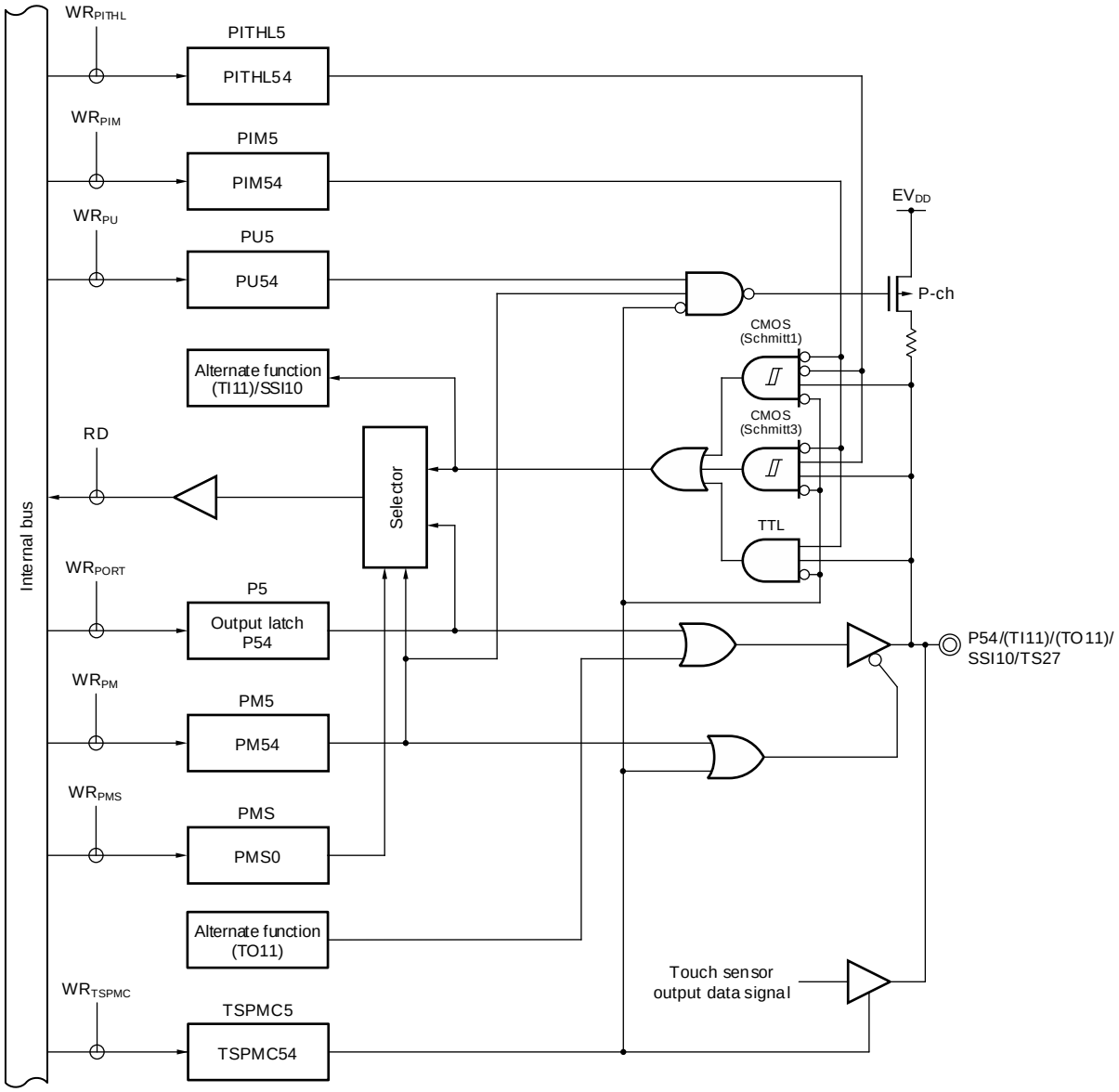
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PMS: Port mode select register
PITHL5: Port input threshold control register 5
TSPMC5: Touch sensor port mode control register 5
RD: Read signal
WRxx: Write signal

Figure 4-27. 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
TSPMC5: Touch sensor port mode control register 5
RD: Read signal
WRxx: Write signal

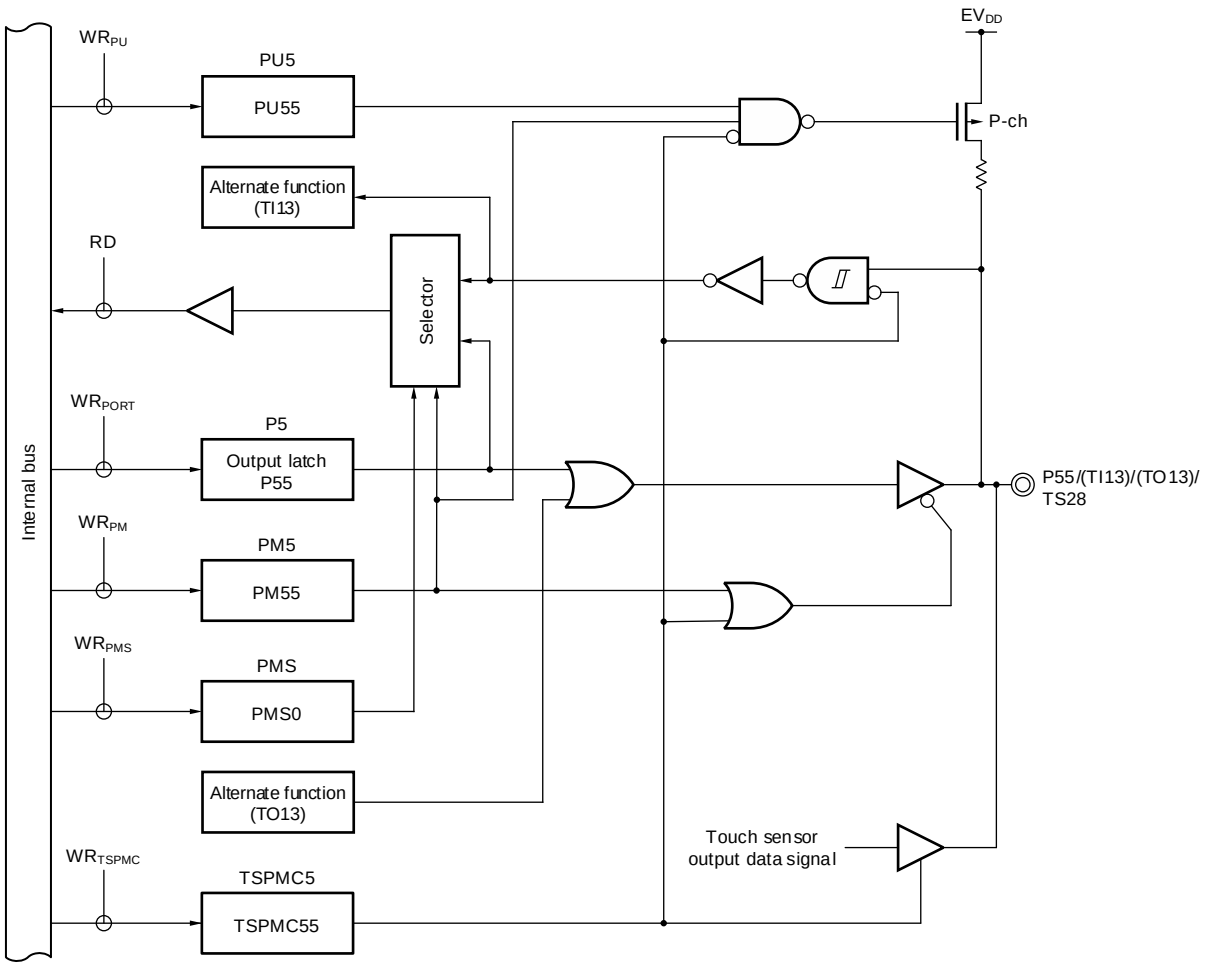
Figure 4-28. Block Diagram of P54



P5:	Port register 5	PITHL5:	Port input threshold control register 5
PU5:	Pull-up resistor option register 5	TSPMC5:	Touch sensor port mode control register 5
PM5:	Port mode register 5	RD:	Read signal
PIM5:	Port input mode register 5	WRxx:	Write signal
PMS:	Port mode select register		

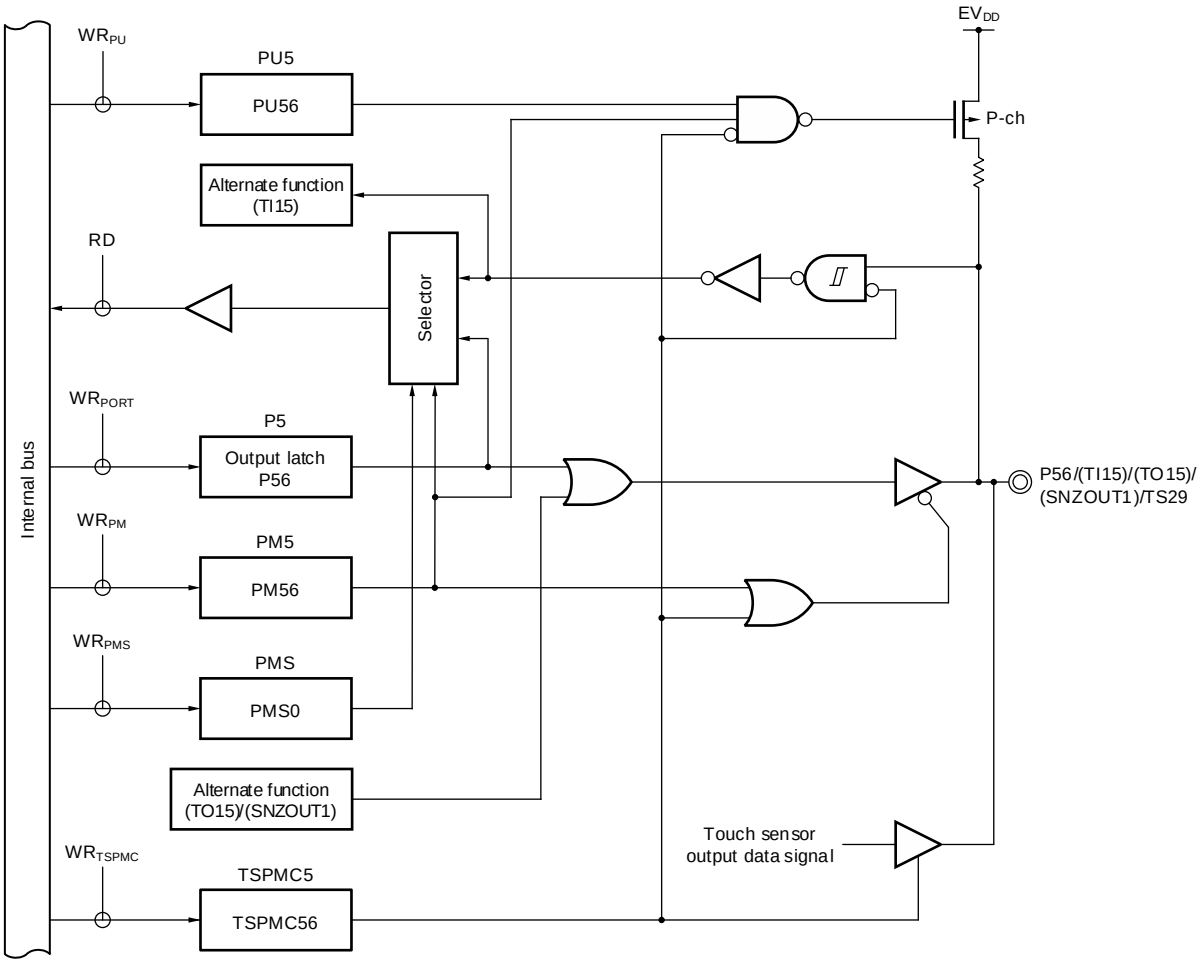
Caution When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.

Figure 4-29. Block Diagram of P55



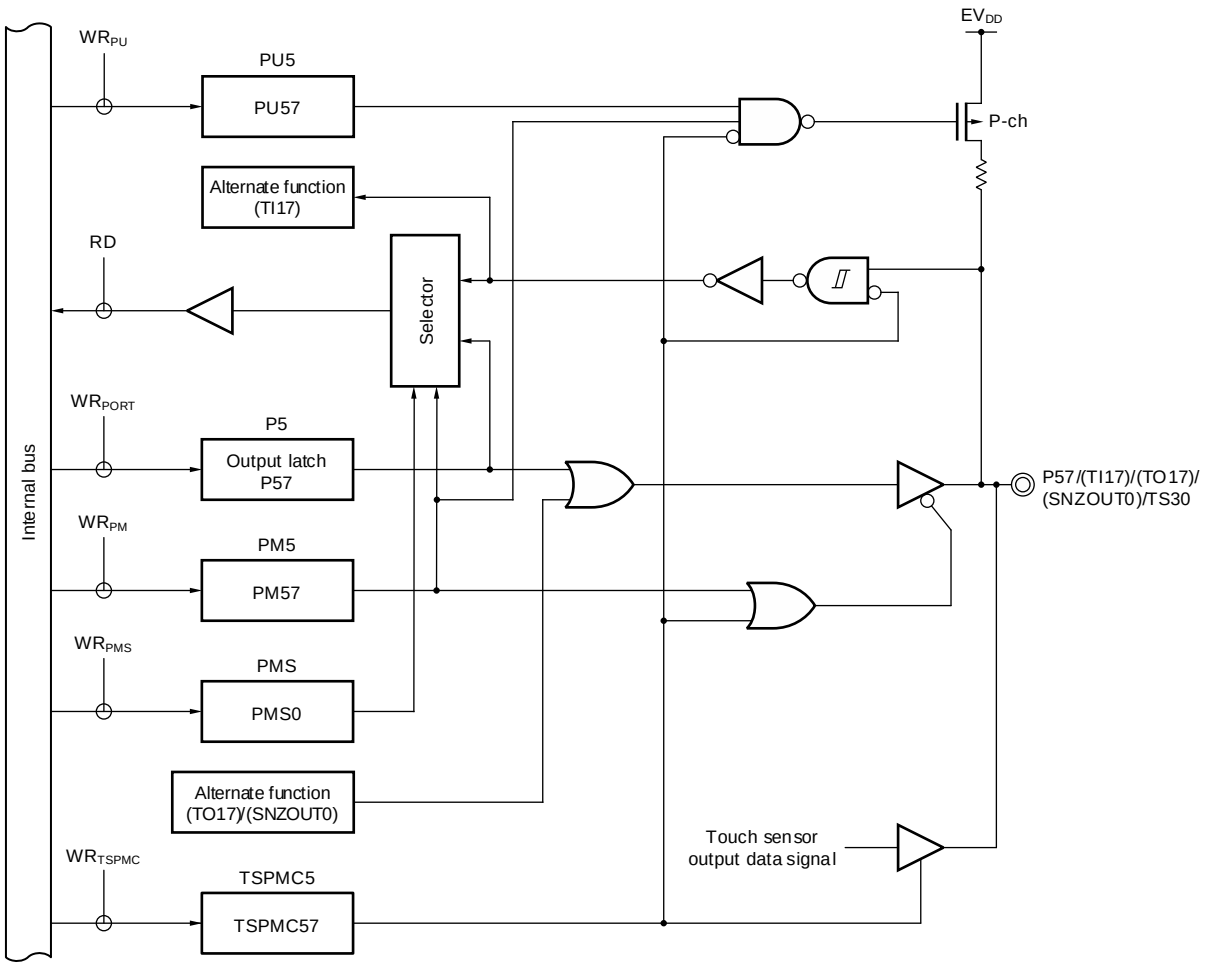
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PMS: Port mode select register
TSPMC5: Touch sensor port mode control register 5
RD: Read signal
WRxx: Write signal

Figure 4-30. 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
TSPMC5: Touch sensor port mode control register 5
RD: Read signal
WRxx: Write signal

Figure 4-31. 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
TSPMC5: Touch sensor port mode control register 5
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, CAN serial data I/O, slave select input, timer I/O, SNOOZE status output, and capacitance measurement.

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	TSPMC6x	PITHL6x	Alternate Function Setting	Remarks
Name	I/O							
P60	Input	1	–	×	0	0	×	CMOS input (Schmitt1 input)
						1		CMOS input (Schmitt3 input)
	Output	0	–	0	0	×	(SCK00/SCL00 output = 1) Note 1	CMOS output
				1	0	×	(TO01 output = 0) Note 2	N-ch O.D output
P61	Input	1	–	×	0	0	×	CMOS input (Schmitt1 input)
						1		CMOS input (Schmitt3 input)
	Output	0	–	0	0	×	(SDA00 output = 1) Note 1 (TO02 output = 0) Note 2 CTXD1 output = 1 Note 5	CMOS output
				1	0	×		N-ch O.D output
P62	Input	1	0	×	0	0	×	CMOS input (Schmitt1 input)
						1		CMOS input (Schmitt3 input)
			1	×	0	0	×	TTL input
	Output	0	×	0	0	×	SCLA0 output = 0 Note 3 (SO00/TXD0 output = 1) Note 1 (TO03 output = 0) Note 2	CMOS output
				1	0	×		N-ch O.D output

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

Table 4-10. Settings of Registers When Using Port 6 (2/2)

Pin Name		PM6x	PIM6x	POM6x	TSPMC6x	PITHL6x	Alternate Function Setting	Remarks
Name	I/O							
P63	Input	1	0	×	0	0	×	CMOS input (Schmitt1 input)
						1		CMOS input (Schmitt3 input)
			1	×	0	0	×	TTL input
	Output	0	×	0	0	×	SDAA0 output = 0 Note 3 (TO07 output = 0) Note 2	CMOS output
				1	0	×		N-ch O.D output
P64	Input	1	–	–	0	–	×	
	Output	0	–	–	0	–	(TO14 output = 0) Note 2 (SNZOUT3 output = 0) Note 4	
P65	Input	1	–	–	0	–	×	
	Output	0	–	–	0	–	(TO16 output = 0) Note 2 (SNZOUT2 output = 0) Note 4	
P66	Input	1	–	–	0	–	×	
	Output	0	–	–	0	–	(TO00 output = 0) Note 2	
P67	Input	1	–	–	0	–	×	
	Output	0	–	–	0	–	(TO02 output = 0) Note 2	

Important To use the port 6 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

Notes 1. When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn and CKOmn bits 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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).

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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR1, 3, 9).

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 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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR6).

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.

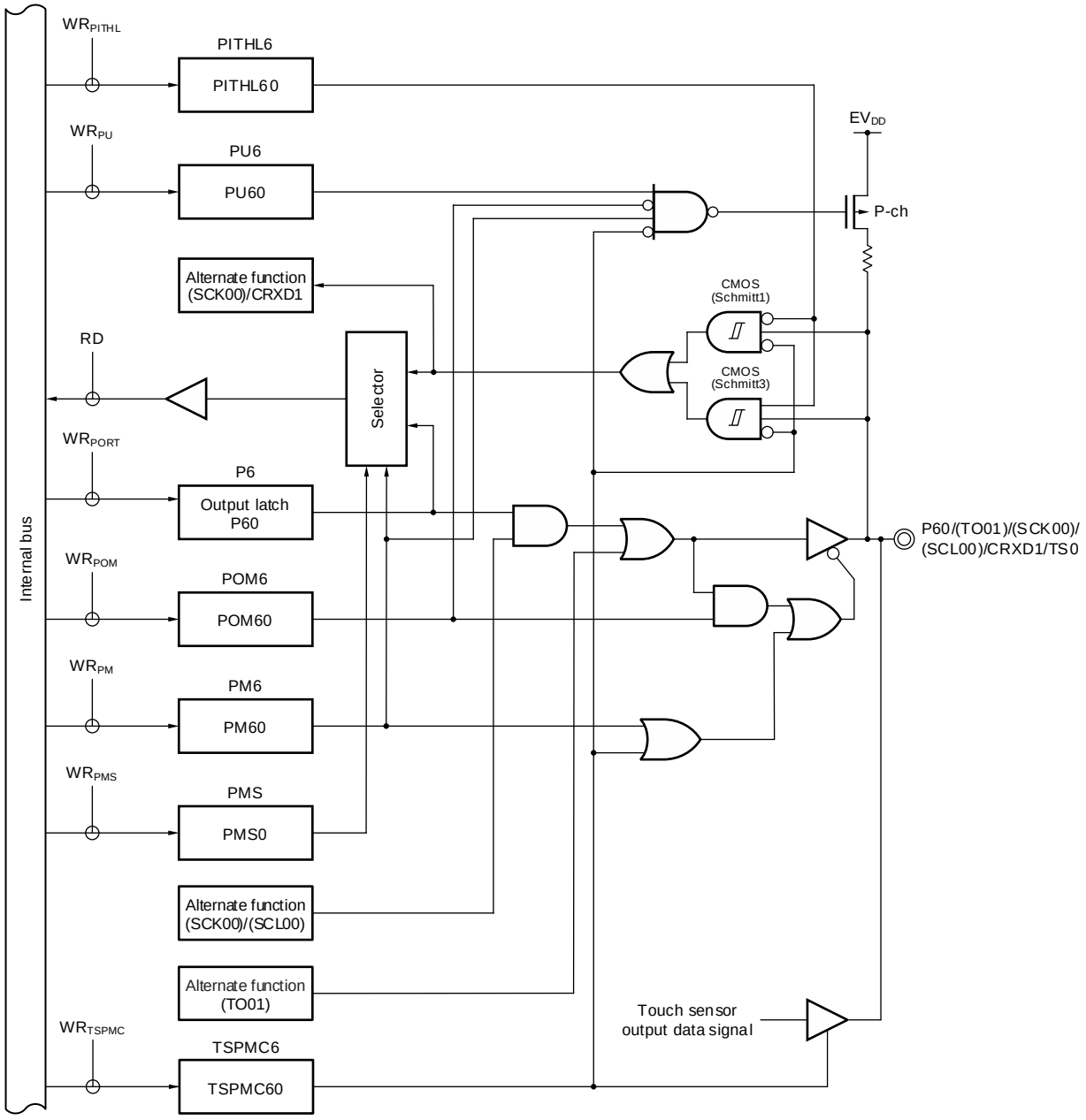
Remarks ×: Don't care
PM6x: Bit of port mode register 6
PIM6x: Bit of port input mode register 6
POM6x: Bit of port output mode register 6
PITHL6x: Bit of port input threshold control register 6
TSPMC6x: Bit of touch sensor port mode control register 6

Table 4-11. P60/TS0 to P67/TS7 Pin Function Setting

TSPMC6 Register	PM6 Register	P60/TS0 to P67/TS7 Pins
Digital input/output selection	Input mode	Digital input
	Output mode	Digital output
Capacitance measurement function selection	Input mode	Capacitance measurement function
	Output mode	Setting prohibited

Figure 4-32 to **Figure 4-39** show block diagrams of port 6 for 100-pin products.

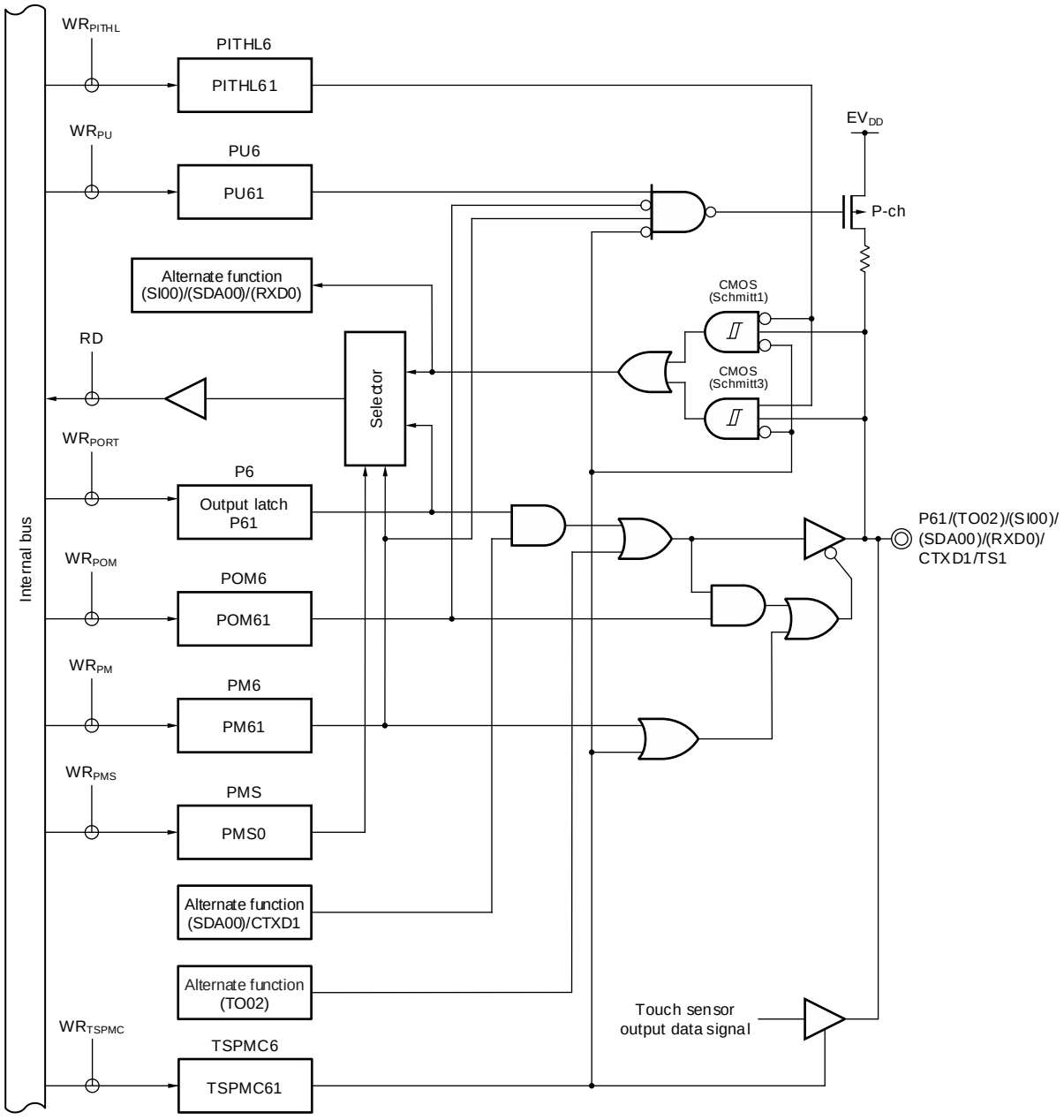
Figure 4-32. Block Diagram of P60



P6:	Port register 6	PITHL6:	Port input threshold control register 6
PU6:	Pull-up resistor option register 6	TSPMC6:	Touch sensor port mode control register 6
PM6:	Port mode register 6	RD:	Read signal
POM6:	Port output mode register 6	WRxx:	Write signal
PMS:	Port mode select register		

Caution The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

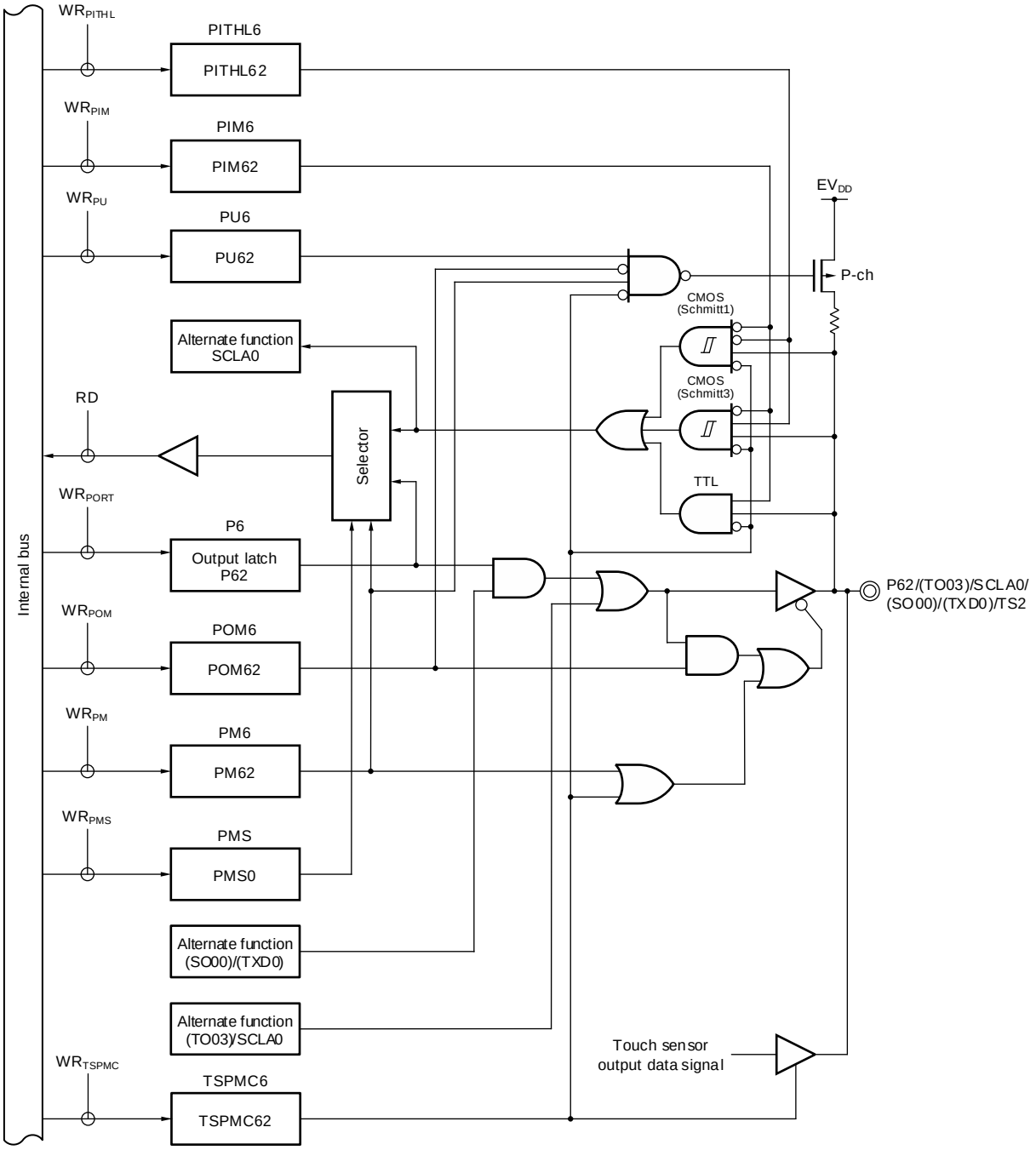
Figure 4-33. Block Diagram of P61



P6:	Port register 6	PITHL6:	Port input threshold control register 6
PU6:	Pull-up resistor option register 6	TSPMC6:	Touch sensor port mode control register 6
PM6:	Port mode register 6	RD:	Read signal
POM6:	Port output mode register 6	WRxx:	Write signal
PMS:	Port mode select register		

Caution The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

Figure 4-34. Block Diagram of P62



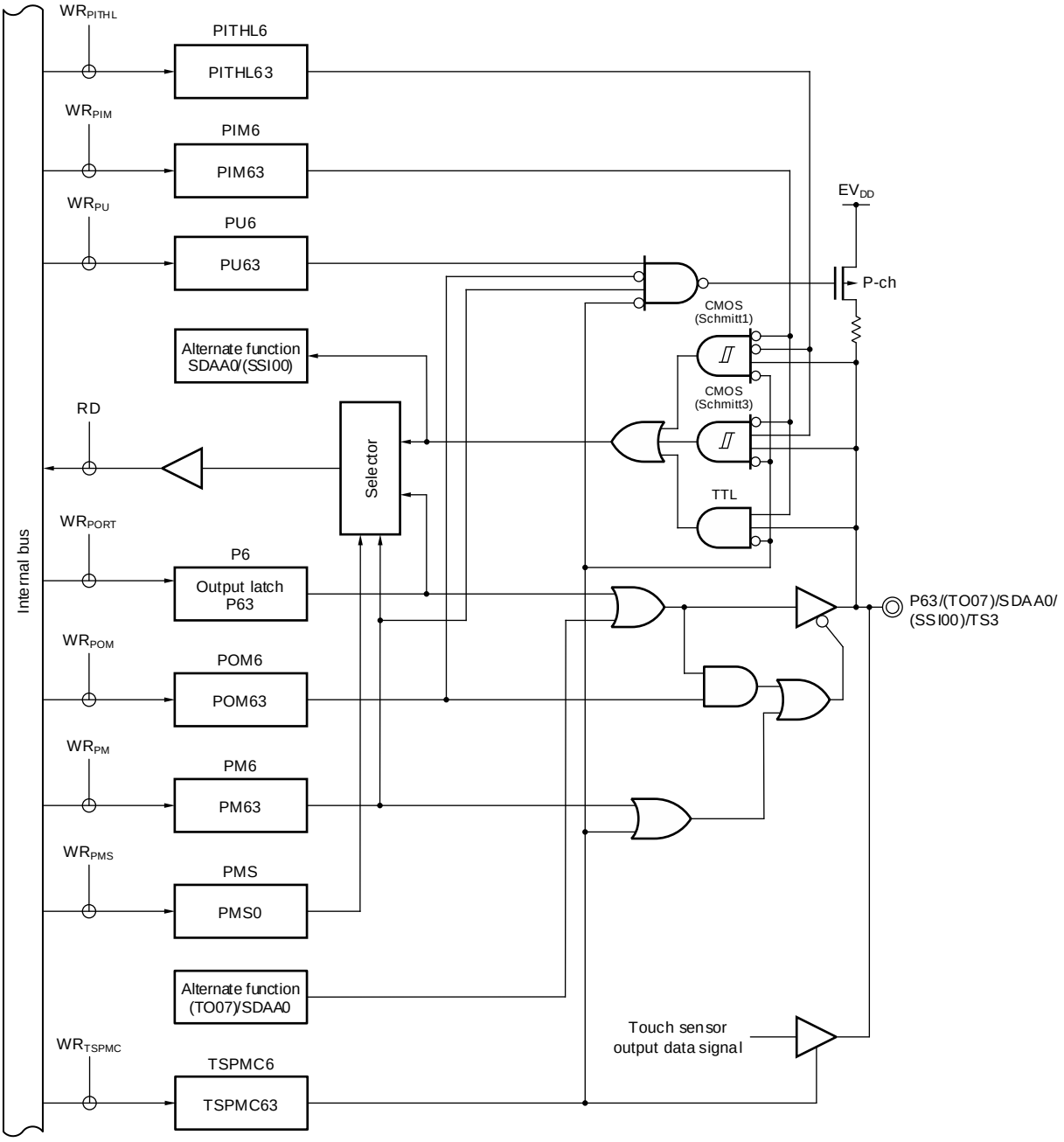
P6:	Port register 6	PITHL6:	Port input threshold control register 6
PU6:	Pull-up resistor option register 6	TSPMC6:	Touch sensor port mode control register 6
PM6:	Port mode register 6	RD:	Read signal
PIM6:	Port input mode register 6	WRxx:	Write signal
POM6:	Port output mode register 6		
PMS:	Port mode select register		

(**Cautions** are listed on the next page.)

Cautions 1. The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

2. When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.

Figure 4-35. Block Diagram of P63



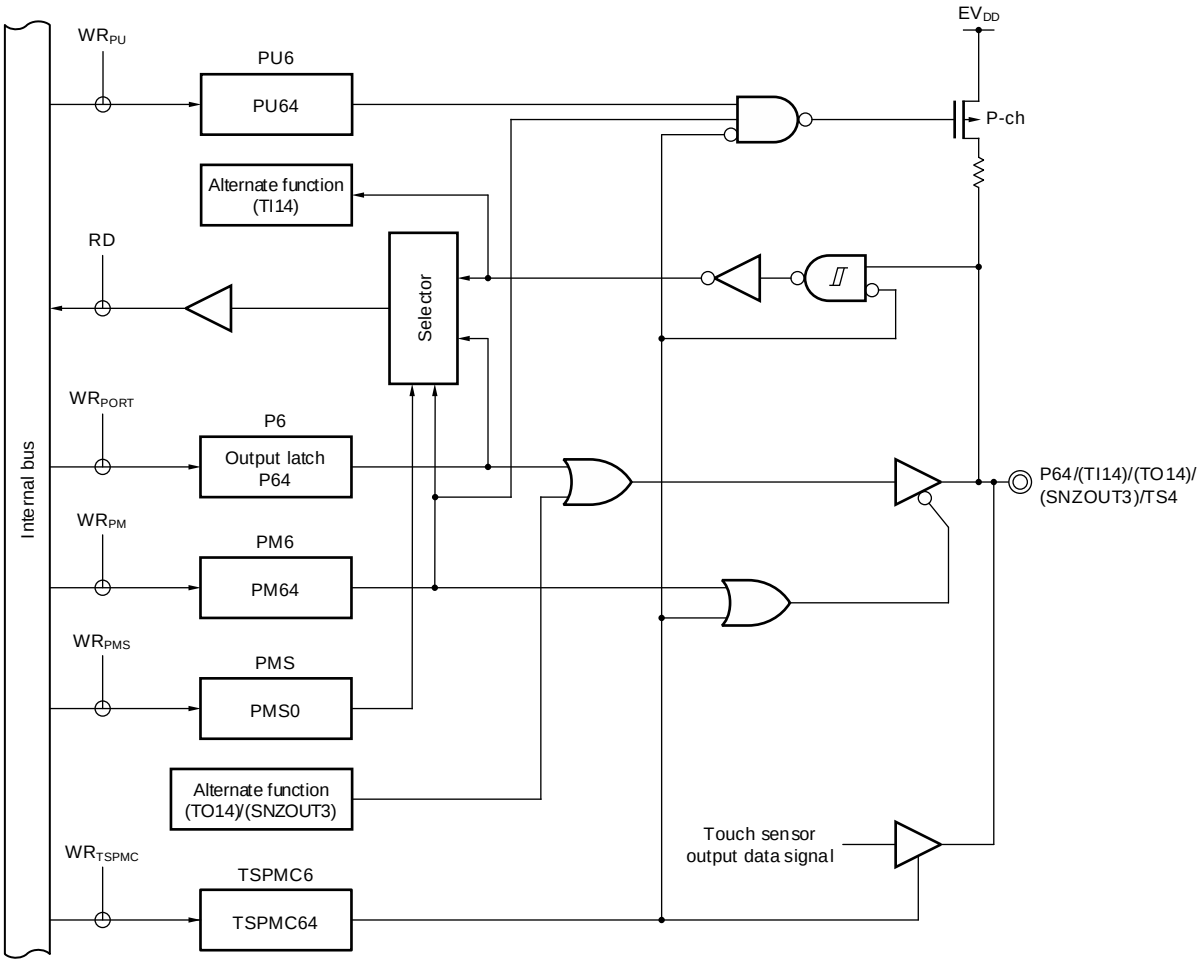
P6:	Port register 6	PITHL6:	Port input threshold control register 6
PU6:	Pull-up resistor option register 6	TSPMC6:	Touch sensor port mode control register 6
PM6:	Port mode register 6	RD:	Read signal
PIM6:	Port input mode register 6	WRxx:	Write signal
POM6:	Port output mode register 6		
PMS:	Port mode select register		

(**Cautions** are listed on the next page.)

Cautions 1. The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

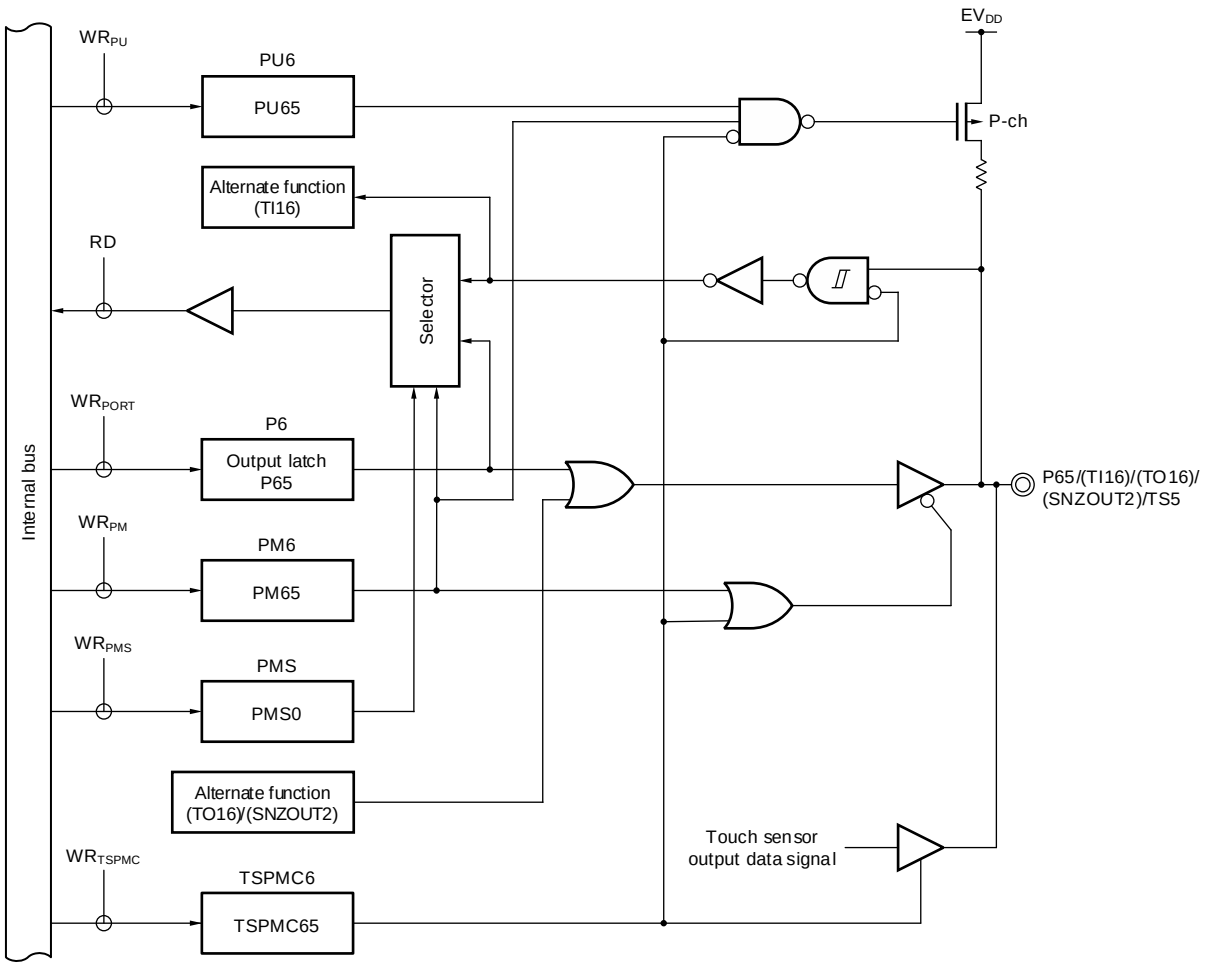
2. When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.

Figure 4-36. Block Diagram of P64



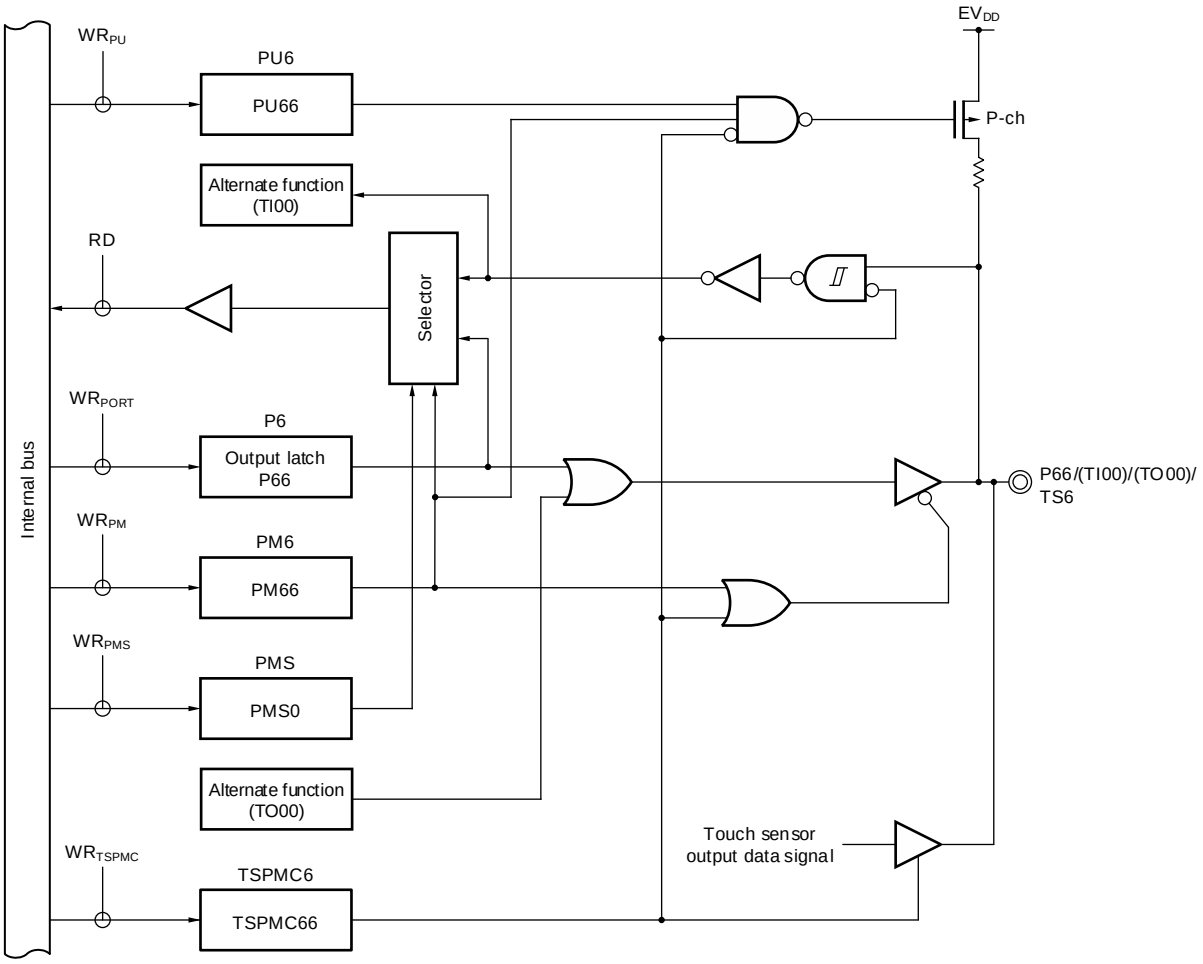
P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
PMS: Port mode select register
TSPMC6: Touch sensor port mode control register 6
RD: Read signal
WRxx: Write signal

Figure 4-37. Block Diagram of P65



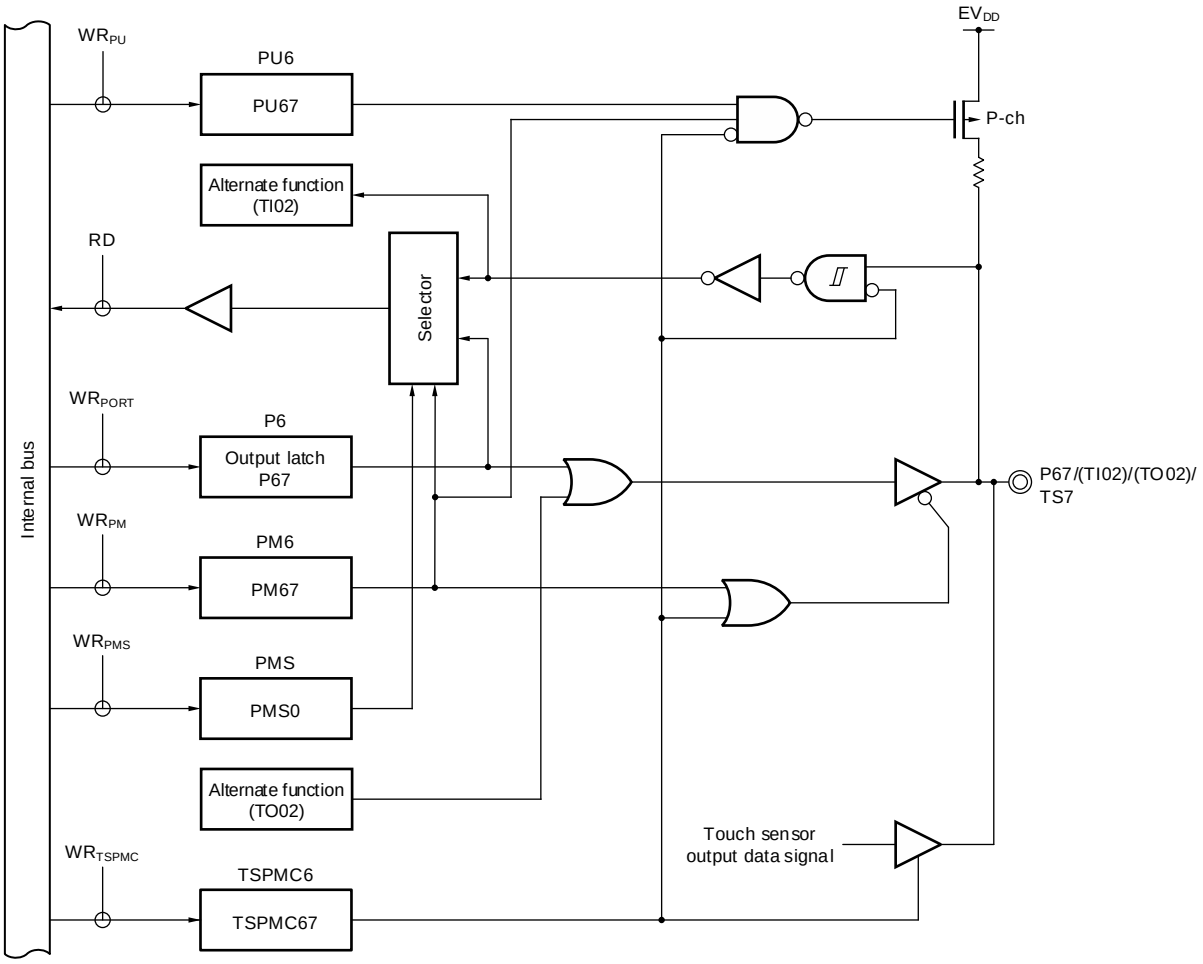
P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
PMS: Port mode select register
TSPMC6: Touch sensor port mode control register 6
RD: Read signal
WRxx: Write signal

Figure 4-38. 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
TSPMC6: Touch sensor port mode control register 6
RD: Read signal
WRxx: Write signal

Figure 4-39. 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
TSPMC6: Touch sensor port mode control register 6
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, serial interfaces data I/O, clock I/O, LIN serial data I/O, slave select input, timer I/O, external interrupt request input, SNOOZE status output, CAN serial data I/O, and capacitance measurement.

To use P70/ANI24 to P74/ANI28 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.

To use P70/ANI24 to P74/ANI28 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/ANI24 to P74/ANI28 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.

Reset signal generation sets P70 to P74 to analog input mode and P75 to P77 to input mode.

Table 4-12. Settings of Registers When Using Port 7

Pin Name		PM7x	PIM7x	POM7x	PMC7x	TSPMC7x	PITHL7x	Alternate Function Setting	Remarks
Name	I/O								
P70	Input	1	0	×	0	0	0	×	CMOS input (Schmitt1 input)
							1		CMOS input (Schmitt3 input)
			1	×	0	0	0	×	TTL input
	Output	0	×	0	0	0	×	SDA11 output = 1 Note 1 TO15 output = 0 Note 2 SNZOUT4 output = 0 Note 3 (LTXD2 output = 1) Note 5	CMOS output
			×	1	0	0	×		N-ch O.D output
P71	Input	1	0	×	0	0	0	×	CMOS input (Schmitt1 input)
							1		CMOS input (Schmitt3 input)
			1	×	0	0	0	×	TTL input
	Output	0	×	0	0	0	×	SCK11 output = 1 Note 1 TO17 output = 0 Note 2 SCL11 output = 1 Note 1 SNZOUT5 output = 0 Note 3	CMOS output
			×	1	0	0	×		N-ch O.D output
P72	Input	1	–	×	0	0	–	×	
	Output	0	–	0	0	0	–	SO11 output = 1 Note 1 SNZOUT6 output = 0 Note 3 (CTXD0 output = 1) Note 4	CMOS output
				1	0	0	–		N-ch O.D output
P73	Input	1	0	–	0	0	0	×	CMOS input (Schmitt1 input)
							1	×	CMOS input (Schmitt3 input)
			1	–	0	0	0	×	TTL input
	Output	0	×	–	0	0	×	SNZOUT7 output = 0 Note 3	CMOS output
P74	Input	1	–	–	0	0	–	×	
	Output	0	–	–	0	0	–	(SO10 output = 1) Note 1 (TXD1 output = 1) Note 1	
P75	Input	1	–	–	–	0	0	×	CMOS input (Schmitt1 input)
							1	×	CMOS input (Schmitt3 input)
	Output	0	–	–	–	0	×	×	
P76	Input	1	–	–	–	0	0	×	CMOS input (Schmitt1 input)
							1	×	CMOS input (Schmitt3 input)
	Output	0	–	–	–	0	×	(SCK10 output = 1) Note 1	
P77	Input	1	–	–	–	0	0	×	CMOS input (Schmitt1 input)
							1	×	CMOS input (Schmitt3 input)
	Output	0	–	–	–	0	×	×	

Important To use the port 7 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

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

Notes 1. When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn and CKOmn bits 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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).

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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR1, 3, 9).

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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR6).

4. 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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR4).

5. When using a pin that is also used for the LIN serial data output function as a general-purpose port, deactivate the corresponding LIN. Alternatively, assign the corresponding function to another pin using the peripheral I/O redirection register (PIOR9).

Remarks ×: Don't care
PM7x: Bit of port mode register 7
PIM7x: Bit of port input mode register 7
POM7x: Bit of port output mode register 7
PMC7x: Bit of port mode control register 7
PITHL7x: Bit of port input threshold control register 7
TSPMC7x: Bit of touch sensor port mode control register 7

Table 4-13. Setting Functions of P70/ANI24 to P74/ANI28 Pins

PMC7 Register	PM7 Register	ADANSA1, ADANSB1 Register	P70/ANI24 to P74/ANI28 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/ANI24 to P74/ANI28 to analog input mode.

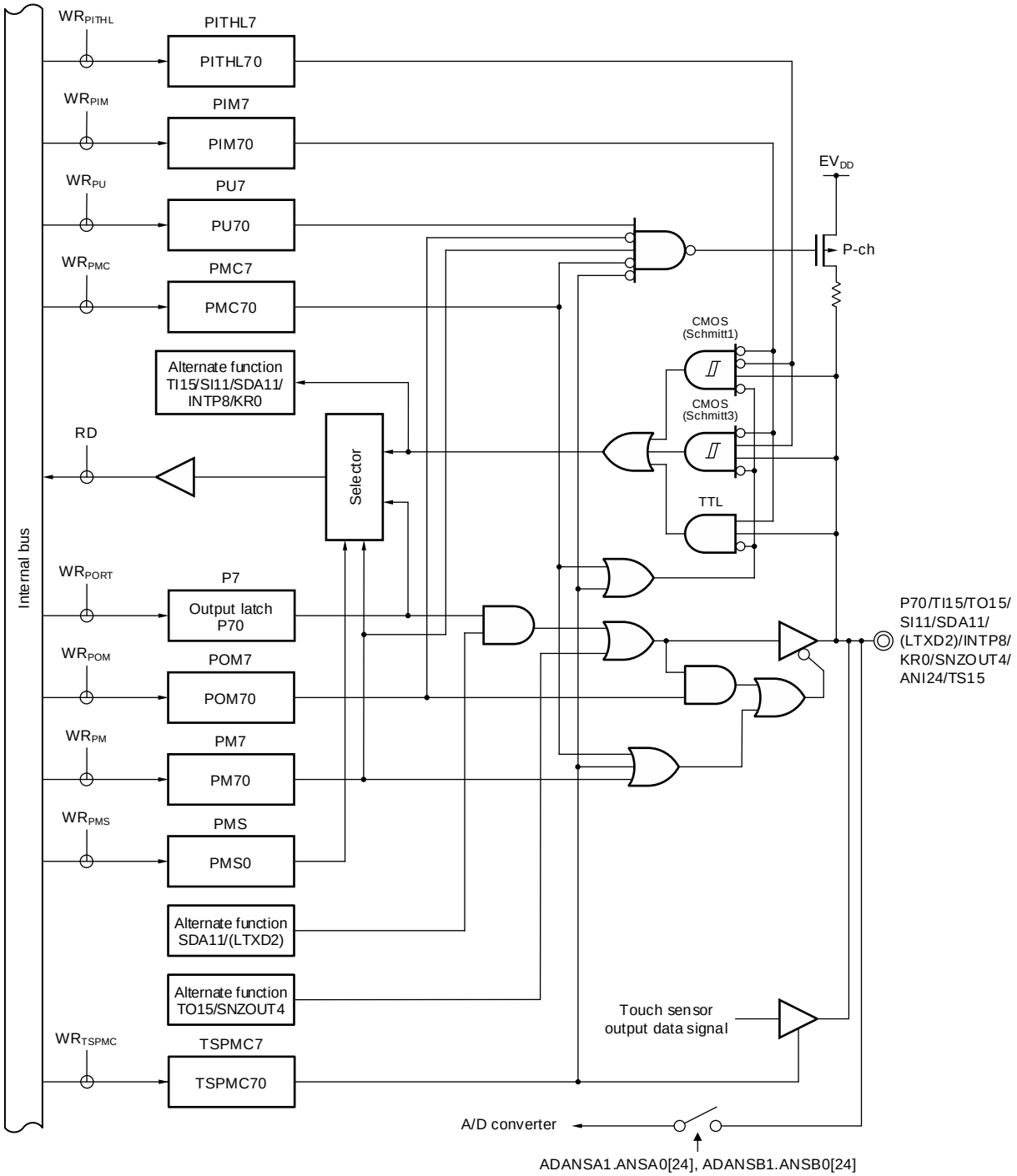
Table 4-14. P70/TS15 to P77/TS8 Pin Function Setting Note

TSPMC7 Register	PM7 Register	P70/TS15 to P77/TS8 Pins
Digital input/output selection	Input mode	Digital input
	Output mode	Digital output
Capacitance measurement function selection	Input mode	Capacitance measurement function
	Output mode	Setting prohibited

Note The above settings are for when a capacitive sensing unit is used as the capacitance measurement function. When selecting the capacitance measurement function, set the corresponding bit to 0 (digital I/O) in the PMC7 register.

Figure 4-40 to **Figure 4-47** show block diagrams of port 7 for 100-pin products.

Figure 4-40. Block Diagram of P70

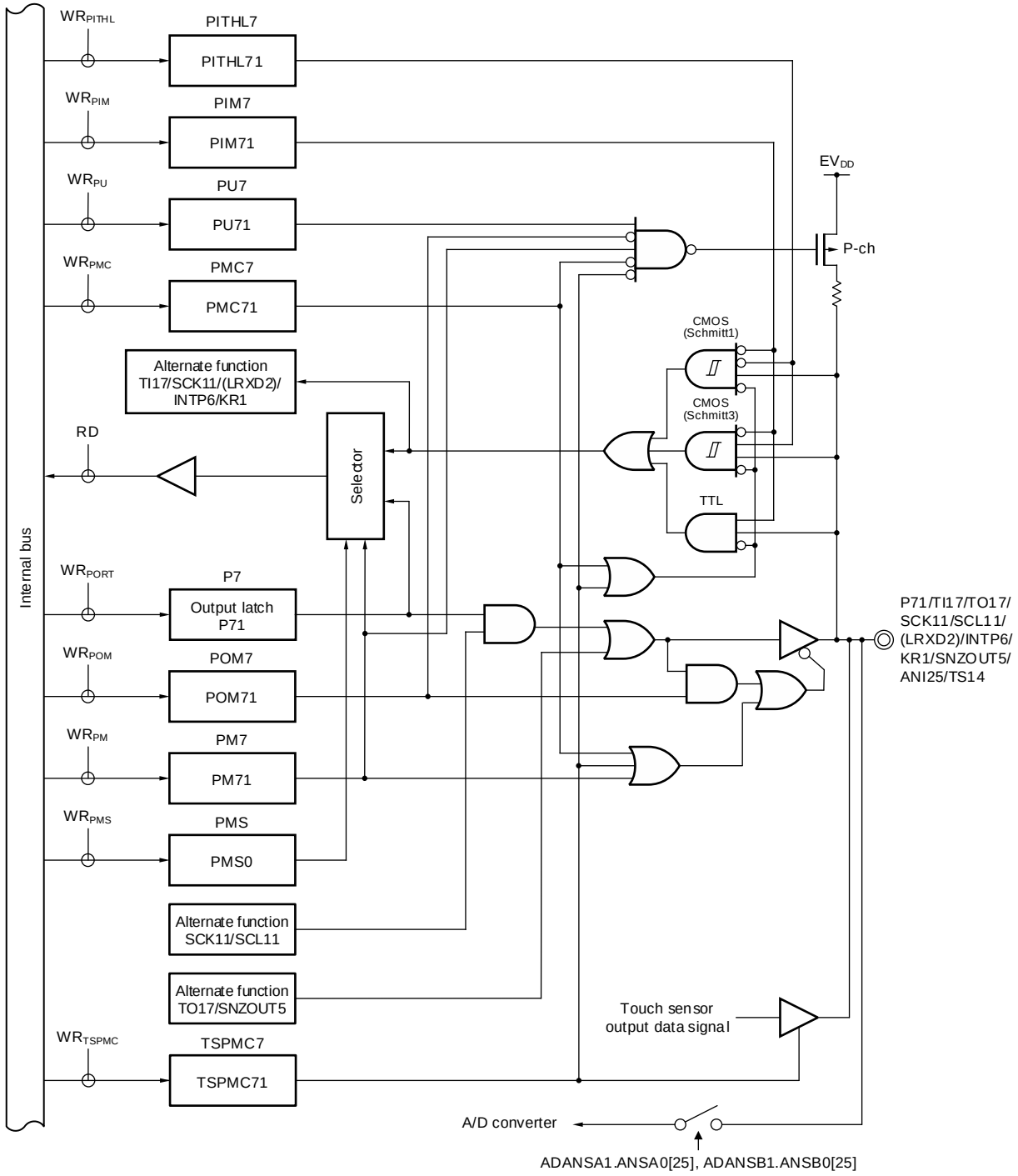


P7:	Port register 7	PMS:	Port mode select register
PU7:	Pull-up resistor option register 7	PITHL7:	Port input threshold control register 7
PM7:	Port mode register 7	TSPMC7:	Touch sensor port mode control register 7
PIM7:	Port input mode register 7	RD:	Read signal
POM7:	Port output mode register 7	WRxx:	Write signal
PMC7:	Port mode control register 7		

(**Cautions** are listed on the next page.)

Cautions
1. **The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.**
2. **When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.**

Figure 4-41. Block Diagram of P71



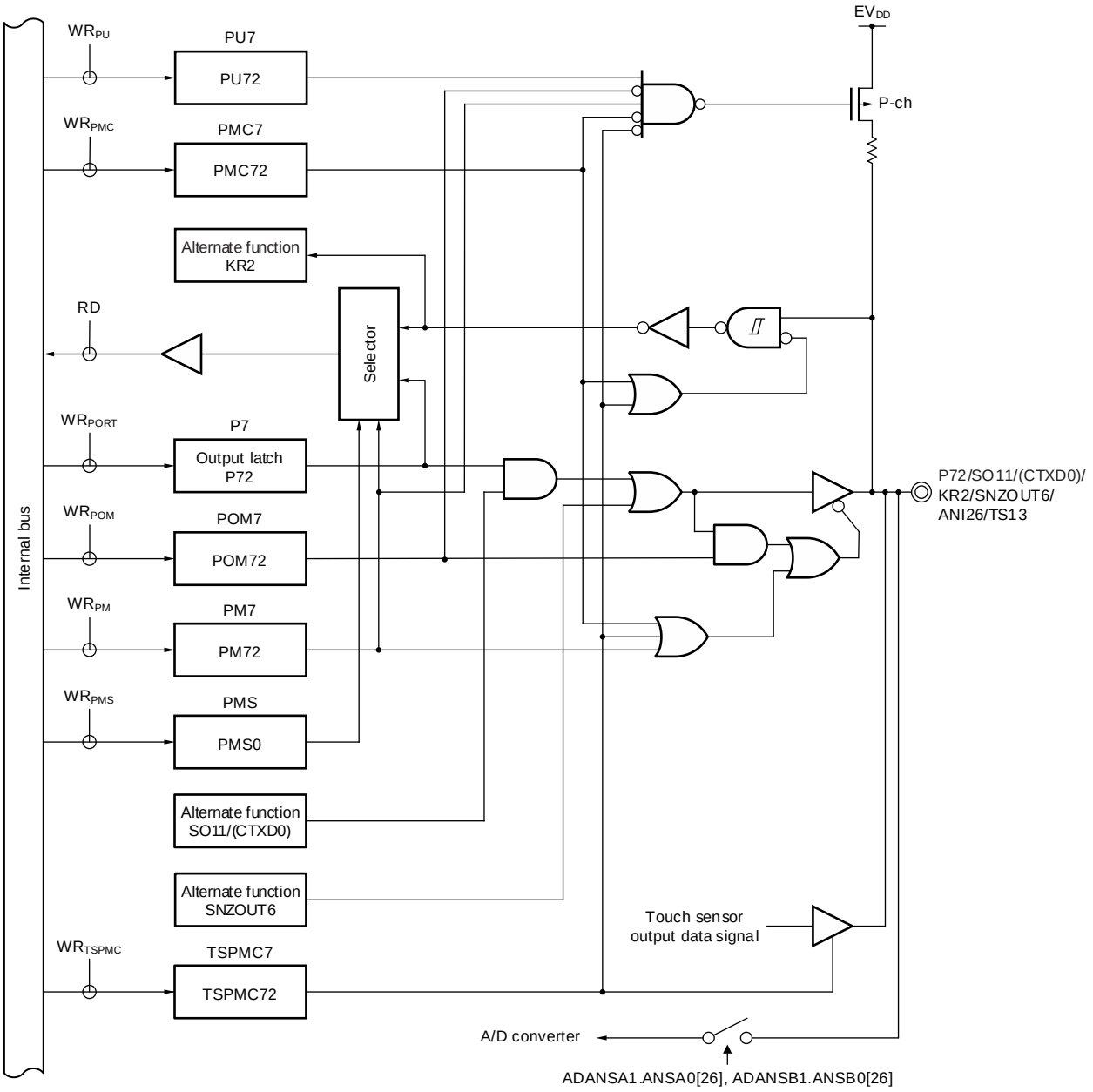
P7:	Port register 7	PMS:	Port mode select register
PU7:	Pull-up resistor option register 7	PITHL7:	Port input threshold control register 7
PM7:	Port mode register 7	TSPMC7:	Touch sensor port mode control register 7
PIM7:	Port input mode register 7	RD:	Read signal
POM7:	Port output mode register 7	WRxx:	Write signal
PMC7:	Port mode control register 7		

(**Cautions** are listed on the next page.)

Cautions 1. The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

2. When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.

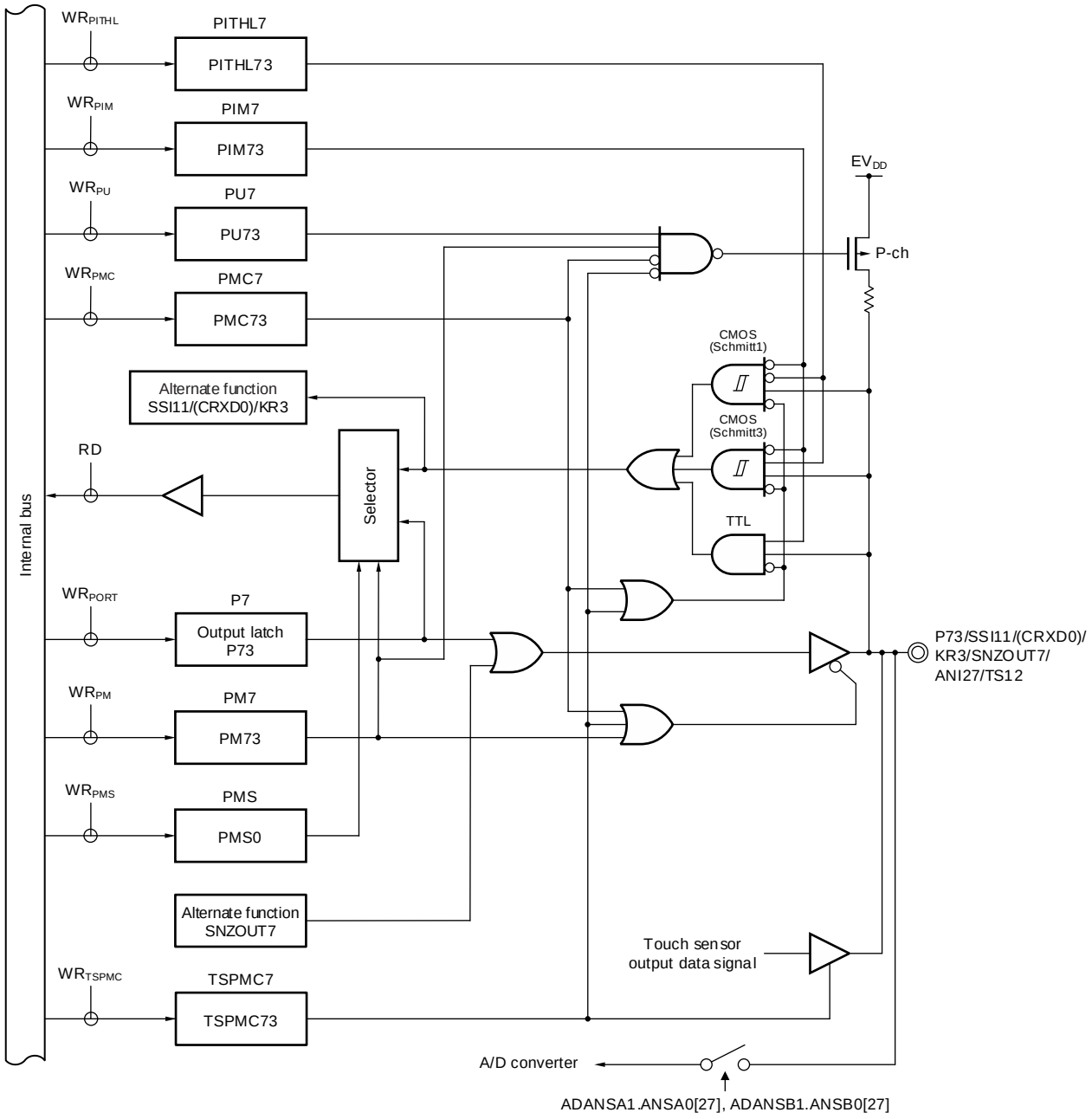
Figure 4-42. Block Diagram of P72



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
TSPMC7: Touch sensor port mode control register 7
RD: Read signal
WRxx: Write signal

Caution The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

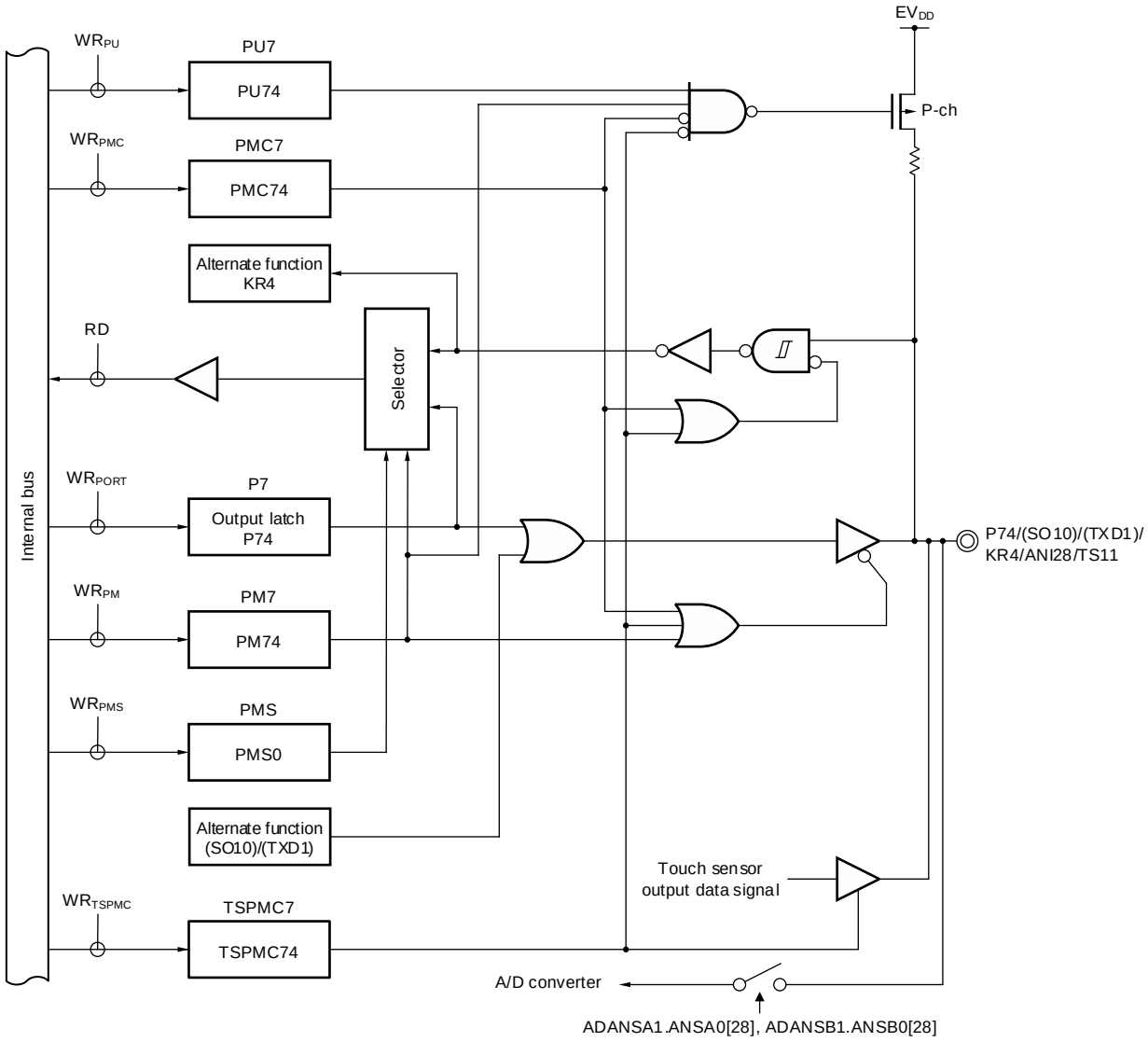
Figure 4-43. Block Diagram of P73



P7:	Port register 7	PMS:	Port mode select register
PU7:	Pull-up resistor option register 7	PITHL7:	Port input threshold control register 7
PM7:	Port mode register 7	TSPMC7:	Touch sensor port mode control register 7
PIM7:	Port input mode register 7	RD:	Read signal
PMC7:	Port mode control register 7	WRxx:	Write signal

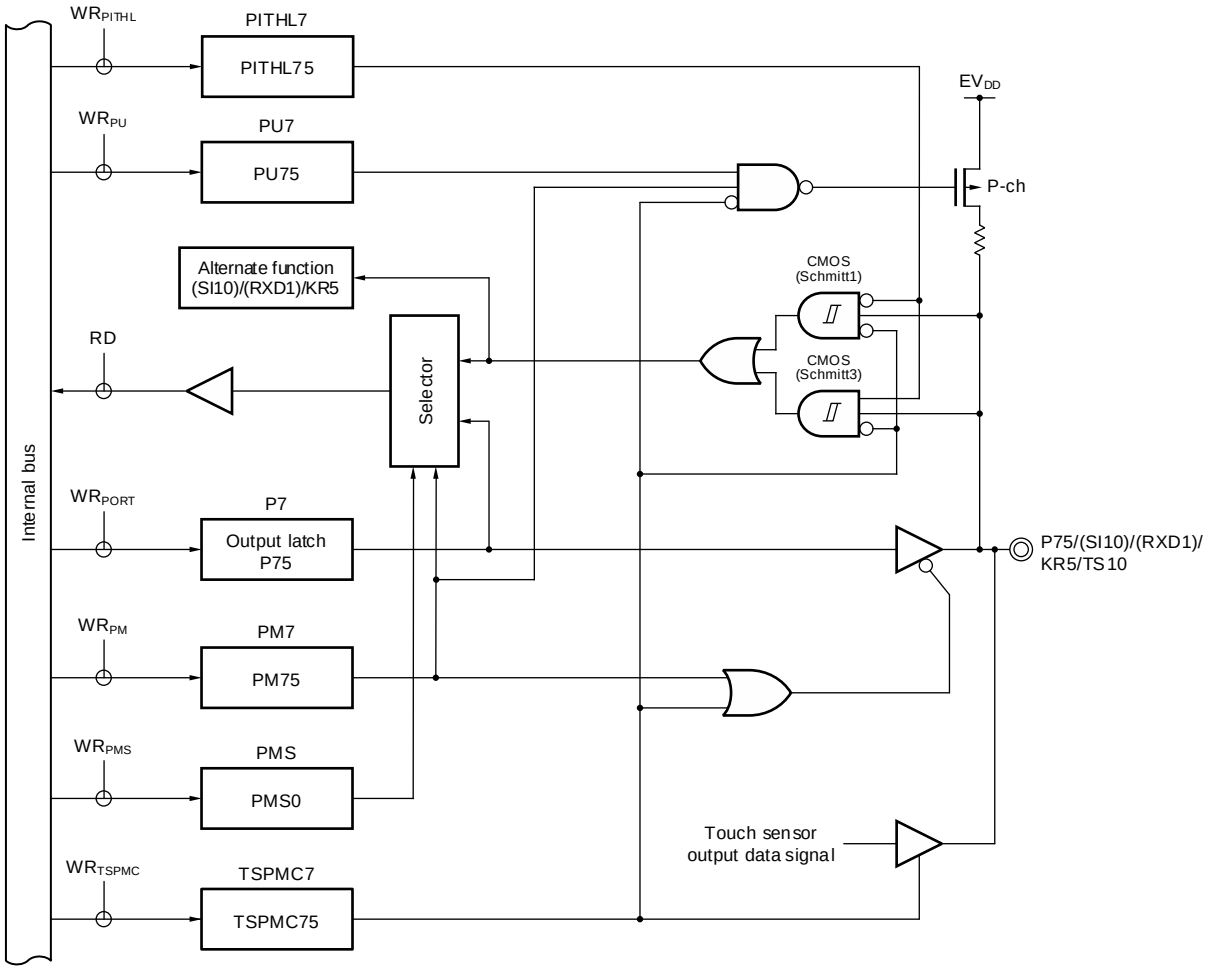
Caution When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.

Figure 4-44. Block Diagram of P74



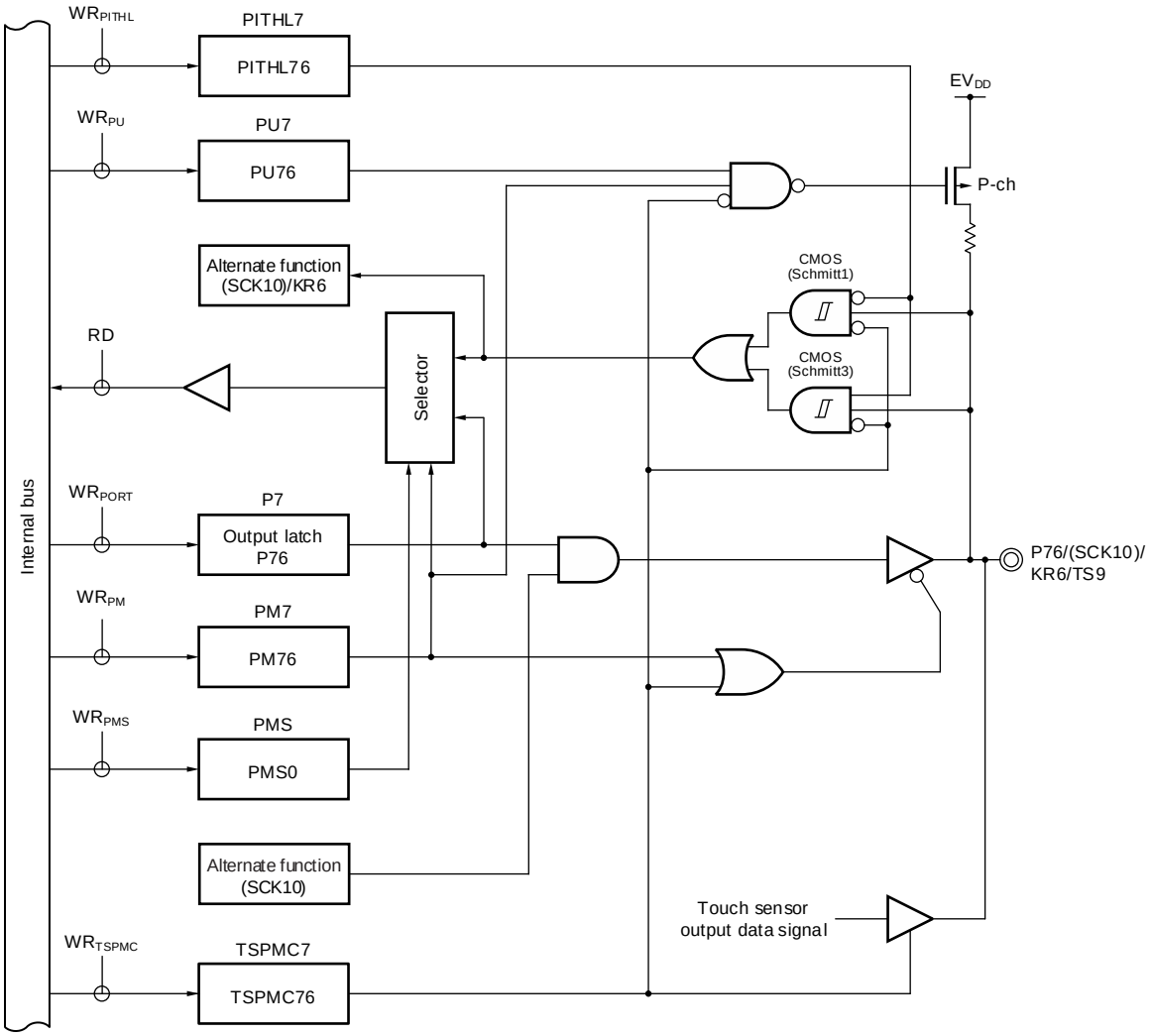
P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
PMC7: Port mode control register 7
PMS: Port mode select register
TSPMC7: Touch sensor port mode control register 7
RD: Read signal
WRxx: Write signal

Figure 4-45. Block Diagram of P75



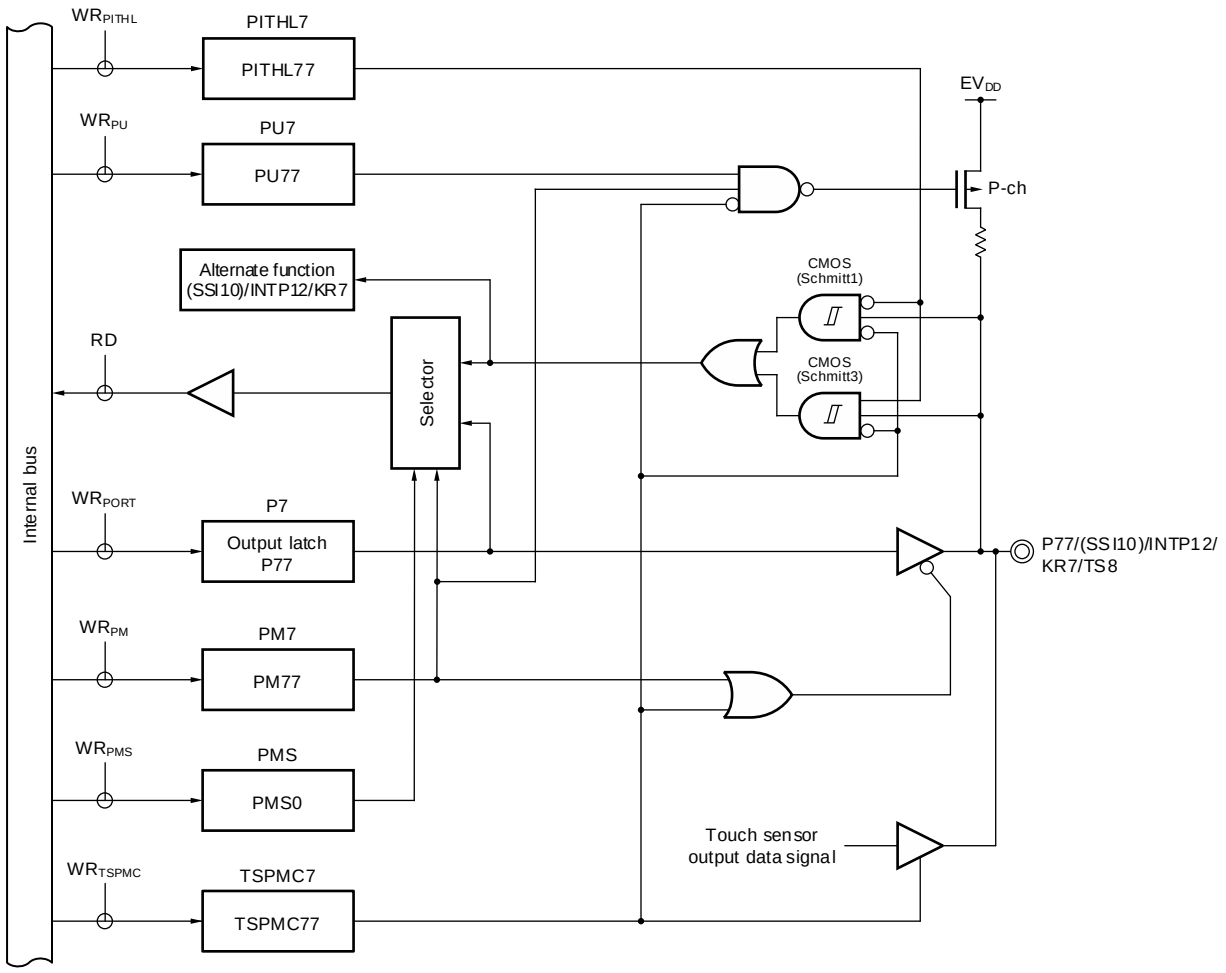
P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
PMS: Port mode select register
PITHL7: Port input threshold control register 7
TSPMC7: Touch sensor port mode control register 7
RD: Read signal
WRxx: Write signal

Figure 4-46. 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
TSPMC7: Touch sensor port mode control register 7
RD: Read signal
WRxx: Write signal

Figure 4-47. 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
TSPMC7: Touch sensor port mode 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, P82 to P85 can also be used for D/A converter output, and analog voltage input and reference voltage input for comparator.

To use P80/ANI0 to P87/ANI7 as digital input pins, set them in the digital I/O mode by using the port mode control register 8 (PMC8) and in the input mode by using the PM8 register.

To use P80/ANI0 to P87/ANI7 as digital output pins, set them in the digital I/O mode by using the port mode control register 8 (PMC8) and in the output mode by using the PM8 register.

To use P80/ANI0 to P87/ANI7 as analog input pins, set them in the analog I/O mode by using the port mode control register 8 (PMC8) and in the input mode by using the PM8 register.

Reset signal generation sets this port to analog input mode.

Table 4-15. Setting Functions of P80/ANI0/ANO0 Pin

PMC8 Register	PM8 Register	DAM Register	DAM2 Register	ADANSA0, ADANSB0 Registers	Functions of ANO0/ANI0/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 output
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).

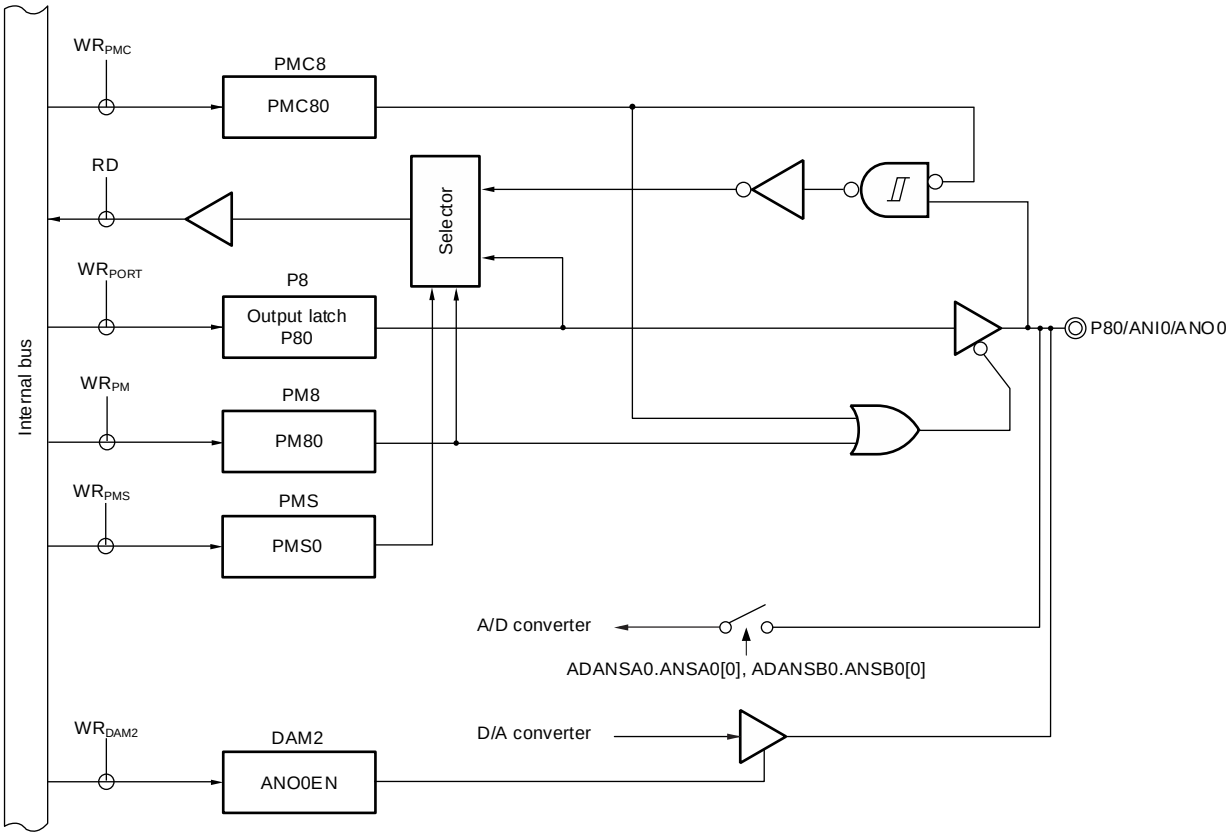
Table 4-16. Setting Functions of P81/ANI1 to P87/ANI7 Pins

PMC8 Register	PM8 Register	ADANSA0, ADANSB0 Registers	P81/ANI1 to P87/ANI7 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/ANI0 to P87/ANI7 to analog input mode.

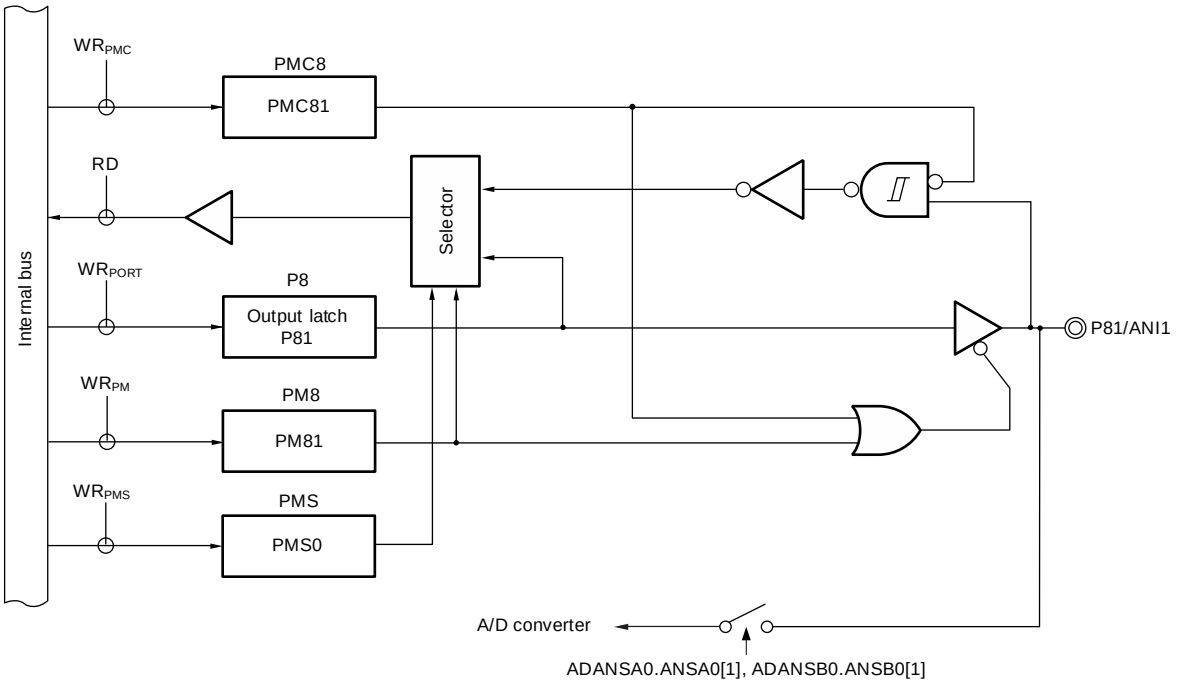
Figure 4-48 to **Figure 4-55** show block diagrams of port 8 for 100-pin products.

Figure 4-48. Block Diagram of P80



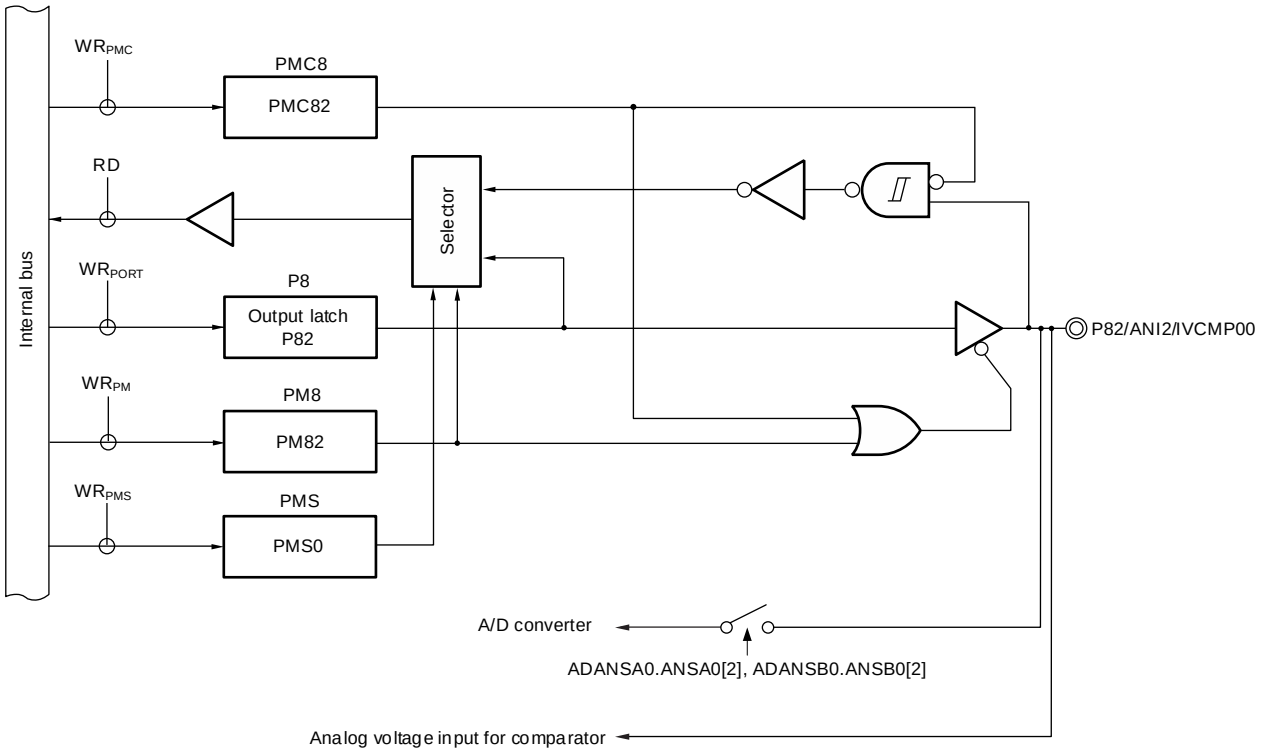
P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
PMC8: Port mode control register 8
DAM2: D/A converter mode register 2
RD: Read signal
WRxx: Write signal

Figure 4-49. Block Diagram of P81



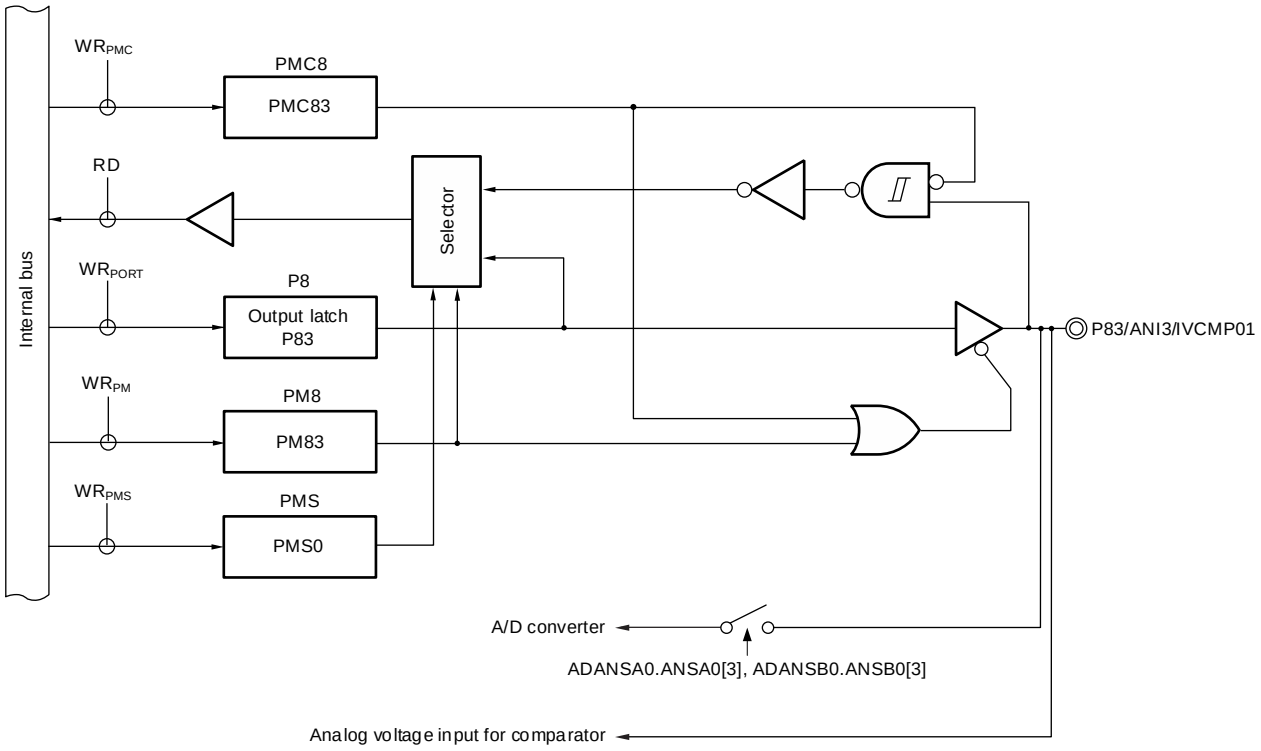
P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
PMC8: Port mode control register 8
RD: Read signal
WRxx: Write signal

Figure 4-50. Block Diagram of P82



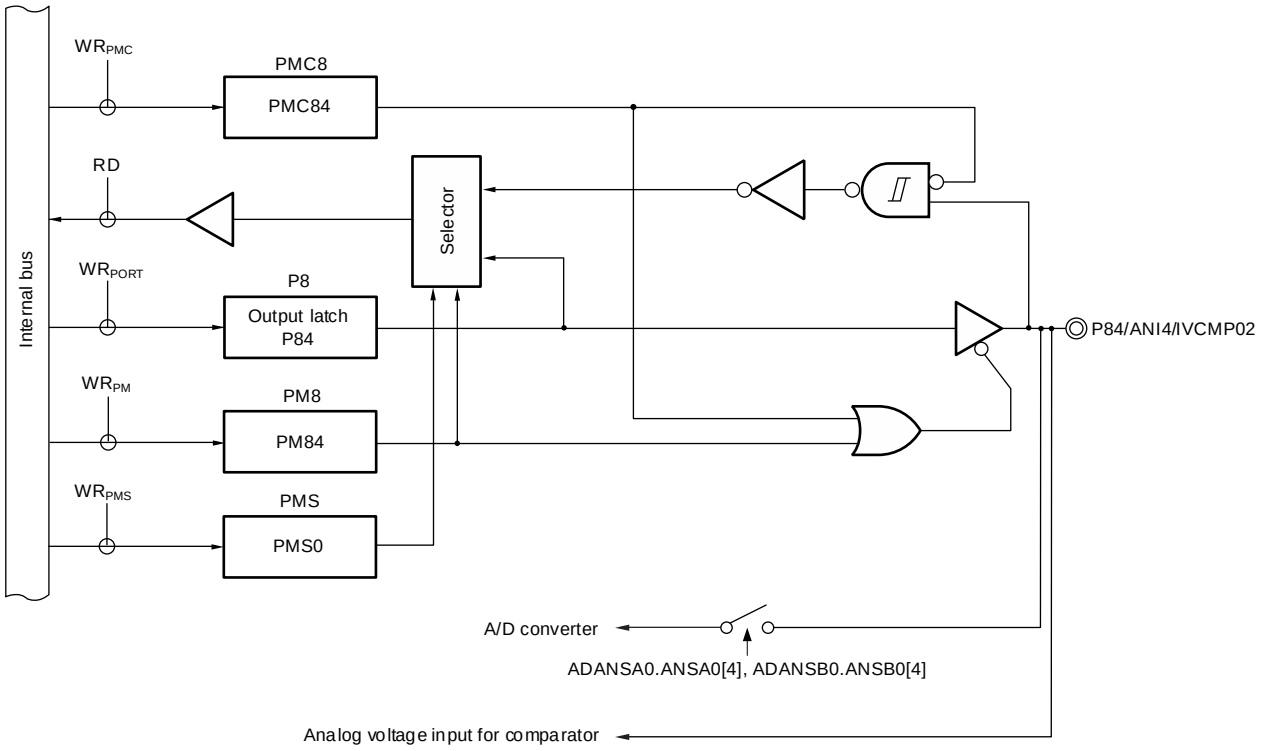
P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
PMC8: Port mode control register 8
RD: Read signal
WRxx: Write signal

Figure 4-51. Block Diagram of P83



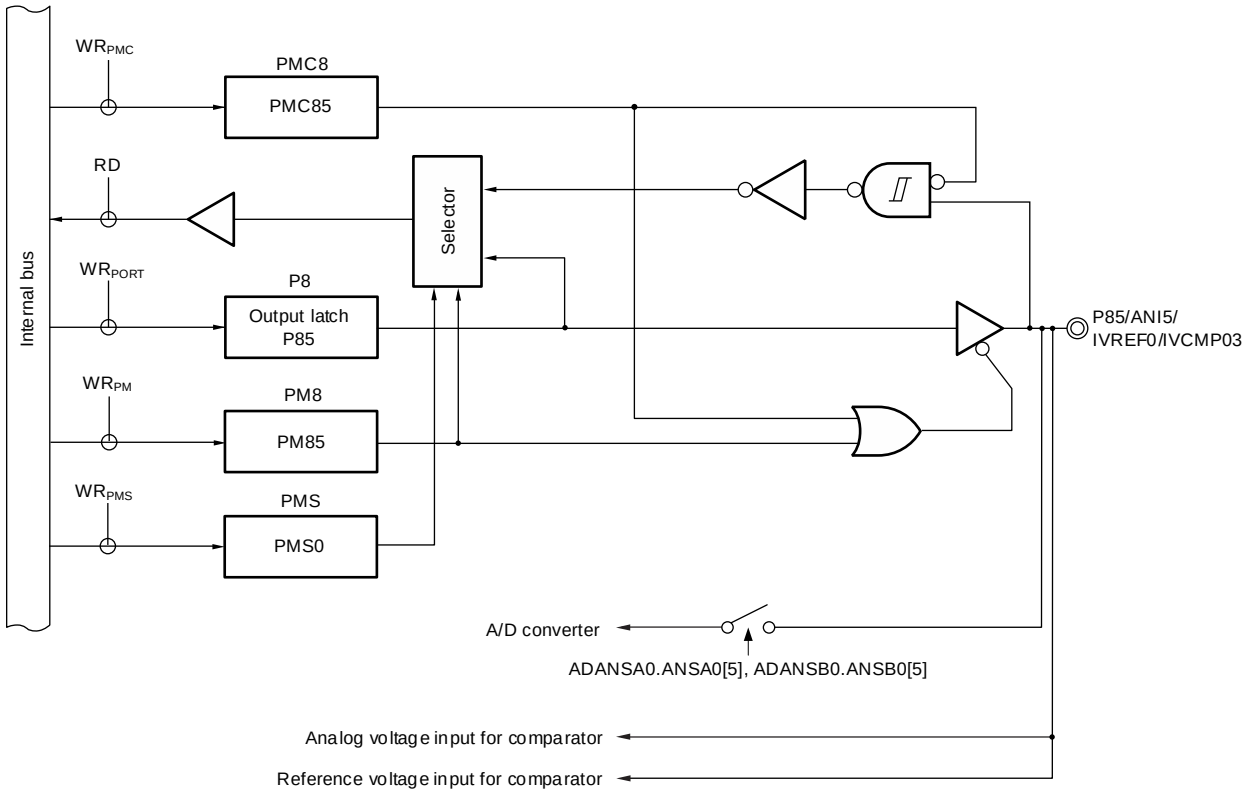
P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
PMC8: Port mode control register 8
RD: Read signal
WRxx: Write signal

Figure 4-52. Block Diagram of P84



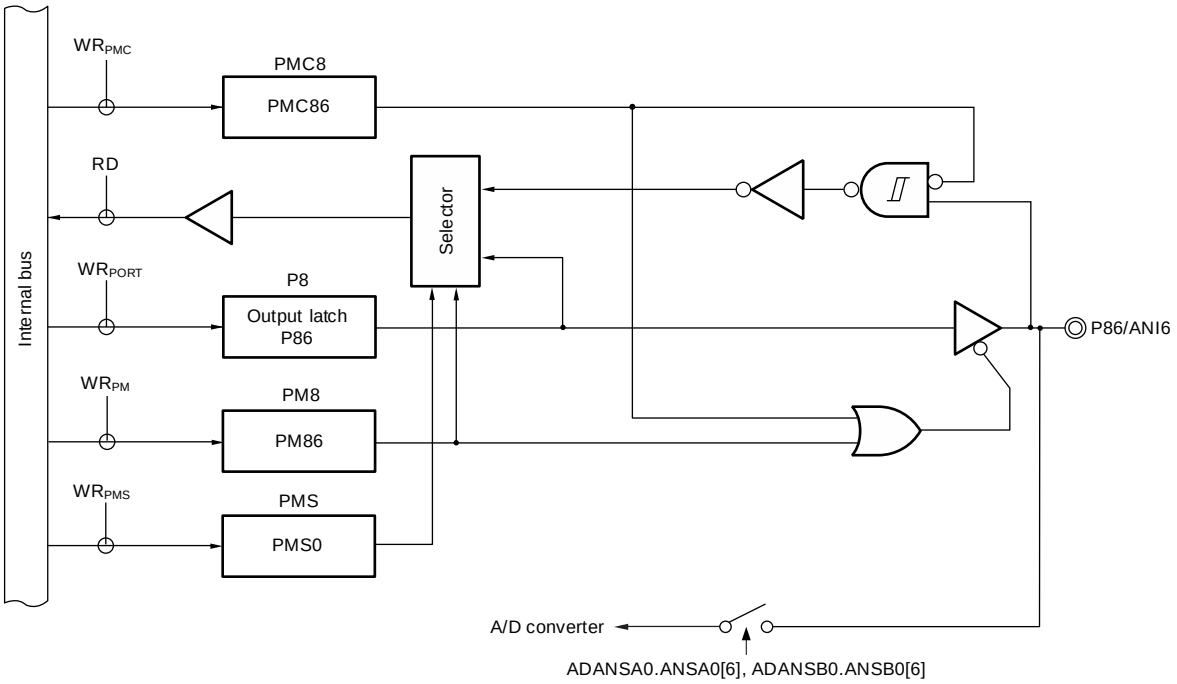
P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
PMC8: Port mode control register 8
RD: Read signal
WRxx: Write signal

Figure 4-53. Block Diagram of P85



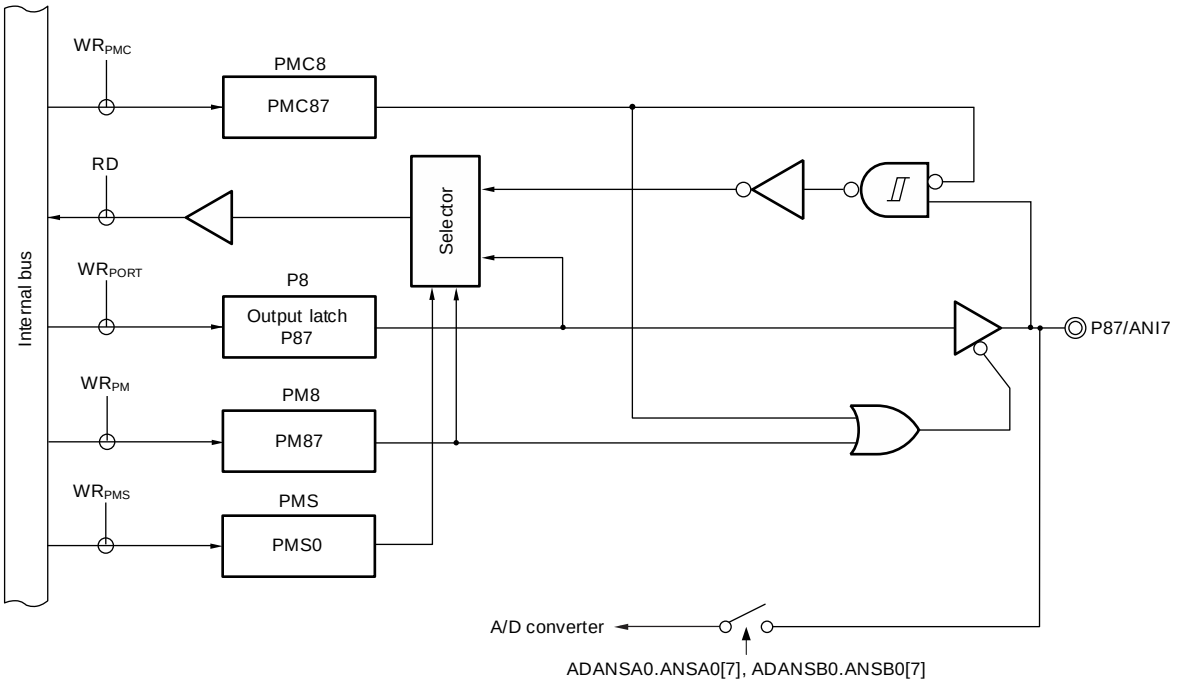
P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
PMC8: Port mode control register 8
RD: Read signal
WRxx: Write signal

Figure 4-54. Block Diagram of P86



P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
PMC8: Port mode control register 8
RD: Read signal
WRxx: Write signal

Figure 4-55. Block Diagram of P87



P8: Port register 8
PM8: Port mode register 8
PMS: Port mode select register
PMC8: Port mode control register 8
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/ANI8 to P97/ANI15 as digital input pins, set them in the digital I/O mode by using the port mode control register 9 (PMC9) and in the input mode by using the PM9 register.

To use P90/ANI8 to P97/ANI15 as digital output pins, set them in the digital I/O mode by using the port mode control register 9 (PMC9) and in the output mode by using the PM9 register.

To use P90/ANI8 to P97/ANI15 as analog input pins, set them in the analog input mode by using the port mode control register 9 (PMC9) and in the input mode by using the PM9 register.

Reset signal generation sets this port to analog input mode.

Table 4-17. Settings of Registers When Using Port 9

Pin Name		PM9x	Alternate Function Setting
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	×

Remarks ×: Don't care
PM9x: Bit of port mode register 9

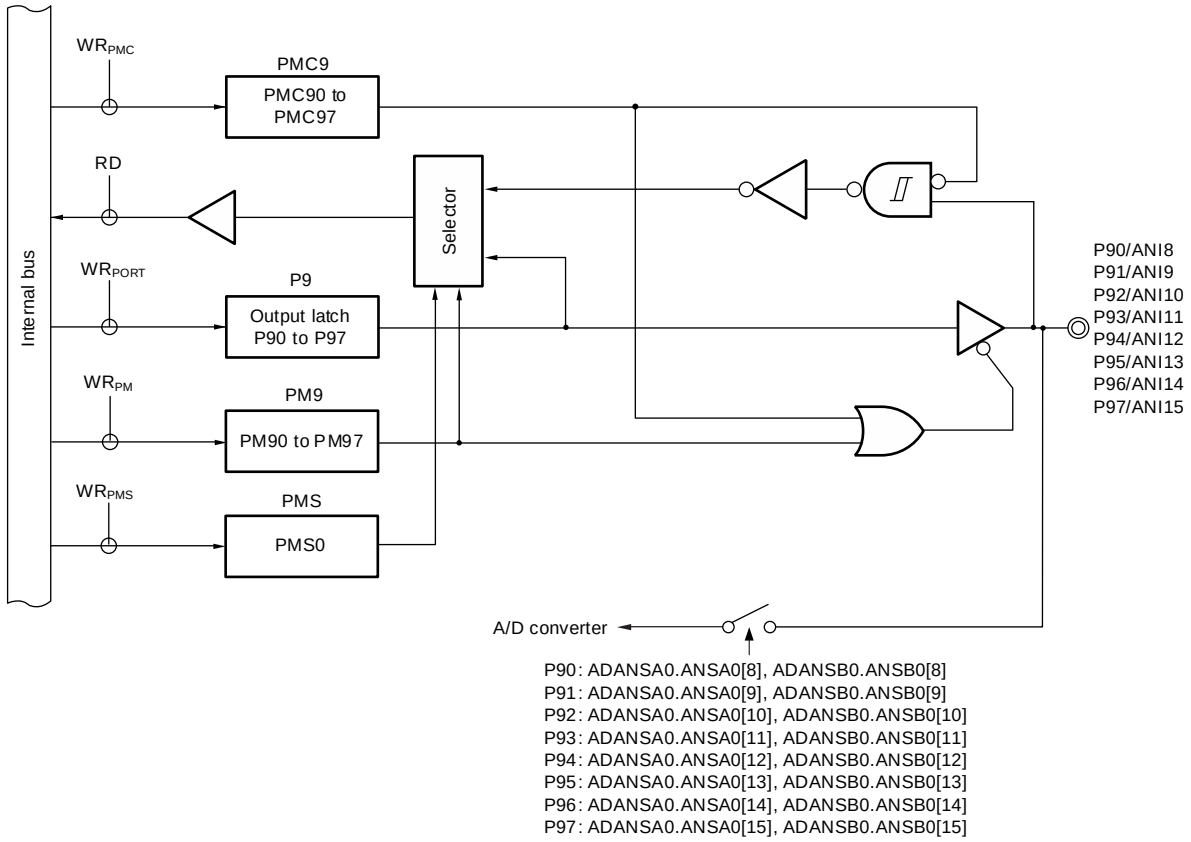
Table 4-18. Setting Functions of P90/ANI8 to P97/ANI15 Pins

PMC9 Register	PM9 Register	ADANSA0, ADANSB0 Registers	P90/ANI8 to P97/ANI15 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/ANI8 to P97/ANI15 to analog input mode.

Figure 4-56 shows a block diagram of port 9 for 100-pin products.

Figure 4-56. Block Diagram of P90 to P97



P9: Port register 9
PM9: Port mode register 9
PMS: Port mode select register
PMC9: Port mode control register 9
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/ANI16 to P105/ANI21 as digital input pins, set them in the digital I/O mode by using the port mode control register 10 (PMC10) and in the input mode by using the PM10 register.

To use P100/ANI16 to P105/ANI21 as digital output pins, set them in the digital I/O mode by using the port mode control register 10 (PMC10) and in the output mode by using the PM10 register.

To use P100/ANI16 to P105/ANI21 as analog input pins, set them in the analog input mode by using the port mode control register 10 (PMC10) and in the input mode by using the PM10 register.

Reset signal generation sets P100 to P105 to analog input mode and P106 to P107 to input mode.

Table 4-19. Settings of Registers When Using Port 10

Pin Name		PM10x	PITHL10x	Alternate Function Setting	Remarks
Name	I/O				
P100 Note 2	Input	1	–	×	
	Output	0	–	×	
P101 Note 2	Input	1	–	×	
	Output	0	–	×	
P102 Note 2	Input	1	–	×	
	Output	0	–	×	
P103 Note 2	Input	1	–	×	
	Output	0	–	×	
P104 Note 2	Input	1	–	×	
	Output	0	–	×	
P105 Note 2	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	×	×	

Important To use the port 10 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

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

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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).

2. These are the settings of registers in the case where setting of the Port mode control register 10 (PMC10) is to select “0” (digital I/O) for the target pin. See **Table 4-20** when using this pin as an analog input.

Remarks ×: Don't care
PM10x: Bit of port mode register 10
PITHL10x: Bit of port input threshold control register 10

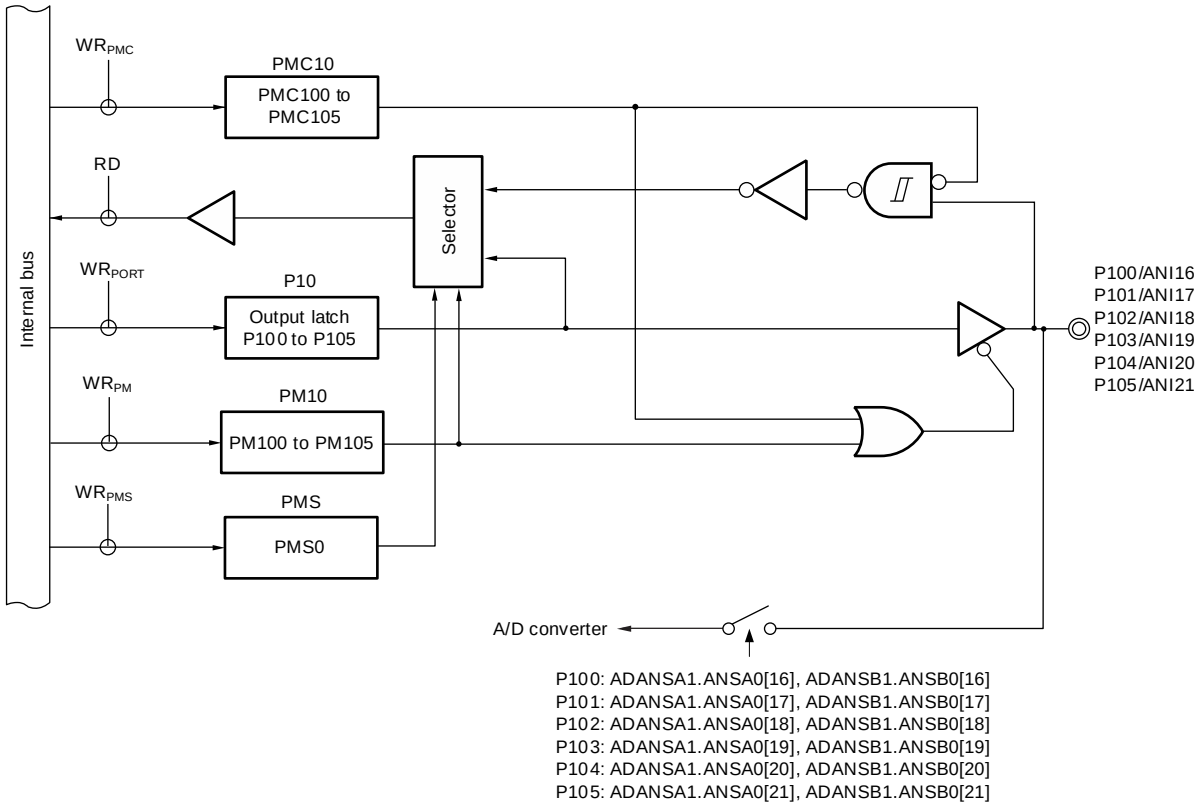
Table 4-20. Setting Functions of P100/ANI16 to P105/ANI21 Pins

PMC10 Register	PM10 Register	ADANSA1, ADANSB1 Registers	P100/ANI16 to P105/ANI21 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/ANI16 to P105/ANI21 to analog input mode.

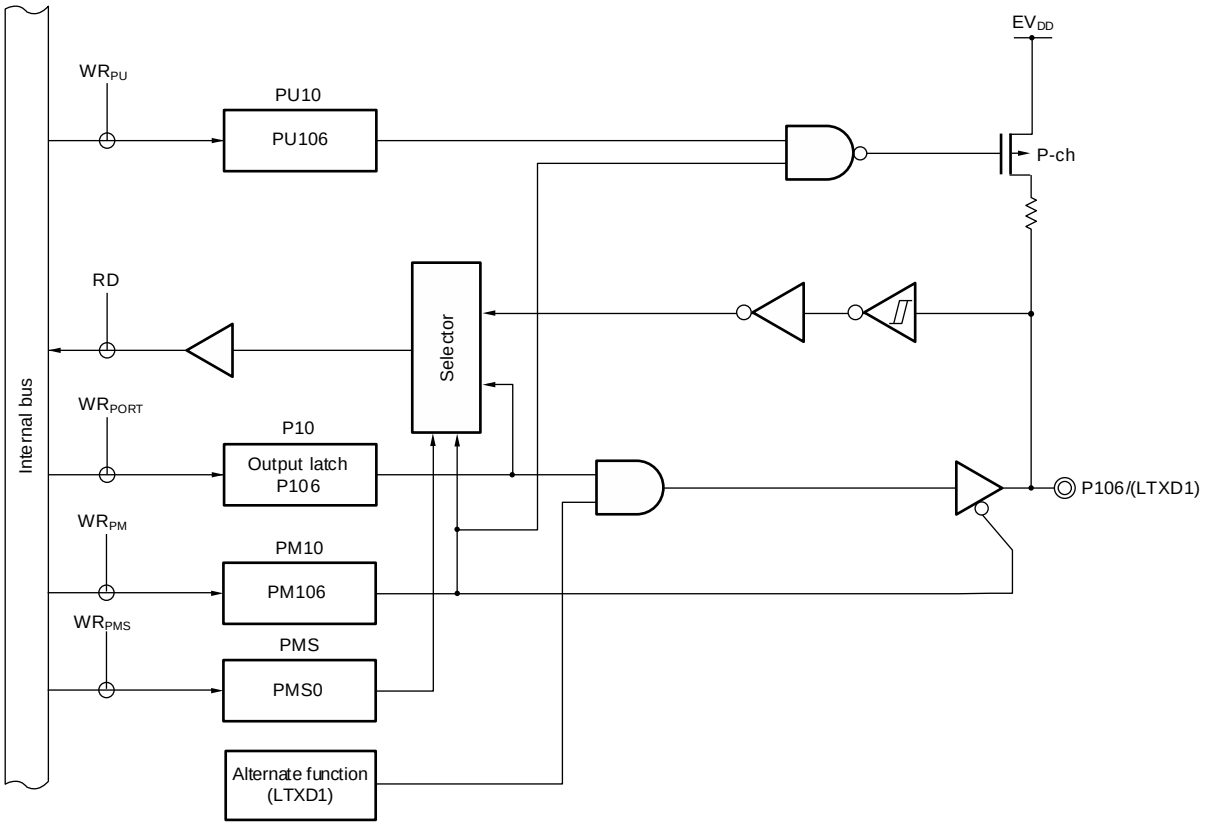
Figure 4-57 to **Figure 4-59** show block diagrams of port 10 for 100-pin products.

Figure 4-57. Block Diagram of P100 to P105



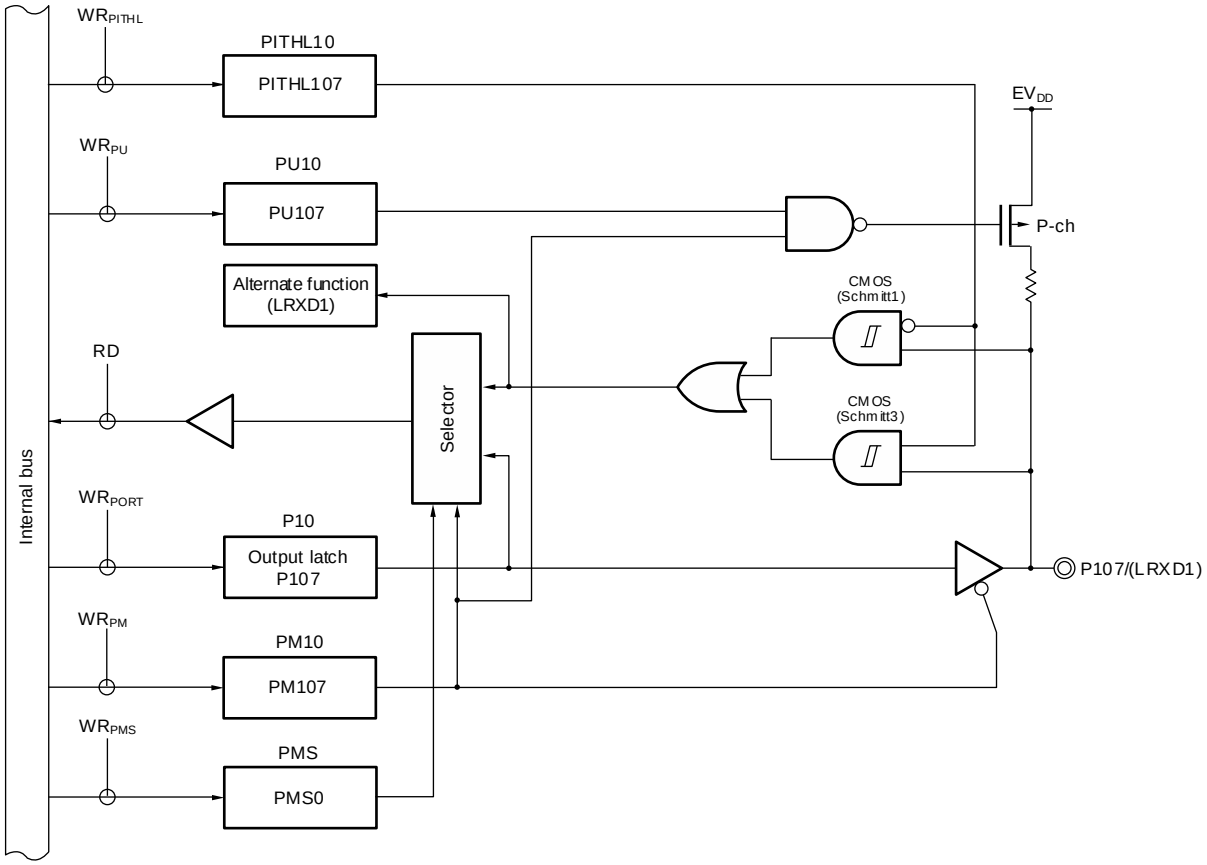
P10: Port register 10
PM10: Port mode register 10
PMS: Port mode select register
PMC10: Port mode control register 10
RD: Read signal
WRxx: Write signal

Figure 4-58. 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 4-59. 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 P120 and 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, clock I/O for serial interfaces (CSI), serial interface data output, slave select input, serial data I/O for LIN, external interrupt request input, and SNOOZE status output.

Reset signal generation sets P120 and P125 to analog input mode and P121 to P124, P126, P127 to input mode.

Table 4-21. Settings of Registers When Using Port 12 (1/2)

Pin Name		PM12x	PIM12x	POM12x	PMC12x	PITHL12x	Alternate Function Setting	Remarks
Name	I/O							
P120	Input	1	–	×	0	0	×	CMOS input (Schmitt1 input)
						1		CMOS input (Schmitt3 input)
	Output	0	–	0	0	×	TRDIOD0 output = 0 Note 1 TO07 output = 0 Note 2 (TO13 output = 0) Note 2 SO01 output = 1 Note 3 (LTXD1 output = 1) Note 5 (SCK10 output = 1) Note 3	CMOS output
				1	0	×		N-ch O.D output
P121	Input	–	–	–	–	–	OSCSEL bit of CMC register = 0 or EXCLK bit = 1	
P122	Input	–	–	–	–	–	OSCSEL bit of CMC register = 0	
P123	Input	–	–	–	–	–	OSCSELS bit of CMC register = 0 or EXCLKS bit = 1 or SELLOSC bit of CKSEL register = 1	
P124	Input	–	–	–	–	–	OSCSELS bit of CMC register = 0 or SELLOSC bit of CKSEL register = 1	

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

Table 4-21. Settings of Registers When Using Port 12 (2/2)

Pin Name		PM12x	PIM12x	POM12x	PMC12x	PITHL12x	Alternate Function Setting	Remarks
Name	I/O							
P125	Input	1	0	–	0	0	×	CMOS input (Schmitt1 input)
						1		CMOS input (Schmitt3 input)
			1	–	0	0	×	TTL input
	Output	0	×	–	0	×	TRDIOB0 output = 0 Note 1 TO03 output = 0 Note 2 SNZOUT1 output = 0 Note 4	
P126	Input	1	–	–	–	–	×	
	Output	0	–	–	–	–	(TO01 output = 0) Note2	
P127	Input	1	–	–	–	–	×	
	Output	0	–	–	–	–	(TO03 output = 0) Note2	

Important To use the port 12 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

Notes 1. When a pin sharing a timer RDe function is to be used as a general-purpose port pin, the target bit for TRDIOij pin output control in the timer RDe output master enable register 1 (TRDOER1) must have the same setting as its initial value (i = A, B, C, D, j = 0, 1). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).

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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR1, 3, 9).

3. When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn and CKOmn bits 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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).

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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR6).

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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).

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.

Remarks ×: Don't care
PM12x: Bit of port mode register 12
PIM12x: Bit of port input mode register 12
POM12x: Bit of port output mode register 12
PMC12x: Bit of port mode control register 12
PITHL12x: Bit of port input threshold control register 12

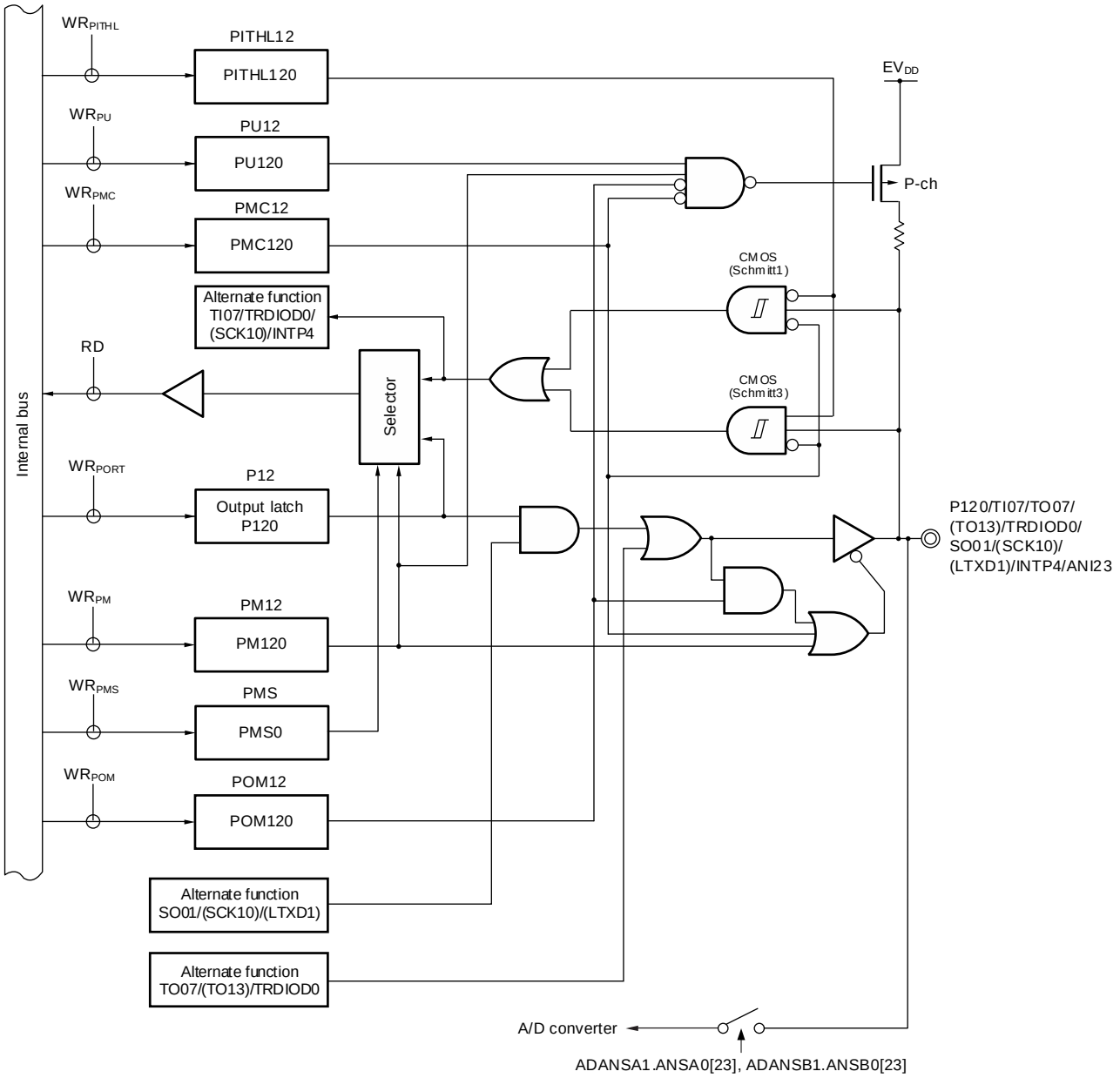
Table 4-22. Setting Functions of P125/ANI22 and P120/ANI23 Pins

PMC12 Register	PM12 Register	ADANSA1, ADANSB1 Registers	P125/ANI22 and P120/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 P125/ANI22 and P120/ANI23 to analog input mode.

Figure 4-60 to **Figure 4-64** show block diagrams of port 12 for 100-pin products.

Figure 4-60. Block Diagram of P120

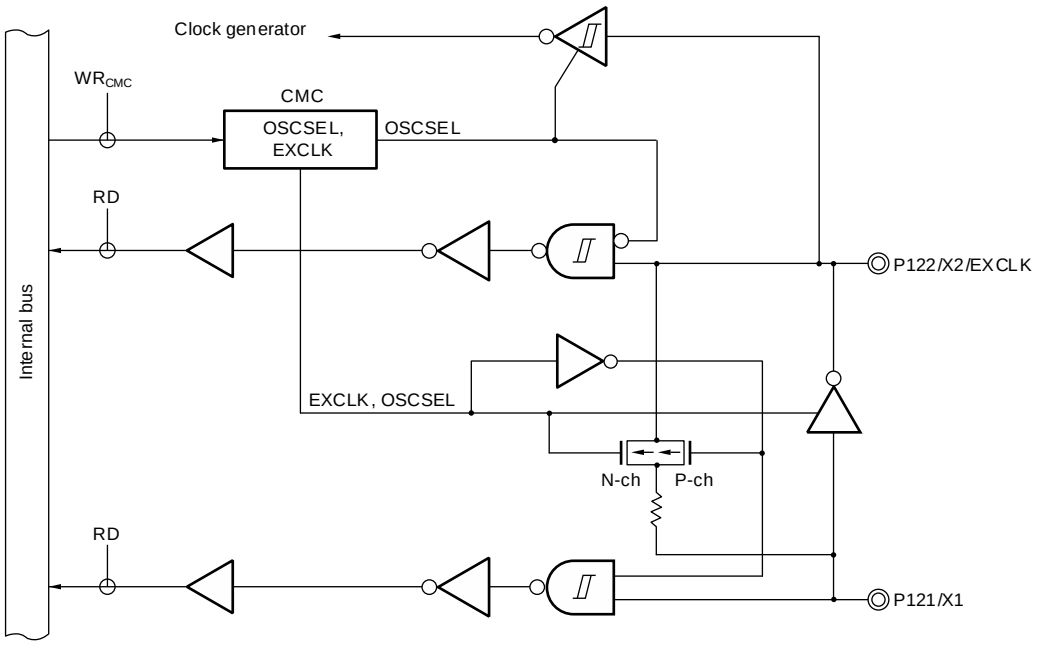


P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
POM12: Port output mode register 12
PMC12: Port mode control register 12
PITHL12: Port input threshold control register 12
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

(**Caution** is listed on the next page.)

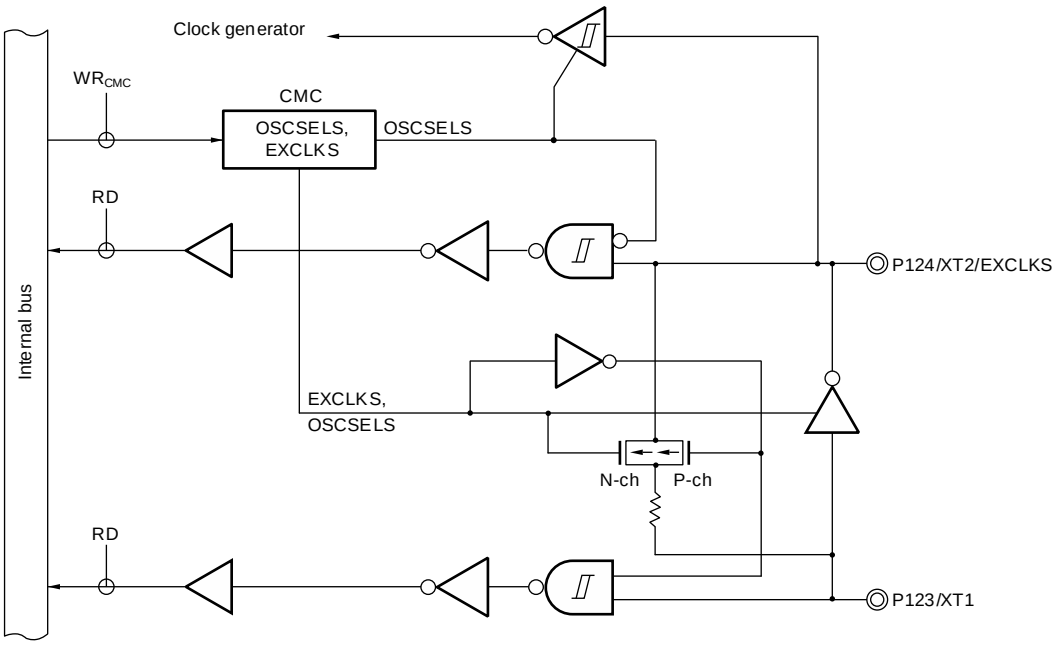
Caution The input buffer is enabled even if the pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the pin when the voltage level on this pin is intermediate.

Figure 4-61. Block Diagram of P121 and P122



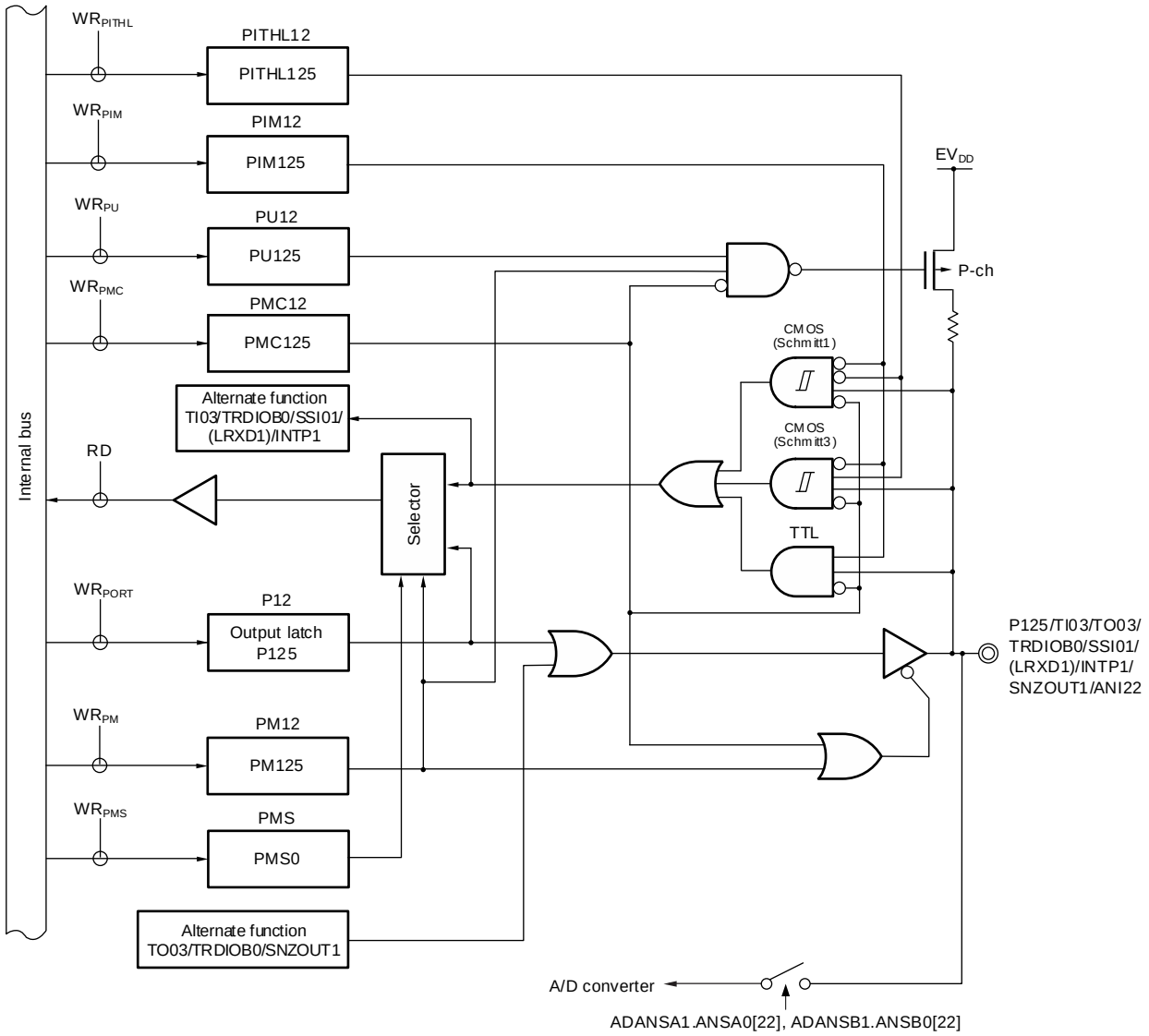
CMC: Clock operation mode control register
RD: Read signal
WRxx: Write signal

Figure 4-62. Block Diagram of P123 and P124



CMC: Clock operation mode control register
RD: Read signal
WRxx: Write signal

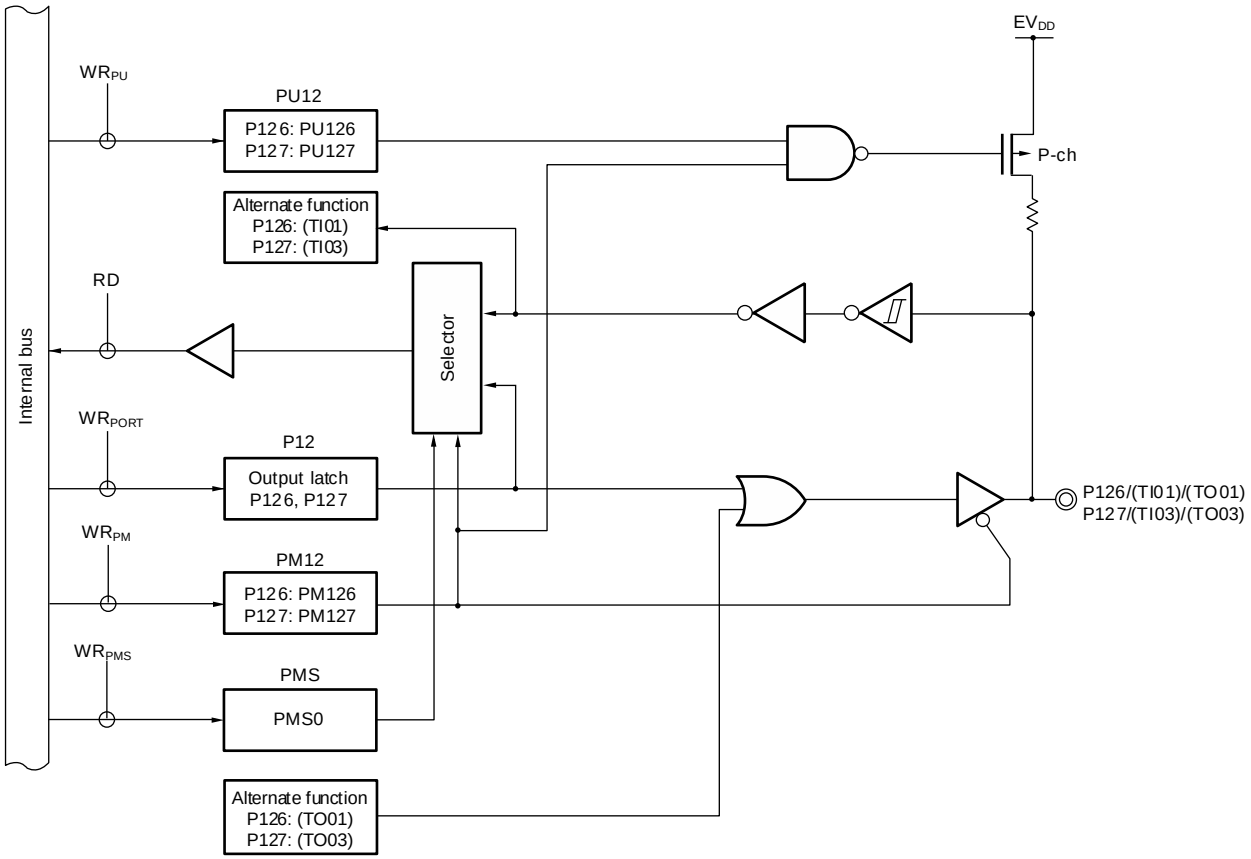
Figure 4-63. Block Diagram of P125



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
RD: Read signal
WRxx: Write signal

Caution When the pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMm) and is driven high, a through current may flow through the pin due to the configuration of the TTL input buffer. When transitioning to standby mode, set the pins to low to reduce current consumption.

Figure 4-64. 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.

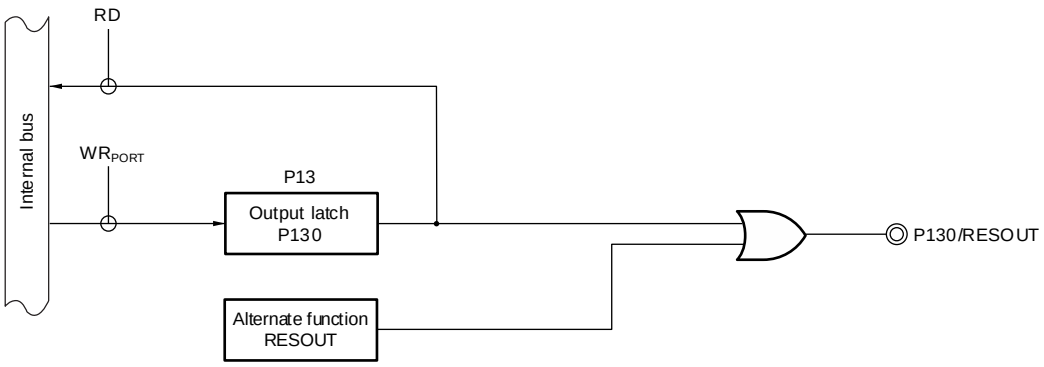
The RESOUT output can be set by an option byte.

Table 4-23. Settings of Registers When Using Port 13

Pin Name		Alternate Function Setting
Name	I/O	
P130	Output	RESOUT
P137	Input	×

Figure 4-65 and **Figure 4-66** show block diagrams of port 13 for 100-pin products.

Figure 4-65. 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.

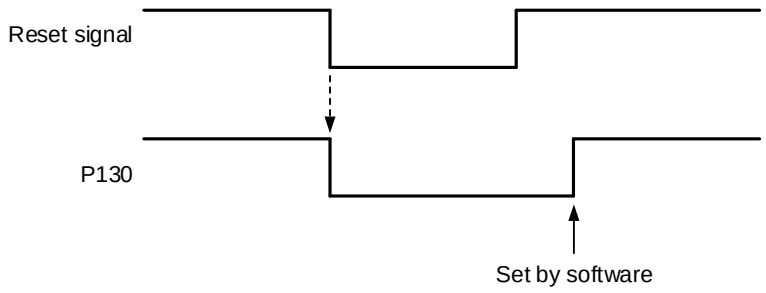
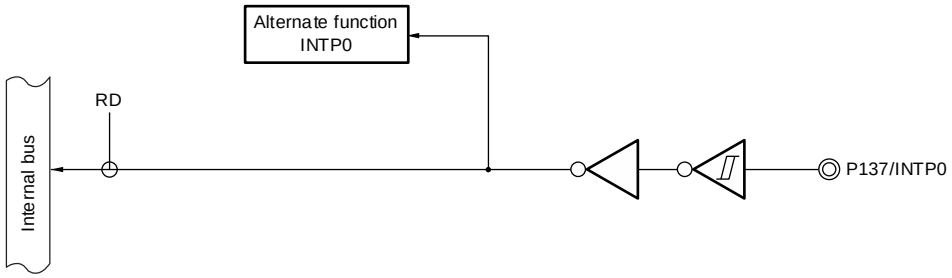


Figure 4-66. Block Diagram of P137



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, and timer RDe counter clear trigger input.

Reset signal generation sets P140 to input mode.

Table 4-24. Settings of Registers When Using Port 14

Pin Name		PM14x	Alternate Function Setting
Name	I/O		
P140	Input	1	×
	Output	0	PCLBUZ0 output = 0 Note

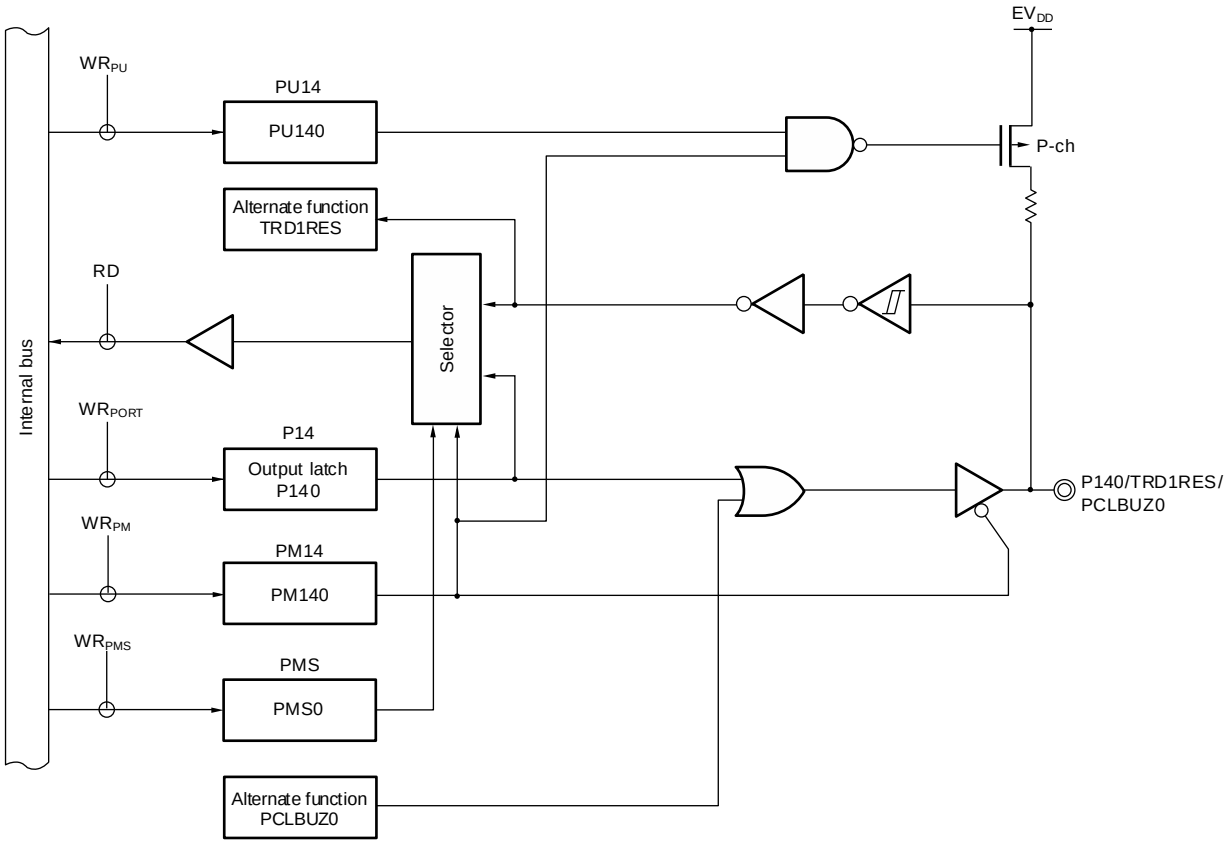
Important To use the port 14 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

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.

Remarks ×: Don't care
PM14x: Bit of port mode register 14

Figure 4-67 shows a block diagram of port 14 for 100-pin products.

Figure 4-67. 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 to P154 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, LIN serial data I/O, 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	Remarks
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	×	(SCK11 output = 1) Note 1	
P154	Input	1	0	×	CMOS input (Schmitt1 input)
			1	×	CMOS input (Schmitt3 input)
	Output	0	–	(SNZOUT7 output = 0) Note 2	
P155	Input	1	–	×	
	Output	0	–	LTXD2 output = 1 Note 3 (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	

Important To use the port 15 as a general-purpose port, set the alternate pin function output to the level indicated by the Alternate Function Setting When Using Pin as Port.

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

Notes 1. When a pin sharing the serial array unit function is to be used as a general-purpose port pin, the SOmn and CKOmn bits 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). Alternatively, assign the corresponding function to another pin by peripheral I/O redirection registers (PIOR4, 9).

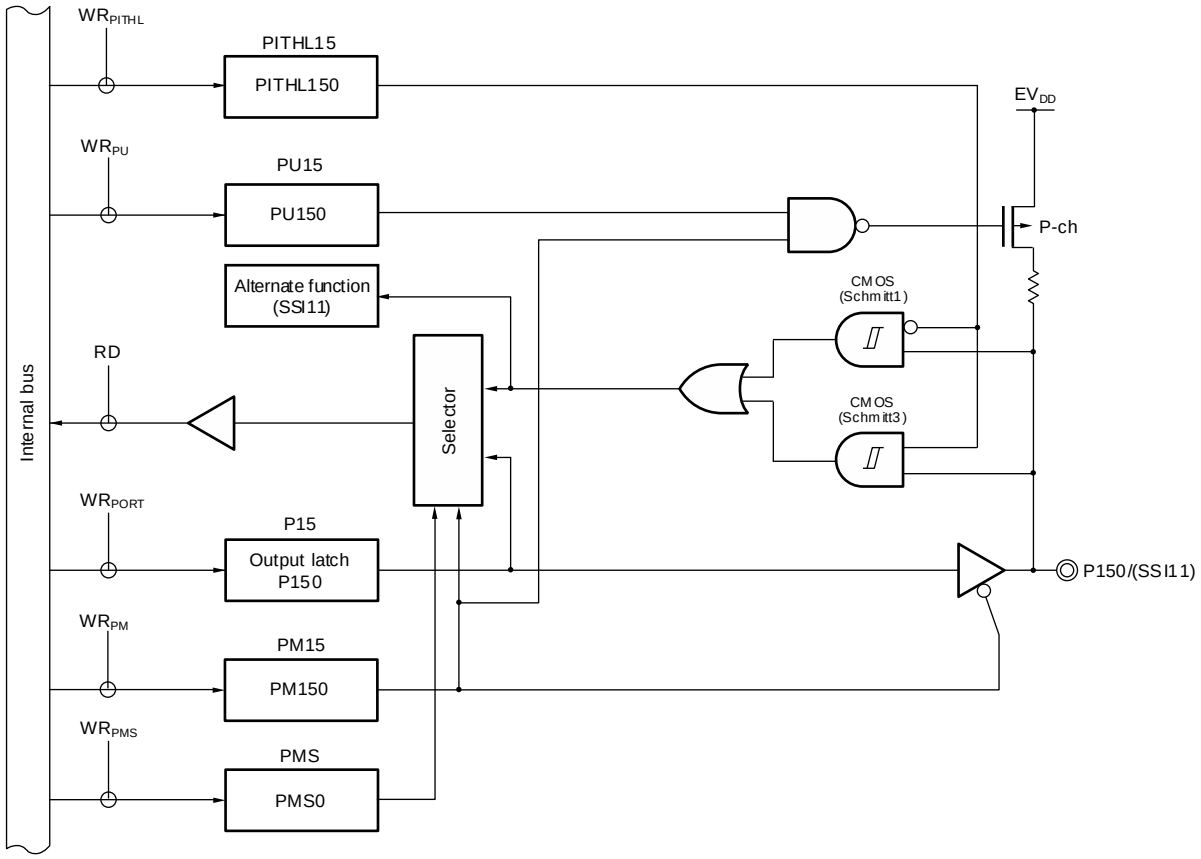
2. When a pin sharing the SNOOZE status output function is to be used as a general-purpose port pin, the OUTEN0 to OUTEN7 bits of the SNOOZE status output control registers 0, 1, 2, 3 (PSNZCNT0, 1, 2, 3) must have the same setting as its initial value. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR6).

3. 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. Alternatively, assign the corresponding function to another pin by peripheral I/O redirection register (PIOR9).

Remarks ×: Don't care
PM15x: Bit of port mode register 15
PITHL15x: Bit of port input threshold control register 15

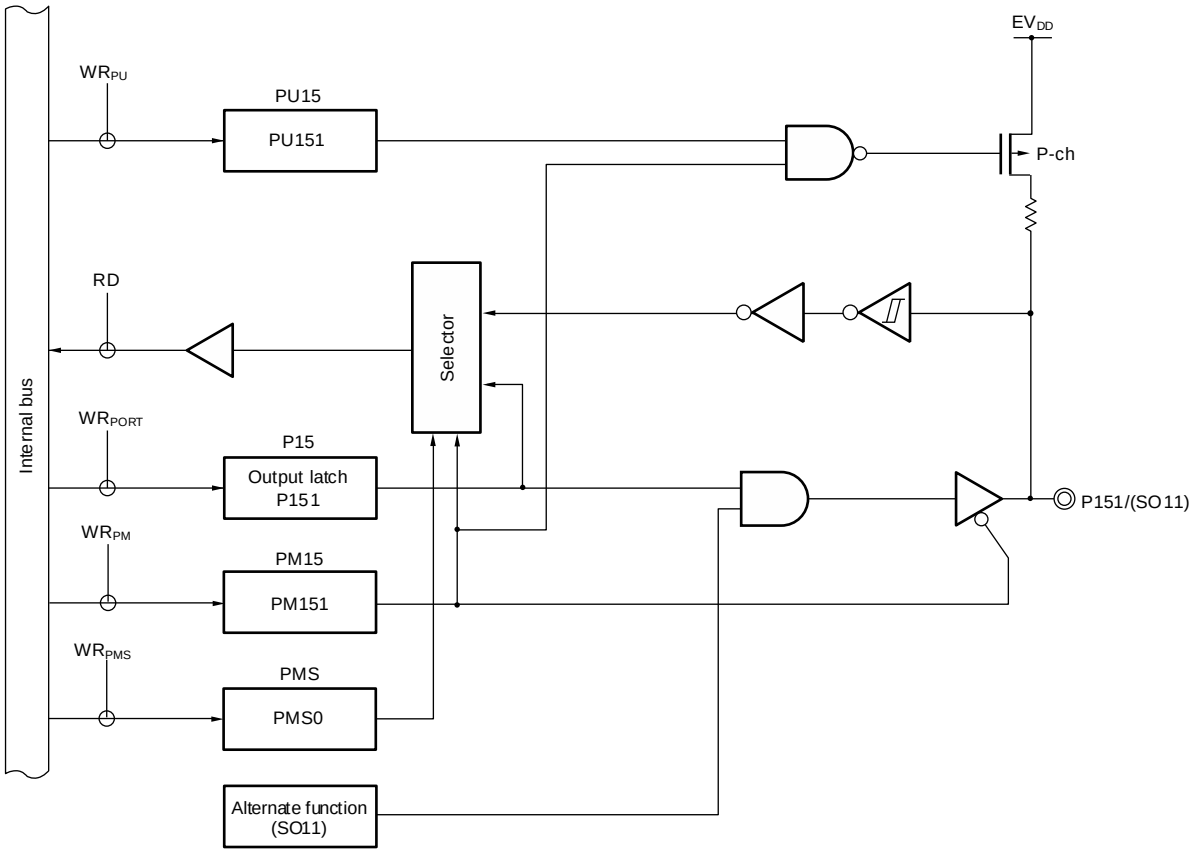
Figure 4-68 to **Figure 4-72** show block diagrams of port 15 for 100-pin products.

Figure 4-68. Block Diagram of P150



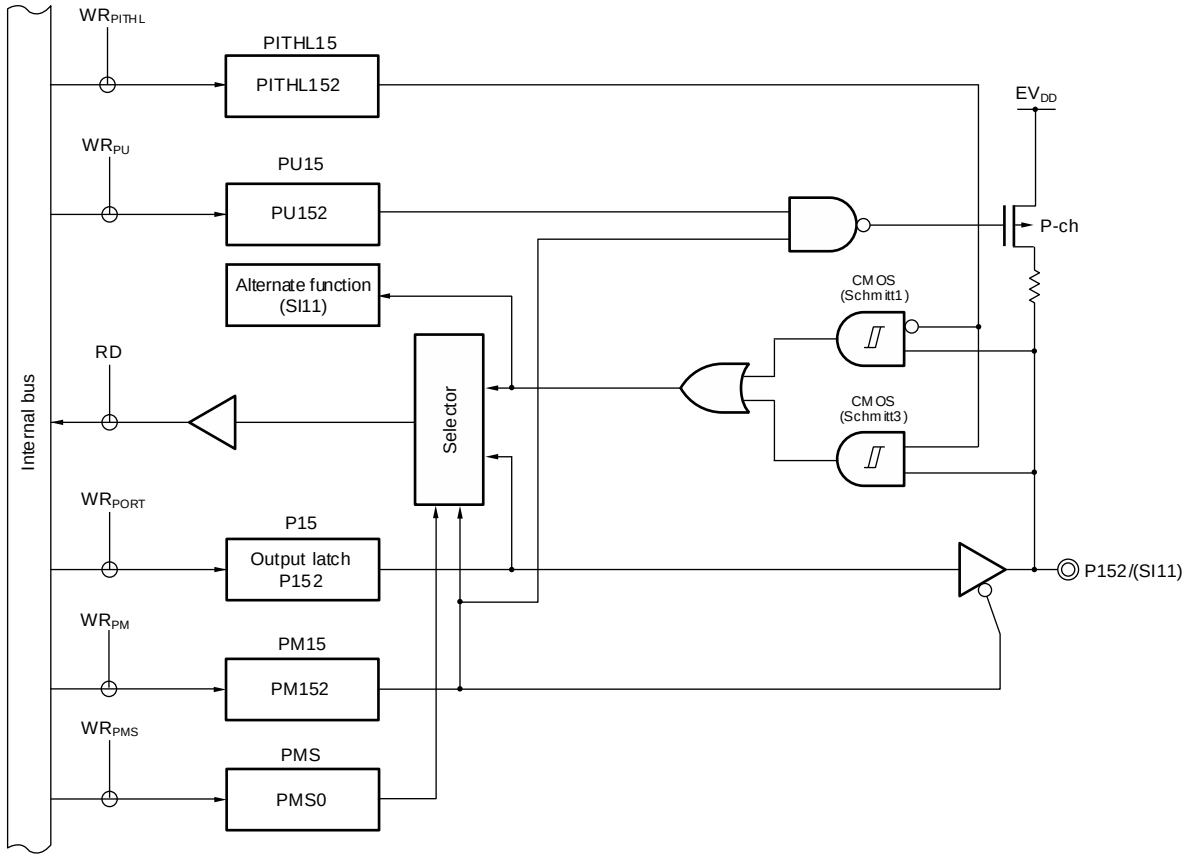
P15: Port register 15
PU15: Pull-up resistor option register 15
PM15: Port mode register 15
PMS: Port mode select register
PITHL15: Port input threshold control register 15
RD: Read signal
WRxx: Write signal

Figure 4-69. Block Diagram of P151



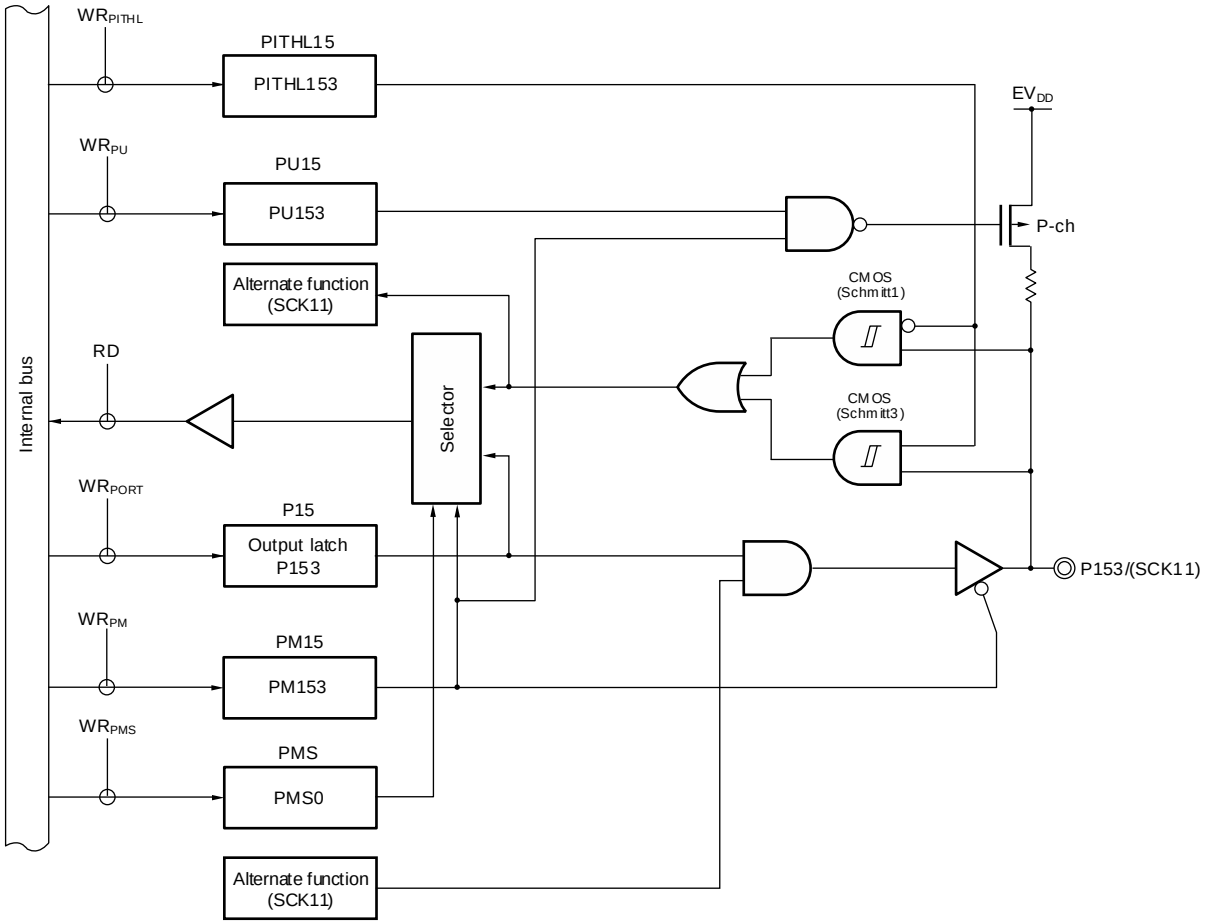
P15: Port register 15
PU15: Pull-up resistor option register 15
PM15: Port mode register 15
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

Figure 4-70. Block Diagram of P152



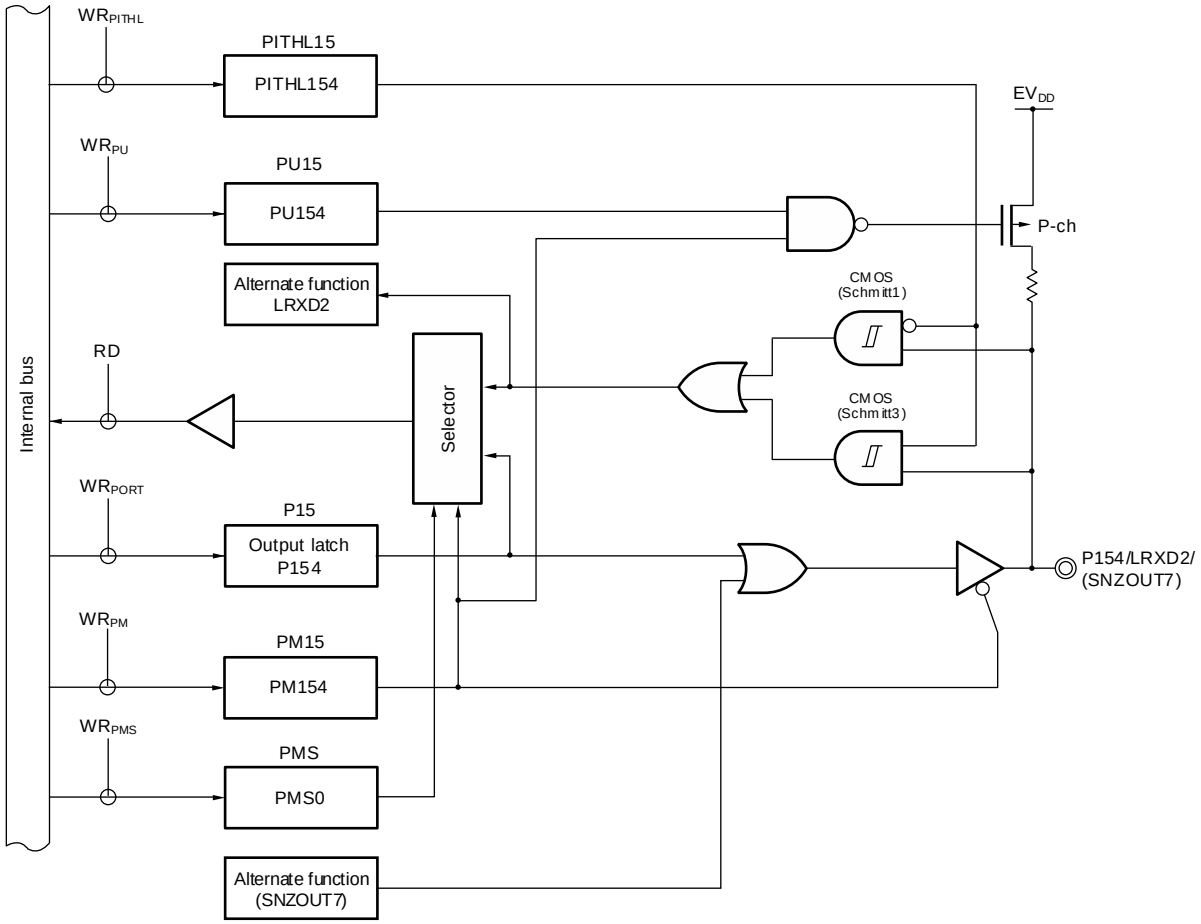
P15: Port register 15
PU15: Pull-up resistor option register 15
PM15: Port mode register 15
PMS: Port mode select register
PITHL15: Port input threshold control register 15
RD: Read signal
WRxx: Write signal

Figure 4-71. Block Diagram of P153



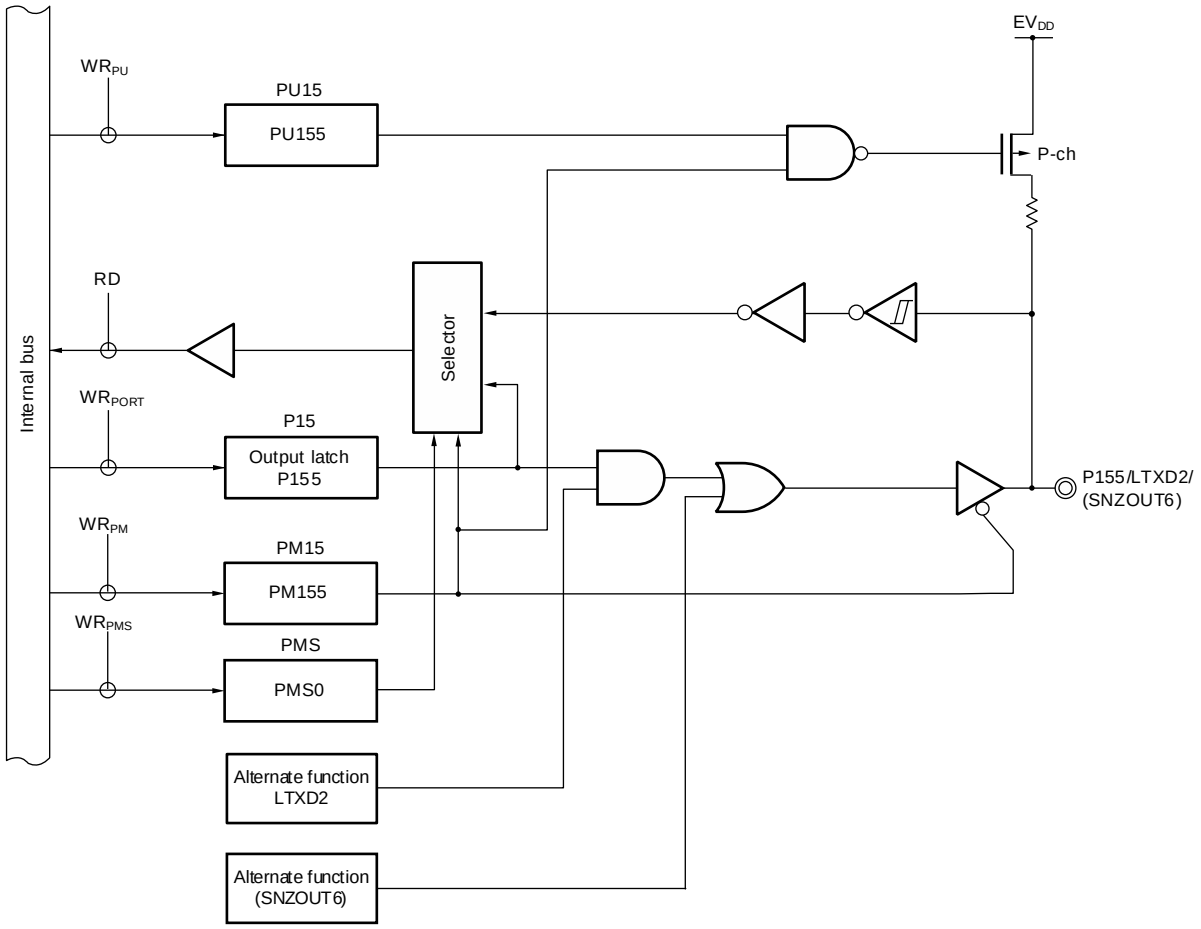
P15: Port register 15
PU15: Pull-up resistor option register 15
PM15: Port mode register 15
PMS: Port mode select register
PITHL15: Port input threshold control register 15
RD: Read signal
WRxx: Write signal

Figure 4-72. Block Diagram of P154



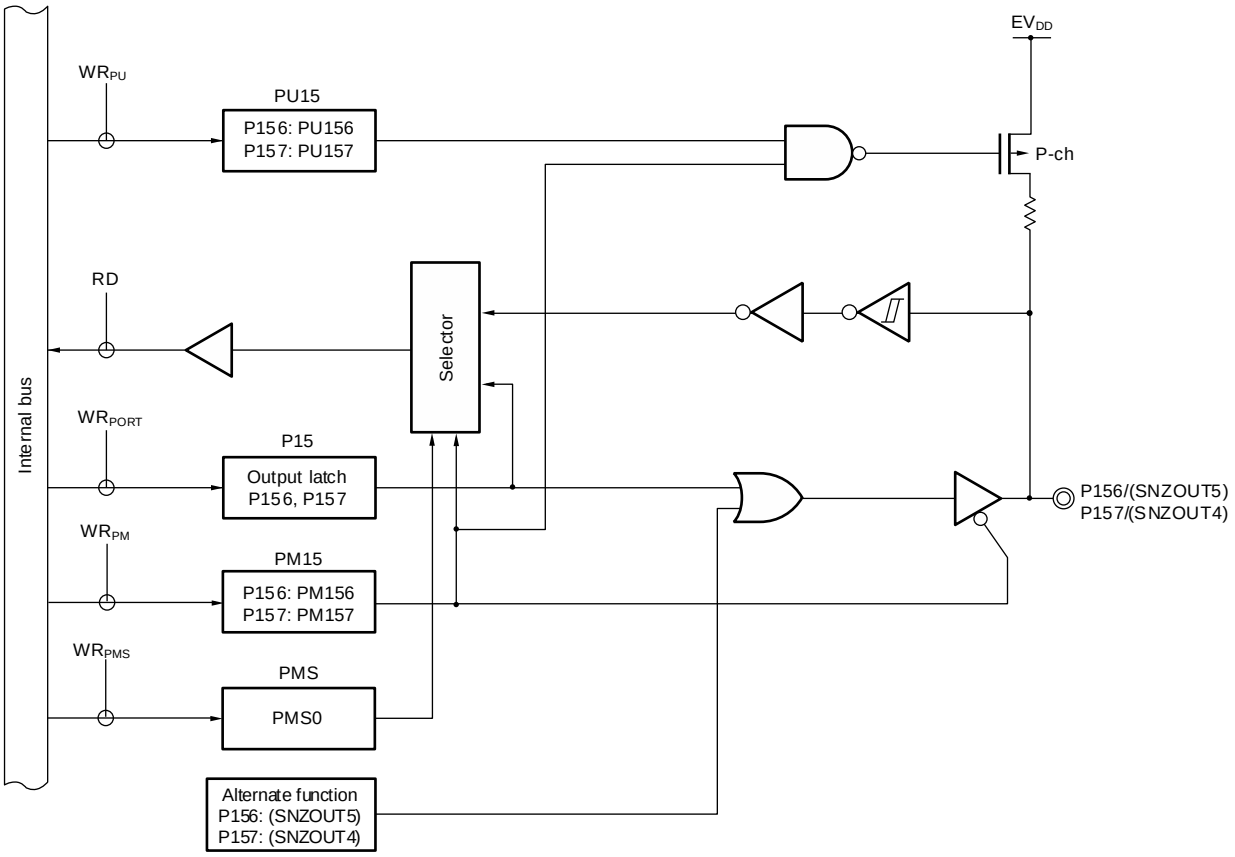
P15: Port register 15
PU15: Pull-up resistor option register 15
PM15: Port mode register 15
PMS: Port mode select register
PITHL15: Port input threshold control register 15
RD: Read signal
WRxx: Write signal

Figure 4-73. Block Diagram of P155



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 4-74. Block Diagram of P156 and 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

Table 4-26. Port Register Configuration (1/2)

Address	Register Name	Symbol	After Reset	Access Size
FFF00H	Port register 0	P0	00H Note 1	1, 8
FFF01H	Port register 1	P1	00H Note 1	1, 8
FFF03H	Port register 3	P3	00H Note 1	1, 8
FFF04H	Port register 4	P4	00H Note 1	1, 8
FFF05H	Port register 5	P5	00H Note 1	1, 8
FFF06H	Port register 6	P6	00H Note 1	1, 8
FFF07H	Port register 7	P7	00H Note 1	1, 8
FFF08H	Port register 8	P8	00H Note 1	1, 8
FFF09H	Port register 9	P9	00H Note 1	1, 8
FFF0AH	Port register 10	P10	00H Note 1	1, 8
FFF0CH	Port register 12	P12	Undefined Note 2	1, 8
FFF0DH	Port register 13	P13	Undefined Note 3	1, 8
FFF0EH	Port register 14	P14	00H Note 1	1, 8
FFF0FH	Port register 15	P15	00H Note 1	1, 8
FFF20H	Port mode register 0	PM0	FFH	1, 8
FFF21H	Port mode register 1	PM1	FFH	1, 8
FFF23H	Port mode register 3	PM3	FFH	1, 8
FFF24H	Port mode register 4	PM4	FFH	1, 8
FFF25H	Port mode register 5	PM5	FFH	1, 8
FFF26H	Port mode register 6	PM6	FFH	1, 8
FFF27H	Port mode register 7	PM7	FFH	1, 8
FFF28H	Port mode register 8	PM8	FFH	1, 8
FFF29H	Port mode register 9	PM9	FFH	1, 8
FFF2AH	Port mode register 10	PM10	FFH	1, 8
FFF2CH	Port mode register 12	PM12	FFH	1, 8
FFF2EH	Port mode register 14	PM14	FFH	1, 8
FFF2FH	Port mode register 15	PM15	FFH	1, 8
F0016H	Peripheral I/O redirection register 0	PIOR0	00H	8
F0017H	Peripheral I/O redirection register 1	PIOR1	00H	8
F0018H	Peripheral I/O redirection register 2	PIOR2	00H	8
F0019H	Peripheral I/O redirection register 3	PIOR3	00H	8
F001AH	Peripheral I/O redirection register 4	PIOR4	00H	8
F001BH	Peripheral I/O redirection register 5	PIOR5	00H	8
F001CH	Peripheral I/O redirection register 6	PIOR6	00H	8
F001DH	Peripheral I/O redirection register 7	PIOR7	00H	8
F001EH	Peripheral I/O redirection register 8	PIOR8	00H	8
F001FH	Peripheral I/O redirection register 9	PIOR9	00H	8

Notes 1. The value of output latch.

2. The values of bits P120 and P125 to P127 is 0 (output latch).

3. P130 bit depends on the setting of User Option Byte (000C2H/040C2H).

Table 4-26. Port Register Configuration (2/2)

Address	Register Name	Symbol	After Reset	Access Size
F0021H	Port input threshold control register 1	PITHL1	00H	1, 8
F0023H	Port input threshold control register 3	PITHL3	00H	1, 8
F0024H	Port input threshold control register 4	PITHL4	00H	1, 8
F0025H	Port input threshold control register 5	PITHL5	00H	1, 8
F0026H	Port input threshold control register 6	PITHL6	00H	1, 8
F0027H	Port input threshold control register 7	PITHL7	00H	1, 8
F002AH	Port input threshold control register 10	PITHL10	00H	1, 8
F002CH	Port input threshold control register 12	PITHL12	00H	1, 8
F002FH	Port input threshold control register 15	PITHL15	00H	1, 8
F0030H	Pull-up resistor option register 0	PU0	00H	1, 8
F0031H	Pull-up resistor option register 1	PU1	00H	1, 8
F0033H	Pull-up resistor option register 3	PU3	00H	1, 8
F0034H	Pull-up resistor option register 4	PU4	01H	1, 8
F0035H	Pull-up resistor option register 5	PU5	00H	1, 8
F0036H	Pull-up resistor option register 6	PU6	00H	1, 8
F0037H	Pull-up resistor option register 7	PU7	00H	1, 8
F003AH	Pull-up resistor option register 10	PU10	00H	1, 8
F003CH	Pull-up resistor option register 12	PU12	00H	1, 8
F003EH	Pull-up resistor option register 14	PU14	00H	1, 8
F003FH	Pull-up resistor option register 15	PU15	00H	1, 8
F0041H	Port input mode register 1	PIM1	00H	1, 8
F0043H	Port input mode register 3	PIM3	00H	1, 8
F0045H	Port input mode register 5	PIM5	00H	1, 8
F0046H	Port input mode register 6	PIM6	00H	1, 8
F0047H	Port input mode register 7	PIM7	00H	1, 8
F004CH	Port input mode register 12	PIM12	00H	1, 8
F0051H	Port output mode register 1	POM1	00H	1, 8
F0053H	Port output mode register 3	POM3	00H	1, 8
F0056H	Port output mode register 6	POM6	00H	1, 8
F0057H	Port output mode register 7	POM7	00H	1, 8
F005CH	Port output mode register 12	POM12	00H	1, 8
F0067H	Port mode control register 7	PMC7	FFH	1, 8
F0068H	Port mode control register 8	PMC8	FFH	1, 8
F0069H	Port mode control register 9	PMC9	FFH	1, 8
F006AH	Port mode control register 10	PMC10	FFH	1, 8
F006CH	Port mode control register 12	PMC12	FFH	1, 8
F0077H	Port mode select register	PMS	00H	1, 8
F0211H	Touch sensor port mode control register 1	TSPMC1	00H	1, 8
F0215H	Touch sensor port mode control register 5	TSPMC5	00H	1, 8
F0216H	Touch sensor port mode control register 6	TSPMC6	00H	1, 8
F0217H	Touch sensor port mode control register 7	TSPMC7	00H	1, 8
F0220H	Port output slew rate register	PSRSEL	00H	1, 8
F0222H	SNOOZE status output control register 0	PSNZCNT0	00H	1, 8
F0223H	SNOOZE status output control register 1	PSNZCNT1	00H	1, 8
F0224H	SNOOZE status output control register 2	PSNZCNT2	00H	1, 8
F0225H	SNOOZE status output control register 3	PSNZCNT3	00H	1, 8

Table 4-27. Port Configuration of RL78/F22 Products (48-pin products) (1/2)

Port Name	Port Bit	Port Latch	I/O Mode Control	Pull-up Control	Input Type Control	Output Type Control	Operating Mode Control	Touch Sensor Mode Control	Input Threshold Control
PORT0	0	P0.0	PM0.0	PU0.0	–	–	–	–	–
PORT1	0	P1.0	PM1.0	PU1.0	PIM1.0	POM1.0	–	TSPMC1.0	PITHL1.0
	1	P1.1	PM1.1	PU1.1	PIM1.1	POM1.1	–	–	PITHL1.1
	2	P1.2	PM1.2	PU1.2	–	POM1.2	–	TSPMC1.2	–
	3	P1.3	PM1.3	PU1.3	PIM1.3	POM1.3	–	TSPMC1.3	PITHL1.3
	4	P1.4	PM1.4	PU1.4	PIM1.4	POM1.4	–	TSPMC1.4	PITHL1.4
	5	P1.5	PM1.5	PU1.5	–	POM1.5	–	TSPMC1.5	–
	6	P1.6	PM1.6	PU1.6	PIM1.6	POM1.6	–	TSPMC1.6	PITHL1.6
	7	P1.7	PM1.7	PU1.7	PIM1.7	POM1.7	–	TSPMC1.7	PITHL1.7
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	–	POM3.2	–	–	–
PORT4	0	P4.0	PM4.0	PU4.0	–	–	–	–	–
	1	P4.1	PM4.1	PU4.1	–	–	–	–	PITHL4.1
PORT6	0	P6.0	PM6.0	PU6.0	–	POM6.0	–	TSPMC6.0	PITHL6.0
	1	P6.1	PM6.1	PU6.1	–	POM6.1	–	TSPMC6.1	PITHL6.1
	2	P6.2	PM6.2	PU6.2	PIM6.2	POM6.2	–	TSPMC6.2	PITHL6.2
	3	P6.3	PM6.3	PU6.3	PIM6.3	POM6.3	–	TSPMC6.3	PITHL6.3
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	TSPMC7.2	–
	3	P7.3	PM7.3	PU7.3	PIM7.3	–	PMC7.3	TSPMC7.3	PITHL7.3
PORT8	0	P8.0	PM8.0	–	–	–	PMC8.0	–	–
	1	P8.1	PM8.1	–	–	–	PMC8.1	–	–
	2	P8.2	PM8.2	–	–	–	PMC8.2	–	–
	3	P8.3	PM8.3	–	–	–	PMC8.3	–	–
	4	P8.4	PM8.4	–	–	–	PMC8.4	–	–
	5	P8.5	PM8.5	–	–	–	PMC8.5	–	–
	6	P8.6	PM8.6	–	–	–	PMC8.6	–	–
	7	P8.7	PM8.7	–	–	–	PMC8.7	–	–
PORT9	0	P9.0	PM9.0	–	–	–	PMC9.0	–	–
	1	P9.1	PM9.1	–	–	–	PMC9.1	–	–
	2	P9.2	PM9.2	–	–	–	PMC9.2	–	–

Remark – : Not provided.

Table 4-27. Port Configuration of RL78/F22 Products (48-pin products) (2/2)

Port Name	Port Bit	Port Latch	I/O Mode Control	Pull-up Control	Input Type Control	Output Type Control	Operating Mode Control	Touch Sensor Mode Control	Input Threshold Control
PORT12	0	P12.0	PM12.0	PU12.0	–	POM12.0	PMC12.0	–	PITHL12.0
	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
PORT13	0	P13.0	–	–	–	–	–	–	–
	7	P13.7	–	–	–	–	–	–	–
PORT14	0	P14.0	PM14.0	PU14.0	–	–	–	–	–

Remark – : Not provided.

Table 4-28. Port Configuration of RL78/F25 Products (100-pin products) (1/3)

Port Name	Port Bit	Port Latch	I/O Mode Control	Pull-up Control	Input Type Control	Output Type Control	Operating Mode Control	Touch Sensor 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	–	–	–	–	–
PORT1	0	P1.0	PM1.0	PU1.0	PIM1.0	POM1.0	–	TSPMC1.0	PITHL1.0
	1	P1.1	PM1.1	PU1.1	PIM1.1	POM1.1	–	TSPMC1.1	PITHL1.1
	2	P1.2	PM1.2	PU1.2	–	POM1.2	–	TSPMC1.2	–
	3	P1.3	PM1.3	PU1.3	PIM1.3	POM1.3	–	TSPMC1.3	PITHL1.3
	4	P1.4	PM1.4	PU1.4	PIM1.4	POM1.4	–	TSPMC1.4	PITHL1.4
	5	P1.5	PM1.5	PU1.5	–	POM1.5	–	TSPMC1.5	–
	6	P1.6	PM1.6	PU1.6	PIM1.6	POM1.6	–	TSPMC1.6	PITHL1.6
	7	P1.7	PM1.7	PU1.7	PIM1.7	POM1.7	–	TSPMC1.7	PITHL1.7
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	–	POM3.2	–	–	–
PORT4	0	P4.0	PM4.0	PU4.0	–	–	–	–	–
	1	P4.1	PM4.1	PU4.1	–	–	–	–	PITHL4.1
	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	–	–	–	–	
PORT5	0	P5.0	PM5.0	PU5.0	–	–	–	TSPMC5.0	PITHL5.0
	1	P5.1	PM5.1	PU5.1	–	–	–	TSPMC5.1	–
	2	P5.2	PM5.2	PU5.2	–	–	–	TSPMC5.2	PITHL5.2
	3	P5.3	PM5.3	PU5.3	–	–	–	TSPMC5.3	PITHL5.3
	4	P5.4	PM5.4	PU5.4	PIM5.4	–	–	TSPMC5.4	PITHL5.4
	5	P5.5	PM5.5	PU5.5	–	–	–	TSPMC5.5	–
	6	P5.6	PM5.6	PU5.6	–	–	–	TSPMC5.6	–
	7	P5.7	PM5.7	PU5.7	–	–	–	TSPMC5.7	–

Remark – : Not provided.

Table 4-28. Port Configuration of RL78/F25 Products (100-pin products) (2/3)

Port Name	Port Bit	Port Latch	I/O Mode Control	Pull-up Control	Input Type Control	Output Type Control	Operating Mode Control	Touch Sensor Mode Control	Input Threshold Control
PORT6	0	P6.0	PM6.0	PU6.0	–	POM6.0	–	TSPMC6.0	PITHL6.0
	1	P6.1	PM6.1	PU6.1	–	POM6.1	–	TSPMC6.1	PITHL6.1
	2	P6.2	PM6.2	PU6.2	PIM6.2	POM6.2	–	TSPMC6.2	PITHL6.2
	3	P6.3	PM6.3	PU6.3	PIM6.3	POM6.3	–	TSPMC6.3	PITHL6.3
	4	P6.4	PM6.4	PU6.4	–	–	–	TSPMC6.4	–
	5	P6.5	PM6.5	PU6.5	–	–	–	TSPMC6.5	–
	6	P6.6	PM6.6	PU6.6	–	–	–	TSPMC6.6	–
	7	P6.7	PM6.7	PU6.7	–	–	–	TSPMC6.7	–
PORT7	0	P7.0	PM7.0	PU7.0	PIM7.0	POM7.0	PMC7.0	TSPMC7.0	PITHL7.0
	1	P7.1	PM7.1	PU7.1	PIM7.1	POM7.1	PMC7.1	TSPMC7.1	PITHL7.1
	2	P7.2	PM7.2	PU7.2	–	POM7.2	PMC7.2	TSPMC7.2	–
	3	P7.3	PM7.3	PU7.3	PIM7.3	–	PMC7.3	TSPMC7.3	PITHL7.3
	4	P7.4	PM7.4	PU7.4	–	–	PMC7.4	TSPMC7.4	–
	5	P7.5	PM7.5	PU7.5	–	–	–	TSPMC7.5	PITHL7.5
	6	P7.6	PM7.6	PU7.6	–	–	–	TSPMC7.6	PITHL7.6
	7	P7.7	PM7.7	PU7.7	–	–	–	TSPMC7.7	PITHL7.7
PORT8	0	P8.0	PM8.0	–	–	–	PMC8.0	–	–
	1	P8.1	PM8.1	–	–	–	PMC8.1	–	–
	2	P8.2	PM8.2	–	–	–	PMC8.2	–	–
	3	P8.3	PM8.3	–	–	–	PMC8.3	–	–
	4	P8.4	PM8.4	–	–	–	PMC8.4	–	–
	5	P8.5	PM8.5	–	–	–	PMC8.5	–	–
	6	P8.6	PM8.6	–	–	–	PMC8.6	–	–
	7	P8.7	PM8.7	–	–	–	PMC8.7	–	–
PORT9	0	P9.0	PM9.0	–	–	–	PMC9.0	–	–
	1	P9.1	PM9.1	–	–	–	PMC9.1	–	–
	2	P9.2	PM9.2	–	–	–	PMC9.2	–	–
	3	P9.3	PM9.3	–	–	–	PMC9.3	–	–
	4	P9.4	PM9.4	–	–	–	PMC9.4	–	–
	5	P9.5	PM9.5	–	–	–	PMC9.5	–	–
	6	P9.6	PM9.6	–	–	–	PMC9.6	–	–
	7	P9.7	PM9.7	–	–	–	PMC9.7	–	–

Remark – : Not provided.

Table 4-28. Port Configuration of RL78/F25 Products (100-pin products) (3/3)

Port Name	Port Bit	Port Latch	I/O Mode Control	Pull-up Control	Input Type Control	Output Type Control	Operating Mode Control	Touch Sensor Mode Control	Input Threshold Control
PORT10	0	P10.0	PM10.0	–	–	–	PMC10.0	–	–
	1	P10.1	PM10.1	–	–	–	PMC10.1	–	–
	2	P10.2	PM10.2	–	–	–	PMC10.2	–	–
	3	P10.3	PM10.3	–	–	–	PMC10.3	–	–
	4	P10.4	PM10.4	–	–	–	PMC10.4	–	–
	5	P10.5	PM10.5	–	–	–	PMC10.5	–	–
	6	P10.6	PM10.6	PU10.6	–	–	–	–	–
	7	P10.7	PM10.7	PU10.7	–	–	–	–	PITHL10.7
PORT12	0	P12.0	PM12.0	PU12.0	–	POM12.0	PMC12.0	–	PITHL12.0
	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	–	–	–	–	–	–	–
	7	P13.7	–	–	–	–	–	–	–
PORT14	0	P14.0	PM14.0	PU14.0	–	–	–	–	–
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	–	–	–	–	PITHL15.4
	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 (PMm)

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 Setting of Port Mode Register and Output Latch When Using Alternate Function**.

Figure 4-75. 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	1	1	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 (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 (Pm)

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 PMm is set to 0 (output mode), the pin level can be read from Pm by setting PMS.0 (port mode select) to 1.

Reset signal generation clears these registers to 00H.

Note When 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, P82 to P85 are set up as analog inputs of the comparator, or P10 to P17, P50 to P57, P60 to P67, P70 to P77 are set up as capacitance measurement function of capacitive sensing unit, if a port is read while in the input mode, 0 is always returned, not the pin level.

Figure 4-76. 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	0	0	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 Note 2	R/W Note 1
P13	P137	0	0	0	0	0	0	P130	FFF0DH	Undefined Note 3	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

Notes 1. P121 to P124 and P137 are read-only.

2. The values of bits P120 and P125 to P127 is 0 (output latch).

3. P130 bit depends on the setting of bit 5 (RESOUTB) in the User Option Byte (000C2H/040C2H).
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 (PUm)

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 4-77. Format of Pull-up Resistor Option Register (100-pin products)

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
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 (m = 0, 1, 3 to 7, 10, 12, 14, 15; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Caution Be sure to set bits for pins that are not present to their initial values.

4.3.4 Port Input Mode Registers (PIMm)

These registers set the input buffer 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 4-78. 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 (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 (POMm)

These registers set the output mode 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 4-79. 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
POM3	0	0	0	0	0	POM32	0	0	F0053H	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 (m = 1, 3, 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 (PMCm)

These registers set 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 4-80. 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	PMC73	PMC72	PMC71	PMC70	F0067H	FFH	R/W
PMC8	PMC87	PMC86	PMC85	PMC84	PMC83	PMC82	PMC81	PMC80	F0068H	FFH	R/W
PMC9	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90	F0069H	FFH	R/W
PMC10	1	1	PMC105	PMC104	PMC103	PMC102	PMC101	PMC100	F006AH	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, 8, 9, 10, 12; n = 0 to 7)
0	Digital I/O (alternate function other than analog input)
1	Analog I/O

Cautions 1. Be sure to set bits for pins that are not present to their initial values.

2. Set port pins specified as analog input pins to input mode by using port mode register m (PMm).

3. Set the channel used for A/D conversion to the input mode by using port mode control registers 7, 8, 9, 10, 12 (PMC7, PMC8, PMC9, PMC10, PMC12).

4. Set the channel used for D/A conversion or comparator to the input mode by using port mode register 8 (PM8).

5. Do not set the pin set by the PMC register as digital I/O by D/A converter mode register (DAM) as D/A conversion operation enable.

6. Do not set the pin set by the PMC register as digital I/O by the comparator I/O select register (CMPSEL).

4.3.7 Touch Sensor Port Mode Control Register (TSPMCm)

These registers set digital I/O or capacitance measurement function 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 00H.

Figure 4-81. Format of Touch Sensor Port Mode Control Register (100-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
TSPMC1	TSPMC17	TSPMC16	TSPMC15	TSPMC14	TSPMC13	TSPMC12	TSPMC11	TSPMC10	F0211H	00H	R/W
TSPMC5	TSPMC57	TSPMC56	TSPMC55	TSPMC54	TSPMC53	TSPMC52	TSPMC51	TSPMC50	F0215H	00H	R/W
TSPMC6	TSPMC67	TSPMC66	TSPMC65	TSPMC64	TSPMC63	TSPMC62	TSPMC61	TSPMC60	F0216H	00H	R/W
TSPMC7	TSPMC77	TSPMC76	TSPMC75	TSPMC74	TSPMC73	TSPMC72	TSPMC71	TSPMC70	F0217H	00H	R/W

TSPMCmn	Pmn pin digital I/O or capacitance measurement function selection (m = 1, 5, 6, 7; n = 0 to 7)
0	Digital I/O
1	Capacitance measurement function

Caution Be sure to set bits for pins that are not present to their initial values.

4.3.8 Port Input Threshold Control Registers (PITHLm)

These registers are used to specify the threshold value of the input buffers 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 4-82. 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	PITHL41	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	PITHL120

Address: F002FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PITHL15	0	0	0	PITHL154	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 (m = 1, 3, 5 to 7, 12, 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 4-83. 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-pin and 24-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 4-84. 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	PIOR settings	100-pin		80-pin		64-pin		48-pin		32-pin		24-pin	
			Setting value		Setting value		Setting value		Setting value		Setting value		Setting value	
			0	1	0	1	0	1	0	1	0	1	0	1
PIOR17	TO07	PIOR90=0	P120	P44	P120	P44	P120	–	P120	–	P120	–	P120	–
		PIOR90=1	P120	P63	P120	P63	P120	P63	P120	P63	P120	P63	P120	P63
PIOR16	TO06	–	P14	P02	P14	P02	P14	–	P14	–	P14	–	P14	–
PIOR15	TO05	–	P15	P00	P15	P00	P15	P00	P15	P00	P15	–	P15	–
PIOR14	TO04	–	P13	P01	P13	P01	P13	–	P13	–	P13	–	P13	–
PIOR13	TO03	PIOR90=0	P125	P127	P125	–	P125	–	P125	–	P125	–	–	–
		PIOR90=1	P125	P62	P125	P62	P125	P62	P125	P62	P125	P62	–	P62
PIOR12	TO02	PIOR90=0	P16	P67	P16	P67	P16	–	P16	–	P16	–	P16	–
		PIOR90=1	P16	P61	P16	P61	P16	P61	P16	P61	P16	P61	P16	–
PIOR11	TO01	PIOR90=0	P30	P126	P30	P126	P30	–	P30	–	P30	–	P30	–
		PIOR90=1	P30	P60	P30	P60	P30	P60	P30	P60	P30	P60	P30	–
PIOR10	TO00	–	P17	P66	P17	P66	P17	–	P17	–	P17	–	P17	–

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 4-85. 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	P57
PIOR26	TI16	P32	P65	P32	P65
PIOR25	TI15	P70	P56	P70	P56
PIOR24	TI14	P31	P64	P31	P64
PIOR23	TI13	P10	P55	P10	P55
PIOR22	TI12	P11	P46	P11	P46
PIOR21	TI11	P12	P54	P12	P54
PIOR20	TI10	P41	P45	P41	P45

Caution The 64-, 48-, 32- and 24-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 4-86. 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	PIOR Settings	100-pin		80-pin		64-pin		48-pin Note		32-pin		24-pin	
			Setting value		Setting value		Setting value		Setting value		Setting value		Setting value	
			0	1	0	1	0	1	0	1	0	1	0	1
PIOR37	TO17	–	P71	P57	P71	P57	P71	–	P71	–	–	–	–	–
PIOR36	TO16	–	P32	P65	P32	P65	P32	–	P32	–	–	–	–	–
PIOR35	TO15	–	P70	P56	P70	P56	P70	–	P70	–	–	–	–	–
PIOR34	TO14	–	P31	P64	P31	P64	P31	–	P31	–	–	–	–	–
PIOR33	TO13	PIOR95=0	P10	P55	P10	P55	P10	–	P10	–	P10	–	–	–
		PIOR95=1	P10	P120	P10	P120	P10	P120	P10	P120	P10	P120	–	P120
PIOR32	TO12	–	P11	P46	P11	P46	P11	–	P11	–	P11	–	P11	–
PIOR31	TO11	–	P12	P54	P12	P54	P12	–	P12	–	P12	–	P12	–
PIOR30	TO10	–	P41	P45	P41	P45	P41	–	P41	–	P41	–	–	–

Note Functions TO14 to TO17 are for RL78/F25 products only.

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.

Bit 7 is read-only. The other bits can be read or written to.

Reset signal generation clears this register to 00H.

Figure 4-87. Format of Peripheral I/O Redirection Register 4 (PIOR4) (1/2)

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	PIOR settings	100-pin		80-pin		64-pin		48-pin	
			Setting value		Setting value		Setting value		Setting value	
			0	1	0	1	0	1	0	1
PIOR46	CRXD0	–	P11	P73	P11	P73	P11	P73	P11 Note	P73 Note
	CTXD0	–	P10	P72	P10	P72	P10	P72	P10 Note	P72 Note
PIOR45	LRXD1	PIOR93=0	P11	P107	P11	–	P11	–	P11 Note	–
		PIOR93=1	P11	P125	P11	P125	P11	P125	P11 Note	P125 Note
	LTXD1	PIOR93=0	P10	P106	P10	–	P10	–	P10 Note	–
		PIOR93=1	P10	P120	P10	P120	P10	P120	P10 Note	P120 Note
PIOR44	LRXD0	–	P14	P43	P14	P43	P14	P43	P14	–
	LTXD0	–	P13	P42	P13	P42	P13	P42	P13	–
PIOR43	SI11	–	P70	P152	P70	–	P70	–	P70	–
	SO11	PIOR92=0	P72	P151	P72	–	P72	–	P72	–
		PIOR92=1	P32	P151	P32	–	P32	–	P32	–
	SCK11	–	P71	P153	P71	–	P71	–	P71	–
	SSI11	–	P73	P150	P73	–	P73	–	P73	–
	SCL11	–	P71	–	P71	–	P71	–	P71	–
	SDA11	–	P70	–	P70	–	P70	–	P70	–

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

Figure 4-87. Format of Peripheral I/O Redirection Register 4 (PIOR4) (2/2)

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	PIOR settings	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
PIOR42	SI10/ RXD1	PIOR91=0	P11	P75	P11	P75	P11	P75	P11	–	P11	–
		PIOR91=1	P41	P75	P41	P75	P41	P75	P41	–	P41	–
	SO10/ TXD1	–	P12	P74	P12	P74	P12	P74	P12	–	P12	–
	SCK10	PIOR91=0	P10	P76	P10	P76	P10	P76	P10	–	P10	–
		PIOR91=1	P120	P76	P120	P76	P120	P76	P120	–	P120	–
	SSI10	–	P54	P77	P54	P77	–	P77	–	–	–	–
	SCL10	–	P14	–	P14	–	P14	–	P14	–	P14	–
	SDA10	–	P13	–	P13	–	P13	–	P13	–	P13	–
PIOR41	SI01	–	P13	P53	P13	P53	P13	P53	P13	–	P13	–
	SO01	–	P120	P51	P120	P51	P120	P51	P120	–	P120	–
	SCK01	–	P14	P52	P14	P52	P14	P52	P14	–	P14	–
	SSI01	–	P125	P50	P125	P50	P125	P50	P125	–	P125	–
	SCL01	–	P14	–	P14	–	P14	–	P14	–	P14	–
	SDA01	–	P13	–	P13	–	P13	–	P13	–	P13	–
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/ SCK00	–	P17	P60	P17	P60	P17	P60	P17	P60	P17	P60
	SSI00	–	P30	P63	P30	P63	P30	P63	P30	P63	P30	P63

Note RL78/F25 products only.

Caution The 24-pin product does not have a PIOR4 register.

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 4-88. 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		24-pin	
		Setting value		Setting value		Setting value		Setting value		Setting value		Setting value	
		0	1	0	1	0	1	0	1	0	1	0	1
PIOR53	INTP3	P17	P50	P17	P50	P17	P50	P17	–	P17	–	P17	–
PIOR52	INTP2	P30	P31	P30	P31	P30	P31	P30	P31	P30	–	P30	–
PIOR50	KR7	P77	–	P77	–	P77	P96	–	P92	–	–	–	–
	KR6	P76	–	P76	–	P76	P95	–	P91	–	–	–	–
	KR5	P75	–	P75	–	P75	P94	–	P90	–	P85	–	–
	KR4	P74	–	P74	–	P74	P93	–	P87	–	P84	–	–
	KR3	P73	–	P73	–	P73	P92	P73	P86	–	P83	–	–
	KR2	P72	–	P72	–	P72	P91	P72	P85	–	P82	–	P82
	KR1	P71	–	P71	–	P71	P90	P71	P84	–	P81	–	P81
	KR0	P70	–	P70	–	P70	P87	P70	P83	–	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 4-89. 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- and 24-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 RDe I/O functions; that is, it specifies which I/O port is assigned to each I/O pin of timer RDe0.

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 4-90. 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		24-pin	
		Setting value		Setting value		Setting value		Setting value		Setting value		Setting value	
		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
PIOR71	TRDIOB0	P125	P11	P125	P11	P125	P11	P125	P11	P125	P11	–	P11
PIOR70	TRDIOA0/ TRDCLK0	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 4-91. 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
PIOR80	RTC1HZ	P15	P03

Caution The 80-, 64-, 48-, 32-, and 24-pin products do not have the PIOR8 register.

4.3.18 Peripheral I/O Redirection Register 9 (PIOR9)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.

PIOR9 enables or disables redirection of the timer array unit functions and the serial communication functions; that is, it specifies which I/O port is assigned to the timer array unit output pin, serial data I/O pin of LIN, and serial data I/O pin of the serial array unit, clock I/O pin.

This register can be set by an 8-bit memory manipulation instruction.

Bits 7 and 6 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 4-92. Format of Peripheral I/O Redirection Register 9 (PIOR9)

Address: F001FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIOR9	0	0	PIOR95	PIOR94	PIOR93	PIOR92	PIOR91	PIOR90

Bit	Function	PIOR settings	100-pin		80-pin		64-pin		48-pin		32-pin		24-pin	
			Setting value		Setting value		Setting value		Setting value		Setting value		Setting value	
			0	1	0	1	0	1	0	1	0	1	0	1
PIOR95	TO13	PIOR33=0	P10	P10	P10	P10	P10	P10	P10	P10	P10	P10	–	–
		PIOR33=1	P55	P120	P55	P120	–	P120	–	P120	–	P120	–	P120
PIOR94 Note	LRXD2	–	P154	P71	–	P71	–	P71	–	P71	–	–	–	–
	LTXD2	–	P155	P70	–	P70	–	P70	–	P70	–	–	–	–
PIOR93 Note	LRXD1	PIOR45=0	P11	P11	P11	P11	P11	P11	P11	P11	–	–	–	–
		PIOR45=1	P107	P125	–	P125	–	P125	–	P125	–	–	–	–
	LTXD1	PIOR45=0	P10	P10	P10	P10	P10	P10	P10	P10	–	–	–	–
		PIOR45=1	P106	P120	–	P120	–	P120	–	P120	–	–	–	–
PIOR92	SO11	PIOR43=0	P72	P32	P72	P32	P72	P32	P72	P32	Use prohibited		Use prohibited	
		PIOR43=1	P151	P151	–	–	–	–	–	–				
PIOR91	SI10/RXD1	PIOR42=0	P11	P41	P11	P41	P11	P41	P11	P41	P11	P41	P11	–
		PIOR42=1	P75	P75	P75	P75	P75	P75	–	–	–	–	–	–
	SCK10	PIOR42=0	P10	P120	P10	P120	P10	P120	P10	P120	P10	P120	–	–
		PIOR42=1	P76	P76	P76	P76	P76	P76	–	–	–	–	–	–
PIOR90	TO01	PIOR11=0	P30	P30	P30	P30	P30	P30	P30	P30	P30	P30	P30	P30
		PIOR11=1	P126	P60	P126	P60	–	P60	–	P60	–	P60	–	P60
	TO02	PIOR12=0	P16	P16	P16	P16	P16	P16	P16	P16	P16	P16	P16	P16
		PIOR12=1	P67	P61	P67	P61	–	P61	–	P61	–	P61	–	P61
	TO03	PIOR13=0	P125	P125	P125	P125	P125	P125	P125	P125	P125	P125	–	–
		PIOR13=1	P127	P62	–	P62	–	P62	–	P62	–	P62	–	P62
	TO07	PIOR17=0	P120	P120	P120	P120	P120	P120	P120	P120	P120	P120	P120	P120
		PIOR17=1	P44	P63	P44	P63	–	P63	–	P63	–	P63	–	P63

Note RL78/F25 products only.

Caution Set the bit the use of which is prohibited to 0.

4.3.19 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 4-93. 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 32-pin and 24-pin products do not have the P140/PCLBUZ0 pin.

PSR14	Control target output port: P14/TO06/TRDIOC0/SCK01/SCL01
0	Normal slew rate
1	Special slew rate (slower than normal slew rate)

PSR120	Control target output port: P120/TO07/(TO13)/TRDIOD0/SO01/(SCK10)/(LTXD1)
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/TO11/(TRDIOD0)/SO10/TXD1/SNZOUT3
0	Normal slew rate
1	Special slew rate (slower than normal slew rate)

PSR10	Control target output port: P10/TO13/TRJO0/SCK10/SCL10/LTXD1/CTXD0
0	Normal slew rate
1	Special slew rate (slower than normal slew rate)

Caution 24-pin products do not have the P10/TO13/TRJO0/SCK10/SCL10 pin.

4.3.20 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 4-94. 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". 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". 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. 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. 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". 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". 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. 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. When PIOR60 = 1: SNOOZE status output to P57 is enabled (only in 100-pin or 80-pin products).

Caution The 24-pin product does not have a SNZOUT1 pin.

4.3.21 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 4-95. 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". 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". 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. 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. 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". 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". 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. 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. When PIOR62 = 1: SNOOZE status output to P65 is enabled (only in 100-pin or 80-pin products).

Caution The 24-pin product does not have a SNZOUT2 pin.

4.3.22 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 4-96. 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". 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". 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. 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. 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". 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". 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. 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. When PIOR64 = 1: SNOOZE status output to P157 is enabled (only in 100-pin products).

Caution The 32-pin and 24-pin products do not have the PSNZCNT2 register.

4.3.23 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 4-97. 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". 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". 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. 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. 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". 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". 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. 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. When PIOR66 = 1: SNOOZE status output to P155 is enabled (only in 100-pin products).

Caution The 32-pin and 24-pin products do not have the PSNZCNT3 register.

4.3.24 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 4-98. 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 Pm.

(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)

Ports 1, 6, 7, and 12 are possible to connect an external device operating on a different potential (3 V) when operating at V_{DD} = 4.0 V to 5.5 V by switching I/O buffers with the port input mode register (PIMm) and port output mode register (POMm).

When receiving input from an external device with a different potential (3 V), set the port input mode registers 1 and 7 (PIM1, PIM7) on a bit-by-bit basis to enable normal input(CMOS)/TTL input buffer switching.

When outputting data to an external device with a different potential (3 V), set the port output mode registers 1, 6, 7, and 12 (POM1, POM6, POM7, and POM12) on a bit-by-bit basis to enable switching the output buffer to the N-ch open-drain, it is possible to cope with different potentials.

(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.

(3) Setting procedure when using I/O pins of IICA0 functions

<1> Externally pull up the pin to be used (on-chip pull-up resistor cannot be used).

In case of IICA0: P63/SDAA0, P62/SCLA0

<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 0.
<4> Set the corresponding bit of the POM6 register to 1 to set the N-ch open-drain output (EV_{DD} tolerance) mode.
<5> Set the corresponding bit of the PIM6 register to 1 to switch to the TTL input buffer.
<6> Set the corresponding bit of the PM6 register 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 IICA0 function.

(4) Setting procedure when using I/O pins of UART function of LIN/UART module (RLIN3)

(a) Use as 3 V input port

<1> Externally pull up the pin to be used (on-chip pull-up resistor cannot be used).

In case of RLIN3 channel.0: P14/LRXD0
In case of RLIN3 channel.1: P11/LRXD1, P125/(LRXD1)
In case of RLIN3 channel.2: P71/(LRXD2)

<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the corresponding bit of the PIMm 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 RLIN3 channel.0: P13/LTXD0
In case of RLIN3 channel.1: P10/LTXD1, P120/(LTXD1)
In case of RLIN3 channel.2: P70/(LTXD2)

<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 POMm registers to 1 to set the N-ch open-drain output (EV_{DD} tolerance) mode.
<5> Set the output mode by manipulating the PMm 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 UART mode of the LIN/UART module (RLIN3).

4.5 Setting 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		PIORx	POMm	PMCm	PMm	Pm	PIMm	TSPMCm	PITHLm
	Function Name	I/O								
P00	INTP9	Input	–	–	–	1	×	–	–	–
	(TI05)	Input	PIOR05=1	–	–	1	×	–	–	–
	(TO05)	Output	PIOR15=1	–	–	0	0	–	–	–
P01	(TI04)	Input	PIOR04=1	–	–	1	×	–	–	–
	(TO04)	Output	PIOR14=1	–	–	0	0	–	–	–
P02	(TI06)	Input	PIOR06=1	–	–	1	×	–	–	–
	(TO06)	Output	PIOR16=1	–	–	0	0	–	–	–
P03	(RTC1HZ)	Output	PIOR80=1	–	–	0	0	–	–	–
P10	TI13	Input	PIOR23=0	×	–	1	×	0/1	0	0/1
	TO13	Output	PIOR33=0	0	–	0	0	×	0	×
	TRJO0	Output	–	0	–	0	0	×	0	×
	SCK10	Input	PIOR42=0, PIOR91=0	×	–	1	×	0/1	0	0/1
		Output	PIOR42=0, PIOR91=0	0/1	–	0	1	×	0	×
	SCL10	Output	PIOR42=0	0/1	–	0	1	×	0	×
	LTXD1	Output	PIOR45=0	0/1	–	0	1	×	0	×
	CTXD0	Output	PIOR46=0	0	–	0	1	×	0	×
	TS26	I/O	–	×	–	1	×	×	1	×
P11	TI12	Input	PIOR22=0	×	–	1	×	0/1	0	0/1
	SI10	Input	PIOR42=0, PIOR91=0	×	–	1	×	0/1	0	0/1
	TO12	Output	PIOR32=0	0	–	0	0	×	0	×
	SDA10	I/O	PIOR42=0	1	–	0	1	0/1	0	0/1
	RXD1	Input	PIOR42=0, PIOR91=0	×	–	1	×	0/1	0	0/1
	LRXD1	Input	PIOR45=0	×	–	1	×	0/1	0	0/1
	CRXD0	Input	PIOR46=0	×	–	1	×	0	0	0
	(TRDIOB0)	Input	PIOR71=1	×	–	1	×	0/1	0	0/1
		Output	PIOR71=1	0	–	0	0	×	0	×
	TS25	I/O	–	×	–	1	×	×	1	×

(**Remarks** are listed on the next page of **Table 4-29 (8/8)**.)

Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (2/8)

Pin Name	Alternate Function		PIORx	POMm	PMCm	PMm	Pm	PIMm	TSPMCm	PITHLm
	Function Name	I/O								
P12	TI11	Input	PIOR21=0	×	–	1	×	–	0	–
	INTP5	Input	–	×	–	1	×	–	0	–
	TO11	Output	PIOR31=0	0	–	0	0	–	0	–
	SO10	Output	PIOR42=0	0/1	–	0	1	–	0	–
	TXD1	Output	PIOR42=0	0/1	–	0	1	–	0	–
	SNZOUT3	Output	PIOR63=0	0	–	0	0	–	0	–
	(TRDIOD0)	Input	PIOR73=1	×	–	1	×	–	0	–
		Output	PIOR73=1	0	–	0	0	–	0	–
	TSCAP	–	–	×	–	1	×	×	1	×
P13	TI04	Input	PIOR04=0	×	–	1	×	0/1	0	0/1
	SI01	Input	PIOR41=0	×	–	1	×	0/1	0	0/1
	TRDIOA0	Input	PIOR70=0	×	–	1	×	0/1	0	0/1
		Output	PIOR70=0	0	–	0	0	×	0	×
	TRDCLK0	Input	PIOR70=0	×	–	1	×	0/1	0	0/1
	TO04	Output	PIOR14=0	0	–	0	0	×	0	×
	SDA01	I/O	PIOR41=0	1	–	0	1	0/1	0	0/1
	LTXD0	Output	PIOR44=0	0/1	–	0	1	×	0	×
	TS24	I/O	–	×	–	1	×	×	1	×
P14	TI06	Input	PIOR06=0	×	–	1	×	0/1	0	0/1
	TO06	Output	PIOR16=0	0	–	0	0	×	0	×
	TRDIOC0	Input	–	×	–	1	×	0/1	0	0/1
		Output	–	0	–	0	0	×	0	×
	SCK01	Input	PIOR41=0	×	–	1	×	0/1	0	0/1
		Output	PIOR41=0	0/1	–	0	1	×	0	×
	SCL01	Output	PIOR41=0	0/1	–	0	1	×	0	×
	LRXD0	Input	PIOR44=0	×	–	1	×	0/1	0	0/1
	TS23	I/O	–	×	–	1	×	×	1	×
P15	TI05	Input	PIOR05=0	×	–	1	×	–	0	–
	TO05	Output	PIOR15=0	0	–	0	0	–	0	–
	TRDIOA1	Input	–	×	–	1	×	–	0	–
		Output	–	0	–	0	0	–	0	–
	SO00	Output	PIOR40=0	0/1	–	0	1	–	0	–
	TXD0	Output	PIOR40=0	0/1	–	0	1	–	0	–
	RTC1HZ	Output	PIOR80=0	0	–	0	0	–	0	–
	(TRDIOA0)	Input	PIOR70=1	×	–	1	×	–	0	–
		Output	PIOR70=1	0	–	0	0	–	0	–
	(TRDCLK0)	Input	PIOR70=1	×	–	1	×	–	0	–
	TS18	I/O	–	×	–	1	×	×	1	×

(**Remarks** are listed on the next page of **Table 4-29 (8/8)**.)

Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (3/8)

Pin Name	Alternate Function		PIORx	POMm	PMCm	PMm	Pm	PIMm	TSPMCm	PITHLm
	Function Name	I/O								
P16	TI02	Input	PIOR02=0	×	–	1	×	0/1	0	0/1
	SI00	Input	PIOR40=0	×	–	1	×	0/1	0	0/1
	TRDIOC1	Input	–	×	–	1	×	0/1	0	0/1
		Output	–	0	–	0	0	×	0	×
	TO02	Output	PIOR12=0	0	–	0	0	×	0	×
	SDA00	I/O	PIOR40=0	1	–	0	1	0/1	0	0/1
	RXD0	Input	PIOR40=0	×	–	1	×	0/1	0	0/1
	TS17	I/O	–	×	–	1	×	×	1	×
P17	TI00	Input	PIOR00=0	×	–	1	×	0/1	0	0/1
	INTP3	Input	PIOR53=0	×	–	1	×	0/1	0	0/1
	TRDIOB1	Input	–	×	–	1	×	0/1	0	0/1
		Output	–	0	–	0	0	×	0	×
	SCK00	Input	PIOR40=0	×	–	1	×	0/1	0	0/1
		Output	PIOR40=0	0/1	–	0	1	×	0	×
	SCL00	Output	PIOR40=0	0/1	–	0	1	×	0	×
	TO00	Output	PIOR10=0	0	–	0	0	×	0	×
	TS16	I/O	–	×	–	1	×	×	1	×
P30	TI01	Input	PIOR01=0	–	–	1	×	0/1	–	0/1
	INTP2	Input	PIOR52=0	–	–	1	×	0/1	–	0/1
	TRDIOD1	Input	–	–	–	1	×	0/1	–	0/1
		Output	–	–	–	0	0	×	–	×
	SSI00	Input	PIOR40=0	–	–	1	×	0/1	–	0/1
	TO01	Output	PIOR11=0	–	–	0	0	×	–	×
	SNZOUT0	Output	PIOR60=0	–	–	0	0	×	–	×
P31	TI14	Input	PIOR24=0	–	–	1	×	–	–	–
	TO14	Output	PIOR34=0	–	–	0	0	–	–	–
	STOPST Note	Output	–	–	–	0	0	–	–	–
	(INTP2)	Input	PIOR52=1	–	–	1	×	–	–	–
P32	TI16	Input	PIOR26=0	×	–	1	×	–	–	–
	INTP7	Input	–	×	–	1	×	–	–	–
	TO16	Output	PIOR36=0	0	–	0	0	–	–	–
	(SO11)	Output	PIOR43=0, PIOR92=1	0/1	–	0	1	–	–	–

Note The STOPST function can be assigned via settings in the STOP status output control register (STPSTC).

(**Remarks** are listed on the next page of **Table 4-29 (8/8)**.)

Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (4/8)

Pin Name	Alternate Function		PIORx	POMm	PMCm	PMm	Pm	PIMm	TSPMCm	PITHLm
	Function Name	I/O								
P40	TOOL0	I/O	–	–	–	×	×	–	–	–
P41	TI10	Input	PIOR20=0	–	–	1	×	–	–	0/1
	TO10	Output	PIOR30=0	–	–	0	0	–	–	×
	TRJIO0	Input	–	–	–	1	×	–	–	0/1
		Output	–	–	–	0	0	–	–	×
	TRD0RES	Input	–	–	–	1	×	–	–	0/1
	VCOUT0	Output	–	–	–	0	0	–	–	×
	SNZOUT2	Output	PIOR62=0	–	–	0	0	–	–	×
	(SI10)	Input	PIOR42=0, PIOR91=1	–	–	1	×	–	–	0/1
	(RXD1)	Input	PIOR42=0, PIOR91=1	–	–	1	×	–	–	0/1
P42	(LTXD0)	Output	PIOR44=1	–	–	0	1	–	–	–
P43	(LRXD0)	Input	PIOR44=1	–	–	1	×	–	–	0/1
P44	(TI07)	Input	PIOR07=1	–	–	1	×	–	–	–
	(TO07)	Output	PIOR17=1, PIOR90=0	–	–	0	0	–	–	–
P45	(TI10)	Input	PIOR20=1	–	–	1	×	–	–	–
	(TO10)	Output	PIOR30=1	–	–	0	0	–	–	–
P46	(TI12)	Input	PIOR22=1	–	–	1	×	–	–	–
	(TO12)	Output	PIOR32=1	–	–	0	0	–	–	–
P47	INTP13	Input	–	–	–	1	×	–	–	–
P50	(SSI01)	Input	PIOR41=1	–	–	1	×	–	0	0/1
	(INTP3)	Input	PIOR53=1	–	–	1	×	–	0	0/1
	TS19	I/O	–	×	–	1	×	×	1	×
P51	INTP11	Input	–	–	–	1	×	–	0	–
	(SO01)	Output	PIOR41=1	–	–	0	1	–	0	–
	TS20	I/O	–	×	–	1	×	×	1	×
P52	(SCK01)	Input	PIOR41=1	–	–	1	×	–	0	0/1
		Output	PIOR41=1	–	–	0	1	–	0	×
	(STOPST) Note	Output	–	–	–	0	0	–	0	0/1
	TS21	I/O	–	×	–	1	×	×	1	×
P53	INTP10	Input	–	–	–	1	×	–	0	0/1
	(SI01)	Input	PIOR41=1	–	–	1	×	–	0	0/1
	TS22	I/O	–	×	–	1	×	×	1	×
P54	SSI10	Input	PIOR42=0	–	–	1	×	0/1	0	0/1
	(TI11)	Input	PIOR21=1	–	–	1	×	0/1	0	0/1
	(TO11)	Output	PIOR31=1	–	–	0	0	×	0	×
	TS27	I/O	–	×	–	1	×	×	1	×
P55	(TI13)	Input	PIOR23=1	–	–	1	×	–	0	–
	(TO13)	Output	PIOR33=1	–	–	0	0	–	0	–
	TS28	I/O	–	×	–	1	×	×	1	×

Note The STOPST function can be assigned via settings in the STOP status output control register (STPSTC).

(**Remarks** are on the next page of **Table 4-29 (8/8)**.)

Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (5/8)

Pin Name	Alternate Function		PIORx	POMm	PMCm	PMm	Pm	PIMm	TSPMCm	PITHLm
	Function Name	I/O								
P56	(TI15)	Input	PIOR25=1	–	–	1	×	–	0	–
	(TO15)	Output	PIOR35=1	–	–	0	0	–	0	–
	(SNZOUT1)	Output	PIOR61=1	–	–	0	0	–	0	–
	TS29	I/O	–	×	–	1	×	×	1	×
P57	(TI17)	Input	PIOR27=1	–	–	1	×	–	0	–
	(TO17)	Output	PIOR37=1	–	–	0	0	–	0	–
	(SNZOUT0)	Output	PIOR60=1	–	–	0	0	–	0	–
	TS30	I/O	–	×	–	1	×	×	1	×
P60	(SCK00)	Input	PIOR40=1	×	–	1	×	–	0	0/1
		Output	PIOR40=1	0	–	0	1	–	0	×
	(SCL00)	Output	PIOR40=1	0/1	–	0	1	–	0	×
	(TO01)	Output	PIOR11=1, PIOR90=1	0	–	0	0	–	0	×
	CRXD1	Input	–	×	–	1	×	0	0	0
	TS0	I/O	–	×	–	1	×	×	1	×
P61	(SI00)	Input	PIOR40=1	×	–	1	×	–	0	0/1
	(SDA00)	I/O	PIOR40=1	1	–	0	1	–	0	0/1
	(RXD0)	Input	PIOR40=1	×	–	1	×	–	0	0/1
	(TO02)	Output	PIOR12=1, PIOR90=1	0	–	0	0	–	0	×
	CTXD1	Output	–	0	–	0	1	×	0	×
	TS1	I/O	–	×	–	1	×	×	1	×
P62	SCLA0	I/O	–	1	–	0	0	0/1	0	0/1
	(SO00)	Output	PIOR40=1	0	–	0	1	×	0	×
	(TXD0)	Output	PIOR40=1	0	–	0	1	×	0	×
	(TO03)	Output	PIOR13=1, PIOR90=1	0	–	0	0	×	0	×
	TS2	I/O	–	×	–	1	×	×	1	×
P63	SDAA0	I/O	–	1	–	0	0	0/1	0	0/1
	(SSI00)	Input	PIOR40=1	×	–	1	×	0	0	0/1
	(TO07)	Output	PIOR17=1, PIOR90=1	0	–	0	0	×	0	×
	TS3	I/O	–	×	–	1	×	×	1	×
P64	(TI14)	Input	PIOR24=1	–	–	1	×	–	0	–
	(TO14)	Output	PIOR34=1	–	–	0	0	–	0	–
	(SNZOUT3)	Output	PIOR63=1	–	–	0	0	–	0	–
	TS4	I/O	–	×	–	1	×	×	1	×
P65	(TI16)	Input	PIOR26=1	–	–	1	×	–	0	–
	(TO16)	Output	PIOR36=1	–	–	0	0	–	0	–
	(SNZOUT2)	Output	PIOR62=1	–	–	0	0	–	0	–
	TS5	I/O	–	×	–	1	×	×	1	×

(**Remarks** are listed on the next page of **Table 4-29 (8/8)**.)

Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (6/8)

Pin Name	Alternate Function		PIORx	POMm	PMCm	PMm	Pm	PIMm	TSPMCm	PITHLm
	Function Name	I/O								
P66	(TI00)	Input	PIOR00=1	–	–	1	×	–	0	–
	(TO00)	Output	PIOR10=1	–	–	0	0	–	0	–
	TS6	I/O	–	×	–	1	×	×	1	×
P67	(TI02)	Input	PIOR02=1	–	–	1	×	–	0	–
	(TO02)	Output	PIOR12=1, PIOR90=0	–	–	0	0	–	0	–
	TS7	I/O	–	×	–	1	×	×	1	×
P70	ANI24	Input	–	×	1	1	×	×	0	×
	TI15	Input	PIOR25=0	×	0	1	×	0/1	0	0/1
	SI11	Input	PIOR43=0	×	0	1	×	0/1	0	0/1
	INTP8	Input	–	×	0	1	×	0/1	0	0/1
	TO15	Output	PIOR35=0	0	0	0	0	×	0	×
	KR0	Input	PIOR50=0	×	0	1	×	0/1	0	0/1
	SDA11	I/O	PIOR43=0	1	0	0	1	0/1	0	0/1
	SNZOUT4	Output	PIOR64=0	0	0	0	0	×	0	×
	TS15	I/O	–	×	0	1	×	×	1	×
	(LTXD2)	Output	PIOR94=1	–	0	0	1	–	0	–
P71	ANI25	Input	–	×	1	1	×	×	0	×
	TI17	Input	PIOR27=0	×	0	1	×	0/1	0	0/1
	INTP6	Input	–	×	0	1	×	0/1	0	0/1
	TO17	Output	PIOR37=0	0	0	0	0	×	0	×
	KR1	Input	PIOR50=0	×	0	1	×	0/1	0	0/1
	SCK11	Input	PIOR43=0	×	0	1	×	0/1	0	0/1
		Output	PIOR43=0	0/1	0	0	1	×	0	×
	SCL11	Output	PIOR43=0	0/1	0	0	1	×	0	×
	SNZOUT5	Output	PIOR65=0	0	0	0	0	×	0	×
	TS14	I/O	–	×	0	1	×	×	1	×
	(LRXD2)	Input	PIOR94=1	–	0	1	×	–	0	0/1
P72	ANI26	Input	–	×	1	1	×	–	0	–
	KR2	Input	PIOR50=0	×	0	1	×	–	0	–
	SO11	Output	PIOR43=0, PIOR92=0	0/1	0	0	1	–	0	–
	SNZOUT6	Output	PIOR66=0	0	0	0	0	–	0	–
	(CTXD0)	Output	PIOR46=1	0	0	0	1	–	0	–
	TS13	I/O	–	×	0	1	×	×	1	×
P73	ANI27	Input	–	–	1	1	×	×	0	×
	KR3	Input	PIOR50=0	–	0	1	×	0/1	0	0/1
	SSI11	Input	PIOR43=0	–	0	1	×	0/1	0	0/1
	SNZOUT7	Output	PIOR67=0	–	0	0	0	×	0	×
	(CRXD0)	Input	PIOR46=1	–	0	1	×	0	0	0
	TS12	I/O	–	×	0	1	×	×	1	×

(**Remarks** are listed on the next page of **Table 4-29 (8/8)**.)

Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (7/8)

Pin Name	Alternate Function		PIORx	POMm	PMCm	PMm	Pm	PIMm	TSPMCm	PITHLm
	Function Name	I/O								
P74	ANI28	Input	–	–	1	1	×	–	0	–
	KR4	Input	PIOR50=0	–	0	1	×	–	0	–
	(SO10)	Output	PIOR42=1	–	0	0	1	–	0	–
	(TXD1)	Output	PIOR42=1	–	0	0	1	–	0	–
	TS11	I/O	–	×	0	1	×	×	1	×
P75	KR5	Input	PIOR50=0	–	–	1	×	–	0	0/1
	(SI10)	Input	PIOR42=1	–	–	1	×	–	0	0/1
	(RXD1)	Input	PIOR42=1	–	–	1	×	–	0	0/1
	TS10	I/O	–	×	–	1	×	×	1	×
P76	KR6	Input	PIOR50=0	–	–	1	×	–	0	0/1
	(SCK10)	Input	PIOR42=1	–	–	1	×	–	0	0/1
		Output	PIOR42=1	–	–	0	1	–	0	×
	TS9	I/O	–	×	–	1	×	×	1	×
P77	INTP12	Input	–	–	–	1	×	–	0	0/1
	KR7	Input	PIOR50=0	–	–	1	×	–	0	0/1
	(SSI10)	Input	PIOR42=1	–	–	1	×	–	0	0/1
	TS8	I/O	–	×	–	1	×	×	1	×
P80	ANI0	Input	–	–	1	1	×	–	–	–
	ANO0	Output	–	–	1	1	×	–	–	–
P81	ANI1	Input	–	–	1	1	×	–	–	–
P82	ANI2	Input	–	–	1	1	×	–	–	–
	IVCMP00	Input	–	–	1	1	×	–	–	–
P83	ANI3	Input	–	–	1	1	×	–	–	–
	IVCMP01	Input	–	–	1	1	×	–	–	–
P84	ANI4	Input	–	–	1	1	×	–	–	–
	IVCMP02	Input	–	–	1	1	×	–	–	–
P85	ANI5	Input	–	–	1	1	×	–	–	–
	IVCMP03	Input	–	–	1	1	×	–	–	–
	IVREF0	Input	–	–	1	1	×	–	–	–
P86-P87	ANI6 to ANI7	Input	–	–	1	1	×	–	–	–
P90-P97	ANI8 to ANI15	Input	–	–	1	1	×	–	–	–
P100-P105	ANI16 to ANI21	Input	–	–	1	1	×	–	–	–
P106	(LTXD1)	Output	PIOR45=1, PIOR93=0	–	–	0	1	–	–	–
P107	(LRXD1)	Input	PIOR45=1, PIOR93=0	–	–	1	×	–	–	0/1

(**Remarks** are listed on the next page of **Table 4-29 (8/8)**.)

Table 4-29. Settings of Port Mode Register and Output Latch When Using Alternate Function (8/8)

Pin Name	Alternate Function		PIORx	POMm	PMCm	PMm	Pm	PIMm	TSPMCm	PITHLm
	Function Name	I/O								
P120	ANI23	Input	–	×	1	1	×	–	–	×
	TI07	Input	PIOR07=0	×	0	1	×	–	–	0/1
	INTP4	Input	–	×	0	1	×	–	–	0/1
	TRDIOD0	Input	PIOR73=0	×	0	1	×	–	–	0/1
		Output	PIOR73=0	0	0	0	0	–	–	×
	SO01	Output	PIOR41=0	0/1	0	0	1	–	–	×
	TO07	Output	PIOR17=0	0	0	0	0	–	–	×
	(SCK10)	Input	PIOR42=0, PIOR91=1	×	0	1	×	–	–	0/1
		Output	PIOR42=0, PIOR91=1	0/1	0	0	1	–	–	×
	(LTXD1)	Output	PIOR45=1, PIOR93=1	0/1	0	0	1	–	–	×
	(TO13)	Output	PIOR33=1, PIOR95=1	0	0	0	0	–	–	×
P125	ANI22	Input	–	–	1	1	×	×	–	×
	TI03	Input	PIOR03=0	–	0	1	×	0/1	–	0/1
	INTP1	Input	–	–	0	1	×	0/1	–	0/1
	TO03	Output	PIOR13=0	–	0	0	0	×	–	×
	TRDIOB0	Input	PIOR71=0	–	0	1	×	0/1	–	0/1
		Output	PIOR71=0	–	0	0	0	×	–	×
	SSI01	Input	PIOR41=0	–	0	1	×	0/1	–	0/1
	SNZOUT1	Output	PIOR61=0	–	0	0	0	×	–	×
	(LRXD1)	Input	PIOR45=1, PIOR93=1	–	0	1	×	0/1	–	0/1
P126	(TI01)	Input	PIOR01=1	–	–	1	×	–	–	–
	(TO01)	Output	PIOR11=1, PIOR90=0	–	–	0	0	–	–	–
P127	(TI03)	Input	PIOR03=1	–	–	1	×	–	–	–
	(TO03)	Output	PIOR13=1, PIOR90=0	–	–	0	0	–	–	–
P130	RESOUT	Output	–	–	–	–	0	–	–	–
P137	INTP0	Input	–	–	–	–	×	–	–	–
P140	PCLBUZ0	Output	–	–	–	0	0	–	–	–
	TRD1RES	Input	–	–	–	1	×	–	–	–
P150	(SSI11)	Input	PIOR43=1	–	–	1	×	–	–	0/1
P151	(SO11)	Output	PIOR43=1	–	–	0	1	–	–	×
P152	(SI11)	Input	PIOR43=1	–	–	1	×	–	–	0/1
P153	(SCK11)	Input	PIOR43=1	–	–	1	×	–	–	0/1
		Output	PIOR43=1	–	–	0	1	–	–	×
P154	(SNZOUT7)	Output	PIOR67=1	–	–	0	0	–	–	×
	LRXD2	Input	PIOR94=0	–	–	1	×	–	–	0/1
P155	(SNZOUT6)	Output	PIOR66=1	–	–	0	0	–	–	×
	LTXD2	Output	PIOR94=0	–	–	0	1	–	–	–
P156	(SNZOUT5)	Output	PIOR65=1	–	–	0	0	–	–	×
P157	(SNZOUT4)	Output	PIOR64=1	–	–	0	0	–	–	×

(**Remarks** are listed on the next page.)

Remarks **1.** ×: Don't care
PIORx: Peripheral I/O redirection register x
POMm: Port output mode register m
PMCm: Port mode control register m
PMm: Port mode register m
Pm: Port register m
PIMm: Port input mode register m
PITHLm: Port input threshold control register m
TSPMCm: Touch sensor port mode control register

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 PIORx, POMm, PMCm, PMm, Pm, PIMm, TSPMCm, and PITHLm 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 (PIORx).

4. The allocation of CTSU pin functions to ports differs between RL78/F25 and RL78/F22 products (**Table 4-29** shows the case of RL78/F25 100-pin products).

CTSU pin function	Port allocation	RL78/F25				RL78/F22		
		100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
TSCAP	P12	✓	✓	✓	✓	✓	✓	✓
TS0	P60	✓	✓	✓	✓	✓	✓	–
TS1	P61	✓	✓	✓	✓	✓	✓	–
TS2	P62	✓	✓	✓	✓	✓	✓	✓
TS3	P63	✓	✓	✓	✓	✓	✓	✓
TS4	P64	✓	✓	–	–	–	–	–
	P73	–	–	–	–	✓	–	–
TS5	P65	✓	✓	–	–	–	–	–
	P72	–	–	–	–	✓	–	–
TS6	P66	✓	✓	–	–	–	–	–
	P17	–	–	–	–	✓	✓	✓
TS7	P67	✓	✓	–	–	–	–	–
	P16	–	–	–	–	✓	✓	✓
TS8	P77	✓	✓	✓	–	–	–	–
	P15	–	–	–	–	✓	✓	✓
TS9	P76	✓	✓	✓	–	–	–	–
	P14	–	–	–	–	✓	✓	✓
TS10	P75	✓	✓	✓	–	–	–	–
	P13	–	–	–	–	✓	✓	✓
TS11	P74	✓	✓	✓	–	–	–	–
	P10	–	–	–	–	✓	✓	–
TS12	P73	✓	✓	✓	✓	–		
TS13	P72	✓	✓	✓	✓			
TS14	P71	✓	✓	✓	✓			
TS15	P70	✓	✓	✓	✓			
TS16	P17	✓	✓	✓	✓			
TS17	P16	✓	✓	✓	✓			
TS18	P15	✓	✓	✓	✓			
TS19	P50	✓	✓	✓	–			
TS20	P51	✓	✓	✓	–			
TS21	P52	✓	✓	✓	–			
TS22	P53	✓	✓	✓	–			
TS23	P14	✓	✓	✓	✓			
TS24	P13	✓	✓	✓	✓			
TS25	P11	✓	✓	✓	✓			
TS26	P10	✓	✓	✓	✓			
TS27	P54	✓	✓	–	–			
TS28	P55	✓	✓	–	–			
TS29	P56	✓	✓	–	–			
TS30	P57	✓	✓	–	–			

4.6 Cautions When Using Port Function

4.6.1 Cautions on 1-bit Manipulation Instruction for Port Register n (Pm)

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 Pm register of a port whose PMmn 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//F22 and RL78/F25.

<1> The Pm register is read in 8-bit units.

<2> The targeted one bit is manipulated.

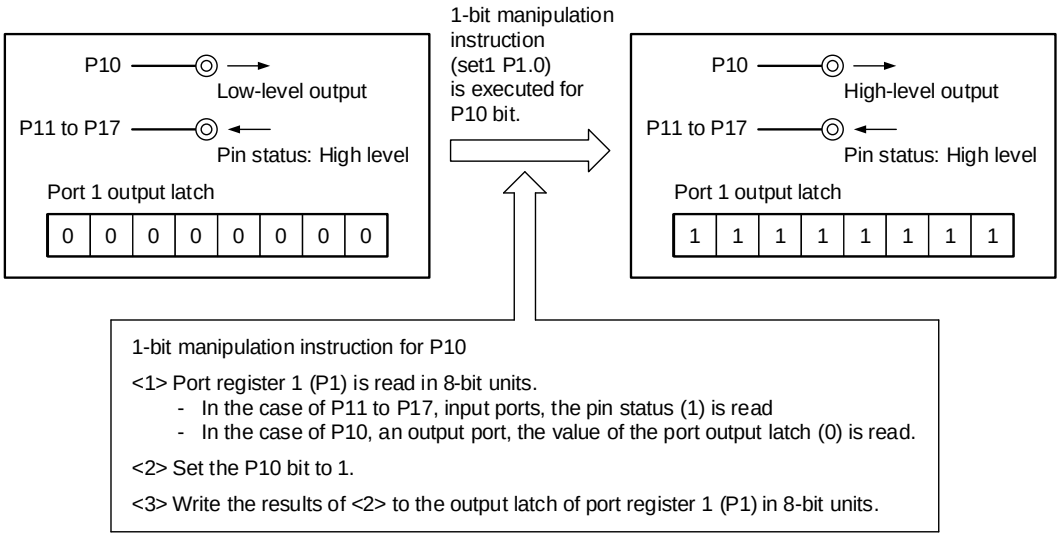
<3> The Pm register is written in 8-bit units.

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 4-99. 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 (PIORx). For details about the alternate function output, see **4.5 Setting 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.

4.6.3 Cautions When Setting Port Related Registers

If any write access to the port register (Pm) is executed by using bit manipulation instructions or operations (AND, OR instructions) in the following cases, the current pin level (1 or 0) will be stored in a port latch for other bits:

- When the pin is set to input (the corresponding bit in PMm register is set to 1). See **4.6.1**.
- The PMS0 bit in the PMS register is set to 1 (reads the output level of the pin). Disable interrupts (DI) before setting the PMS0 bit before reading the port level.

4.6.4 Cautions When Setting Peripheral I/O Redirection Register

Some pin functions may require multiple peripheral I/O redirection register (PIORx) settings when assigned to a port. These functions should be configured with care in combination.

- TO01, TO02, TO03, TO07, TO13 (Timer output pin for timer array unit 0, 1)

Note that PIOR90 bit selects these functions (TO01, TO02, TO03, and TO07) together.

Pin Function	Assigned Port Name (Condition of PIORx Register)		
TO01	P30 (PIOR11 = 0)	P126 (PIOR11 = 1, PIOR90 = 0)	P60 (PIOR11 = 1, PIOR90 = 1)
TO02	P16 (PIOR12 = 0)	P67 (PIOR12 = 1, PIOR90 = 0)	P61 (PIOR12 = 1, PIOR90 = 1)
TO03	P125 (PIOR13 = 0)	P127 (PIOR13 = 1, PIOR90 = 0)	P62 (PIOR13 = 1, PIOR90 = 1)
TO07	P120 (PIOR17 = 0)	P44 (PIOR17 = 1, PIOR90 = 0)	P63 (PIOR17 = 1, PIOR90 = 1)
TO13	P10 (PIOR33 = 0)	P55 (PIOR33 = 1, PIOR95 = 0)	P120 (PIOR33 = 1, PIOR95 = 1)

For the following, also note the pin functions used at the same time.

- SCK10, SI10, SO11 (CSI10, CSI11 input/output pin for serial array unit 1)

Pin Function	Assigned Port Name (Condition of PIORx Register)		
SCK10	P10 (PIOR42 = 0, PIOR91 = 0)	P76 (PIOR42 = 1)	P120 (PIOR42 = 0, PIOR91 = 1)
SI10	P11 (PIOR42 = 0, PIOR91 = 0)	P75 (PIOR42 = 1)	P41 (PIOR42 = 0, PIOR91 = 1)
SO10 Note	P12 (PIOR42 = 0)	P74 (PIOR42 = 1)	
SSI10 Note	P54 (PIOR42 = 0)	P77 (PIOR42 = 1)	
SCK11 Note	P71 (PIOR43 = 0)		P153 (PIOR43 = 1)
SI11 Note	P70 (PIOR43 = 0)		P152 (PIOR43 = 1)
SO11	P72 (PIOR43 = 0, PIOR92 = 0)	P32 (PIOR43 = 0, PIOR92 = 1)	P151 (PIOR43 = 1)
SSI11 Note	P73 (PIOR43 = 0)		P150 (PIOR43 = 1)

- RXD1 (UART1 input pin for serial array unit 1)

Pin Function	Assigned Port Name (Condition of PIORx Register)		
RXD1	P11 (PIOR42 = 0, PIOR91 = 0)	P41 (PIOR42 = 0, PIOR91 = 1)	P75 (PIOR42 = 1)
TXD1 Note	P12 (PIOR42 = 0)		P74 (PIOR42 = 1)

- LRXD1, LTXD1 (LIN/UART input/output pin for RLIN3 channel 1)

Pin Function	Assigned Port Name (Condition of PIORx Register)		
LRXD1	P11 (PIOR45 = 0)	P107 (PIOR45 = 1, PIOR93 = 0)	P125 (PIOR45 = 1, PIOR93 = 1)
LTXD1	P10 (PIOR45 = 0)	P106 (PIOR45 = 1, PIOR93 = 0)	P120 (PIOR45 = 1, PIOR93 = 1)

Note These pin functions are indicated to be used in combination with the corresponding pin functions.

CHAPTER 5 CLOCK GENERATOR

The clock generator should be used within the range satisfying the values specified in **CHAPTER 37** and **CHAPTER 38 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.

	RL78/F22, F25 48, 64, 80, and 100-pin Products	RL78/F22 24-pin and 32-pin Products
X1, X2 pins	√	√
EXCLK pin	√	√
XT1, XT2 pins	√	–
EXCLKS pin	√	–

Caution RL78/F22 24-pin and 32-pin products do not have the subsystem clock.

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 = 2 MHz 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} = 80, 64, 40, 32, 20, 16, 8, 4, or 2 MHz (TYP.) by using the user option byte (000C2H/040C2H). When 80 MHz or 64 MHz is selected as f_{IH}, f_{CLK} is set to 40 MHz or 32 MHz, respectively, after a reset release. The CPU always starts operating with this high-speed on-chip oscillator clock. 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). See **Figure 5-12. Format of High-speed on-chip oscillator frequency select register (HOCODIV)** for more information.

The table below shows the High-speed OCO settable frequencies for RL78/F22 and RL78/F25 products.

Power Supply Voltage	Oscillation Frequency [MHz]								
	2	4	8	16	20	32	40	64	80
2.7 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	√	√
1.8 V ≤ V_{DD} < 2.7 V	√	√	√	√	–	–	–	–	–

An external main system clock (f_{EX} = 2 MHz 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)).

The available frequency of the CPU/peripheral hardware clock (f_{CLK}) depends on the supply voltage V_{DD}.

(2) PLL clock

This clock oscillates the clock whose f_{PLL} is 24 MHz, 32 MHz, 40 MHz, 48 MHz, 64 MHz, or 80 MHz by oscillating the main system clock at 4 MHz, 8 MHz, 16 MHz, or 20 MHz and multiplying by 3, 4, 5, 6, 8, or 10 times. When setting f_{PLL} to 80 MHz, 64 MHz or 48 MHz, the division of f_{CLK} should be set to 40 MHz, 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, 8 MHz, 16 MHz **Note 1**, or 20 MHz **Note 2**. The main system clock (X1 oscillation or external oscillation) can be used as PLL input clock.

2. Set the multiplier of the PLL clock by bits 1 (PLLMUL), 3 (PLLMULA), and 4 (PLLDIV0) in the PLL control register (PLLCTL).

Notes 1. When setting the PLL input clock frequency to 16 MHz, set the FMAINDIV1 and FMAINDIV0 bits in the CKSEL register to 10B (divided by 2).

2. When setting the PLL input clock frequency to 20 MHz, set the FMAINDIV1 and FMAINDIV0 bits in the CKSEL register to 11B (divided by 4).

(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 RDe (f_{SL})
- Clock output/buzzer output control circuit (f_{SL})
- Channel 1 of Timer Array Unit 0 (f_{IL})

This circuit operates when at least bit 4 (WUTMMCK0) in the operation speed mode control register (OSMC) or bit 0 (SELLOSC) in the clock select register (CKSEL) 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 SELLOSC 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 (80 MHz max.) **Notes 1, 3**
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_{PLL}: PLL clock frequency (80 MHz max.) **Notes 2, 3**
f_{MP}: Main/PLL select clock frequency (80 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 80 MHz or 64 MHz after a reset release. When supplying 80 MHz or 64 MHz to timer RDe and RS-CANFD lite, set f_{CLK} to f_{IH}.

2. When setting f_{PLL} to 80 MHz or 64 MHz, the division of f_{MP} should be set within the range of 2 MHz to 40 MHz by the MDIV2 to MDIV0 bits in the f_{MP} clock division register (MDIV). When supplying 80 MHz or 64 MHz to timer RDe and RS-CANFD lite, set f_{CLK} to f_{PLL}.

3. When supplying 80 MHz or 64 MHz to timer RDe and RS-CANFD lite, set the MDIV register to 01H ($f_{MP}/2$ is selected).

4. When 1.8 V ≤ V_{DD} < 2.7 V, be sure to set the f_{CLK} frequency to 16 MHz or less.

Caution f_{IH} cannot be used as a CAN communication clock (f_{CAN}).

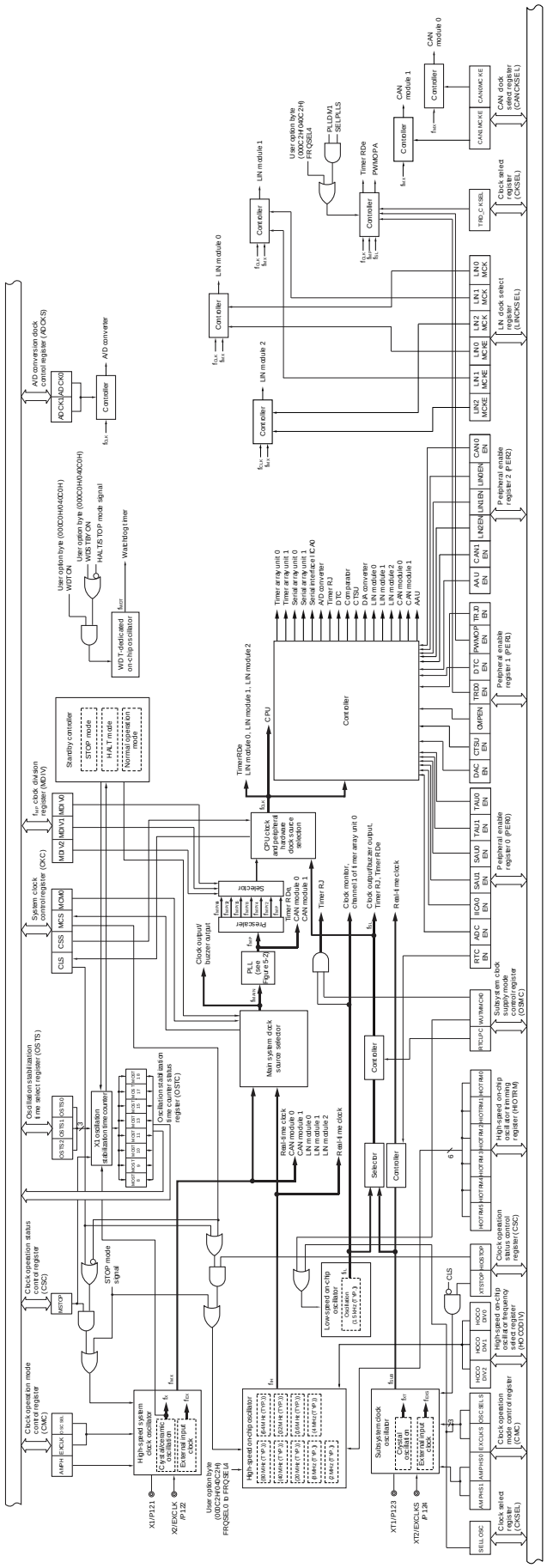
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	• Clock operation mode control register (CMC) • 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) • A/D conversion clock control register (ADCKS)
Oscillators	• X1 oscillator • XT1 oscillator • High-speed on-chip oscillator clock • Low-speed on-chip oscillator clock • PLL circuit • WDT-dedicated low-speed on-chip oscillator

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 (80 MHz max.) **Notes 1, 3**
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_{PLL}: PLL clock frequency (80 MHz max.) **Notes 2, 3**
f_{MP}: Main system/PLL select clock frequency (80 MHz max.)

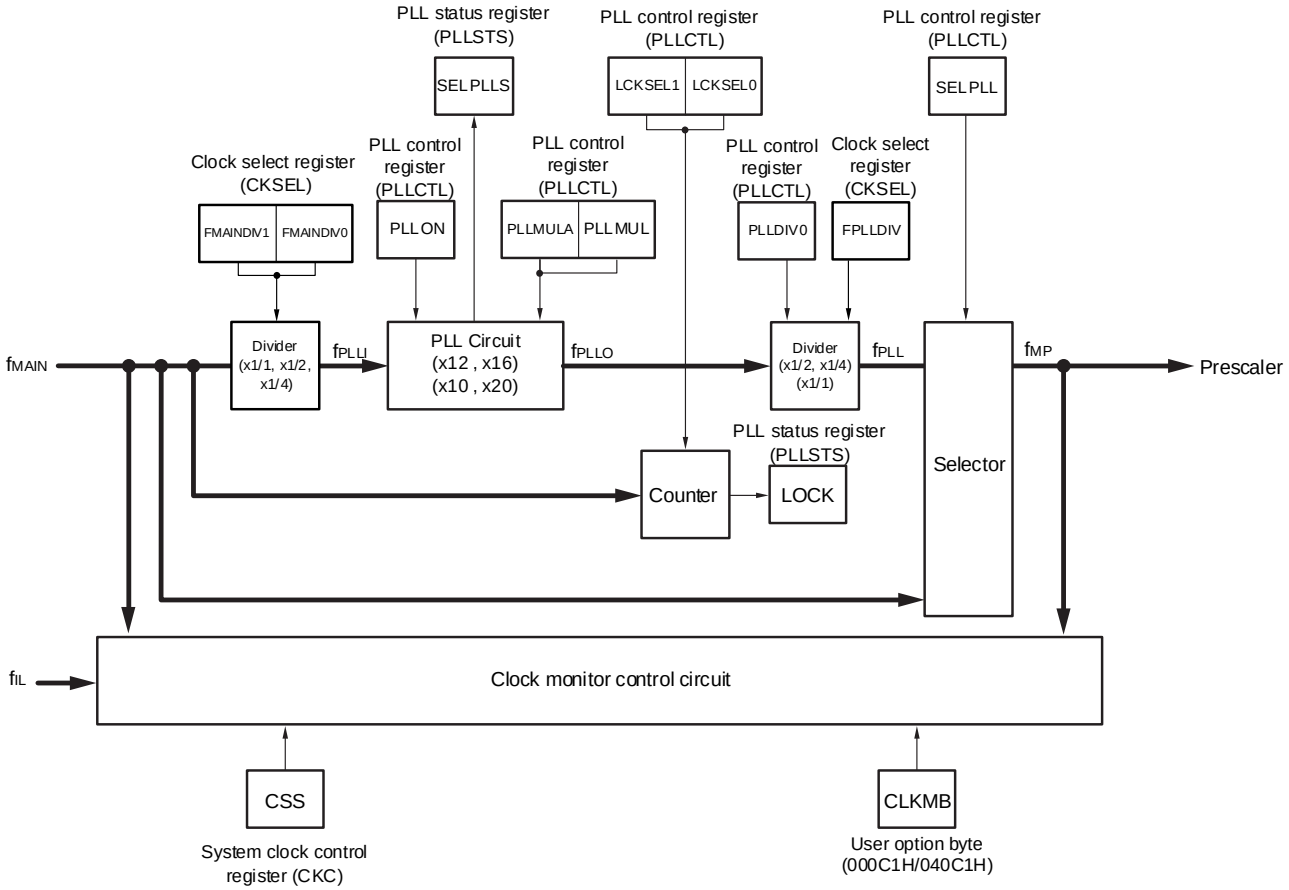
Notes 1. f_{IH} is controlled by hardware so that the MDIV register is set to 01H (f_{MP} = two frequency division) when f_{IH} is set to 80 MHz or 64 MHz after a reset release. When supplying 80 MHz or 64 MHz to timer RDe and RS-CANFD lite, set f_{CLK} to f_{IH}.

2. When setting f_{PLL} to 80 MHz or 64 MHz, the division of f_{MP} should be set within the range of 2 MHz to 40 MHz by the MDIV2 to MDIV0 bits in the f_{MP} clock division register (MDIV). When supplying 80 MHz or 64 MHz to timer RDe and RS-CANFD lite, set f_{CLK} to f_{PLL}.

3. When supplying 80 MHz or 64 MHz to timer RDe and RS-CANFD lite, set the MDIV register to 01H (f_{MP}/2 is selected).

Caution f_{IH} cannot be used as a CAN communication clock (f_{CAN}).

Figure 5-2. Block Diagram of PLL Circuit



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

Table 5-2. Clock Generator Register Configuration

Address	Register Name	Symbol	After Reset	Access Size
FFFA0H	Clock operation mode control register	CMC	00H	8
FFFA1H	Clock operation status control register	CSC	C0H	1, 8
FFFA2H	Oscillation stabilization time counter status register	OSTC	00H	1, 8
FFFA3H	Oscillation stabilization time select register	OSTS	07H	8
FFFA4H	System clock control register	CKC	00H	1, 8
F00A0H	High-speed on-chip oscillator trimming register	HIOTRM	Note 1	8
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV	Note 2	8
F00F3H	Operation speed mode control register	OSMC	00H	8
F02C0H	Peripheral enable register 0	PER0	00H	1, 8
F02C1H	Peripheral enable register 1	PER1	00H	1, 8
F02C2H	Peripheral enable register 2	PER2	00H	1, 8
F02C3H	CAN clock select register	CANCKSEL	00H	1, 8
F02C4H	LIN clock select register	LINCKSEL	00H	1, 8
F02C7H	A/D conversion clock control register	ADCKS	00H	8
F02C8H	PLL control register	PLLCTL	00H	1, 8
F02C9H	PLL status register	PLLSTS	00H	1, 8
F02CAH	Clock select register	CKSEL	00H	1, 8
F02CBH	f_{MP} clock division register	MDIV	00H/01H Note 3	8

Notes 1. The value after reset is the value adjusted at shipment.

2. The value set with the FRQSEL[2:0] bits of the user option byte (000C2H/040C2H).

3. The value of the FRQSEL4 bit in the user option byte (000C2H/040C2H) becomes the initial value of the MDIV0 bit in the MDIV register.

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 Notes 1, 2	EXCLKS	OSCSELS Note 1			
x	0	0	Input port mode	Input port	
0	0	1	XT1 oscillation mode	Crystal 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	2 MHz ≤ fx ≤ 10 MHz
1	2 MHz ≤ fx ≤ 20 MHz

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

Notes 1. 24-pin 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 XT1 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 2 MHz 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 on 24-pin and 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 040C2H), stop counting by the timer RDe (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 37 and CHAPTER 38 ELECTRICAL SPECIFICATIONS.

4. **When selecting f_{IH} as the count source for timer RDe, 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 on 24-pin and 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 CSC 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 5-3.**
   Before stopping the clock oscillation, check the conditions before the clock oscillation is stopped.

Table 5-3. Condition Before Stopping Clock Oscillation and Flag Setting

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock 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). (CLS (bit 7 of the CKC register) = 0 and MCS (bit 5 of the CKC register) = 0, or CLS = 1)	MSTOP = 1
XT1 clock External subsystem clock	CPU and peripheral hardware clocks operate with a clock other than the subsystem clock. (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. (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

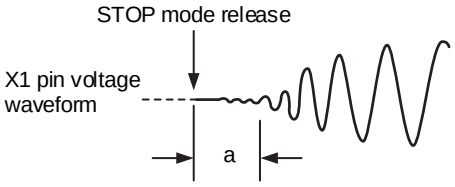
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									f_X = 10 MHz	f_X = 20 MHz
0	0	0	0	0	0	0	0	$2^8/f_X$ max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	$2^8/f_X$ min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	$2^9/f_X$ min.	51.2 μs min.	25.6 μs 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 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

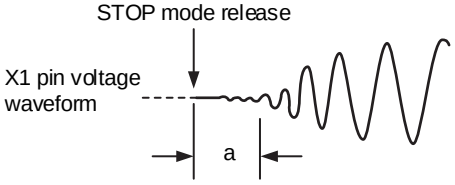
OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	12.8 μs
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_X$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.63 ms
1	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 RDe
- DTC
- PWM option unit
- Timer RJ
- AAU
- LIN0
- LIN1
- LIN2
- RS-CANFD lite 0
- RS-CANFD lite 1
- Capacitive sensing unit

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)

Address: F02C0H 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. • SFR used by the real-time clock (RTC) cannot be written. • The real-time clock (RTC) is in the reset status.
1	Enables input clock supply. • SFR used by the real-time clock (RTC) can be read and written.

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. • SFR used by the A/D converter cannot be written. • The A/D converter is in the reset status.
1	Enables input clock supply. • SFR used by the A/D converter can be read and written.

IICA0EN	Control of serial interface IICA0 input clock supply
0	Stops input clock supply. • SFR used by the serial interface IICA0 cannot be written. • The serial interface IICA0 is in the reset status.
1	Enables input clock supply. • SFR used by the serial interface IICA0 can be read and written.

SAU1EN	Control of serial array unit 1 input clock supply
0	Stops input clock supply. • SFR used by the serial array unit 1 cannot be written. • The serial array unit 1 is in the reset status.
1	Enables input clock supply. • 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. • SFR used by the serial array unit 0 cannot be written. • The serial array unit 0 is in the reset status.
1	Enables input clock supply. • SFR used by the serial array unit 0 can be read and written.

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

TAU1EN	Control of timer array unit 1 input clock supply
0	Stops input clock supply. • SFR used by timer array unit 1 cannot be written. • Timer array unit 1 is in the reset status.
1	Enables input clock supply. • 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. • SFR used by timer array unit 0 cannot be written. • Timer array unit 0 is in the reset status.
1	Enables input clock supply. • SFR used by timer array unit 0 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 Bit 6 must be set to 0.

Figure 5-9. Format of Peripheral Enable Register 1 (PER1)

Address: F02C1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER1	DACEN Note 1	CTSUEN	CMPEN Note 1	TRD0EN Note 2	DTCEN	PWMOPEN	0	TRJ0EN

DACEN Note 1	Control of D/A converter input clock supply
0	Stops input clock supply. • SFR used by the D/A converter cannot be written. • The D/A converter is in the reset status.
1	Enables input clock supply. • SFR used by the D/A converter can be read and written.

CTSUEN	Control of capacitive sensing unit input clock supply
0	Stops input clock supply. • SFR used by the capacitive sensing unit cannot be written. • The capacitive sensing unit is in the reset status.
1	Enables input clock supply. • SFR used by the capacitive sensing unit can be read and written.

CMPEN Note 1	Control of comparator input clock supply
0	Stops input clock supply. • SFR used by comparator cannot be written. • Comparator is in the reset status.
1	Enables input clock supply. • SFR used by comparator can be read and written.

TRD0EN Note 2	Control of timer RDe input clock supply
0	Stops input clock supply. • SFR used by timer RDe cannot be written. • Timer RDe is in the reset status.
1	Enables input clock supply. • SFR used by timer RDe can be read and written.

DTCEN	Control of DTC input clock supply
0	Stops input clock supply. • DTC cannot run.
1	Enables input clock supply. • DTC can run.

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

PWMOPEN	Control of PWMOPA input clock supply
0	Stops input clock supply. • SFR used by PWMOPA cannot be written. • PWMOPA is in the reset status.
1	Enables input clock supply. • SFR used by PWMOPA can be read and written.

TRJ0EN	Control of timer RJ0 input clock supply
0	Stops input clock supply. • SFR used by timer RJ0 cannot be written. • Timer RJ0 is in the reset status.
1	Enables input clock supply. • SFR used by timer RJ0 can be read and written.

Notes 1. Only in the RL78/F25.

2. When FRQSEL4 = 1 in the user option byte (000C2H/040C2H), 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, 5, and 7 in the RL78/F22 products**
- **Bit 1 in the RL78/F25 products**

Figure 5-10. Format of Peripheral Enable Register 2 (PER2)

Address: F02C2H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	1	<0>
PER2	0	AAUEN	CAN1EN Note	LIN2EN Note	LIN1EN Note	LIN0EN	0	CAN0EN Note

AAUEN	Control of AAU input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by AAU. • AAU is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by AAU.

CAN1EN Note	Control of CAN1 input clock supply/control of CAN1 wakeup interrupt
0	Stops input clock supply. • Disables writing to the SFR used by CAN1. • CAN1 is in the reset state. Disables CAN1 wakeup interrupt.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by CAN1. Enables CAN1 wakeup interrupt.

LIN2EN Note	Control of LIN2 input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by LIN2. • LIN2 is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by LIN2.

LIN1EN Note	Control of LIN1 input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by LIN1. • LIN1 is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by LIN1.

LIN0EN	Control of LIN0 input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by LIN0. • LIN0 is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by LIN0.

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

CAN0EN Note	Control of CAN0 input clock supply/control of CAN0 wakeup interrupt
0	Stops input clock supply. • Disables writing to the SFR used by CAN0. • CAN0 is in the reset state. Disables CAN0 wakeup interrupt.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by CAN0. Enables CAN0 wakeup interrupt.

Note Only in the RL78/F25 products.

Caution Be sure to clear the following bits to 0.

- **Bits 0, 1, 3, 4, 5, and 7 in the RL78/F22 products**
- **Bits 1 and 7 in the RL78/F25 products**

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 5-11. 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 (See **Table 24-1** 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 0 (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/040C2H) 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/040C2H).

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/040C2H)).

Figure 5-12. 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/040C2H) 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			
			32 MHz base	40 MHz base	64 MHz base	80 MHz base
			FRQSEL4 = 0		FRQSEL4 = 1	
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 32 MHz	f_{IH} = 40 MHz	f_{IH} = 64 MHz	f_{IH} = 80 MHz
0	0	1	f_{IH} = 16 MHz	f_{IH} = 20 MHz	f_{IH} = 32 MHz	f_{IH} = 40 MHz
0	1	0	f_{IH} = 8 MHz	f_{IH} = 10 MHz	f_{IH} = 16 MHz	f_{IH} = 20 MHz
0	1	1	f_{IH} = 4 MHz	f_{IH} = 5 MHz	f_{IH} = 8 MHz	f_{IH} = 10 MHz
1	0	0	f_{IH} = 2 MHz	Setting prohibited	f_{IH} = 4 MHz	f_{IH} = 5 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.**

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 5-13. 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	HIOTRM6	HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0

HIOTRM6	HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0	High-speed on-chip oscillator
0	0	0	0	0	0	0	Minimum speed
0	0	0	0	0	0	1	
0	0	0	0	0	1	0	
0	0	0	0	0	1	1	
0	0	0	0	1	0	0	
			•				
			•				
			•				
1	1	1	1	1	1	0	
1	1	1	1	1	1	1	Maximum speed

Note The value after reset is the value adjusted at shipment.

Caution The HIOTRM register holds a 6-bit value used to adjust the high-speed on-chip oscillator. Increasing this value by 1 increases the frequency by approximately 0.016%.

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 5-14. Format of CAN Clock Select Register (CANCKSEL)

Address: F02C3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
CANCKSEL	0	0	0	0	0	0	CAN1MCKE	CAN0MCKE

CAN1MCKE	Control the supply or stop of X1 clock (f_X) to CAN1
0	Stops the supply of X1 clock (f_X) to CAN1
1	Enables the supply of X1 clock (f_X) to CAN1

CAN0MCKE	Control of supplying or stopping CAN0 X1 clock (f_X)
0	Stops the supply of X1 clock (f_X) to CAN0
1	Enables the supply of X1 clock (f_X) to CAN0

Remark This register is installed only in the RL78/F25 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 5-15. Format of LIN Clock Select Register (LINCKSEL)

Address: F02C4H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	3	<2>	<1>	<0>
LINCKSEL	0	LIN2MCKE Note	LIN1MCKE Note	LIN0MCKE	0	LIN2MCK Note	LIN1MCK Note	LIN0MCK

LIN2MCKE Note	Control of supplying or stopping LIN2 communication clock source
0	Stops LIN2 communication clock source supply
1	Enables LIN2 communication clock source supply

LIN1MCKE Note	Control of supplying or stopping LIN1 communication clock source
0	Stops LIN1 communication clock source supply
1	Enables LIN1 communication clock source supply

LIN0MCKE	Control of supplying or stopping LIN0 communication clock source
0	Stops LIN0 communication clock source supply
1	Enables LIN0 communication clock source supply

LIN2MCK Note	Control of selecting LIN2 communication clock source
0	Selects the f_{CLK} clock
1	Selects the f_{MX} clock

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/F25.

Cautions 1. Select the LINn (n = 0 to 2) operating clock with the LINnMCK bit before setting the LINnMCKE bit to 1.

2. When operating LINn in SNOOZE mode, set the LINnMCK bit to 0.

3. In case of LINnMCK is set to 1 (select f_{MX} clock), 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 RDe, clock output/buzzer output and PLL count source clock division, and PLL frequency division. 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 5-16. Format of Clock Select Register (CKSEL)

Address: F02CAH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	<2>	1	<0>
CKSEL	FPLLDIV	FMAINDIV1	FMAINDIV0	0	0	TRD_CKSEL	0	SELLOSC Notes 5, 6, 7

FPLLDIV	Control of PLL frequency division selection
0	The PLLDIV0 bit in the PLLCTL register is enabled.
1	No division

FMAINDIV1	FMAINDIV0	Control of PLL input clock (f_{PLLI}) division selection
0	0	No division
1	0	Divided by 2 (f_{MAIN} = 16 MHz input only)
1	1	Divided by 4 (f_{MAIN} = 20 MHz input only)
0	1	Setting prohibited

TRD_CKSEL	Control of TRDe clock selection
0	Selects f_{CLK} or f_{MP} Note 1
1	Selects f_{SL} Note 2

SELLOSC Notes 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 (f_{IH} = 80 MHz or 64 MHz) in the user option byte (000C2H/040C2H) or PLLDIV1 = 1 (f_{PLL} > 40 MHz) in the PLLCTL register, f_{MP} (f_{MP} > 40 MHz) can be used as the clock for timer RDe.

2. When f_{SL} is selected as the timer RDe 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. 24-pin 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, 5, 6, 8, 10, 12, 16, or 20 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 5-17. Format of PLL Control Register (PLLCTL)

Address: F02C8H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PLLCTL	LCKSEL1	LCKSEL0	PLLDIV1	PLLDIV0	PLLMULA	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	Selects 1024/f_{MAIN}.

PLLDIV1	Control of PLL output clock selection
0	When $f_{PLL} \leq 40$ MHz
1	When $f_{PLL} > 40$ 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})

PLLMULA	PLLMUL	Control of PLL multiplication selection
0	0	Multiplies the clock frequency by 12
0	1	Multiplies the clock frequency by 16
1	0	Multiplies the clock frequency by 10
1	1	Multiplies the clock frequency by 20

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 is 0).

2. When the clock monitor detects that the main system/PLL select clock has been stopped, the SELPLL bit is not automatically cleared.

Cautions

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 PLL clock has been stopped, even if the SELPLL bit is set to 1 (SELPLL = 1), the clock through mode continues. Reset the MCU to release the clock through mode, after the clock through mode is entered by the clock monitor function.**
5. **The counter for the lock-up wait time should be set to a period of at least 50 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 2 MHz to 40 MHz can be selected.**

PLLCTL Register				CKSEL Register			Inputtable Frequency (f_{MAIN}) Note	PLL input clock (f_{PLLI}) Note	Multiplication	Division	Outputtable Frequency (f_{PLL}) Note
PLLMULA	PLLMUL	PLLDIV1	PLLDIV0	FMAINDIV1	FMAINDIV0	FPLLDIV					
1	1	0	0	0	0	0	4 MHz ± 2%	4 MHz ± 2%	× 20	1/2	40 MHz ± 2%
1	1	1	0	0	0	1	4 MHz ± 2%	4 MHz ± 2%	× 20	1/1	80 MHz ± 2%
0	0	0	1	0	0	0	8 MHz ± 2%	8 MHz ± 2%	× 12	1/4	24 MHz ± 2%
0	1	0	1	0	0	0	8 MHz ± 2%	8 MHz ± 2%	× 16	1/4	32 MHz ± 2%
1	0	0	0	0	0	0	8 MHz ± 2%	8 MHz ± 2%	× 10	1/2	40 MHz ± 2%
0	0	1	0	0	0	0	8 MHz ± 2%	8 MHz ± 2%	× 12	1/2	48 MHz ± 2%
0	1	1	0	0	0	0	8 MHz ± 2%	8 MHz ± 2%	× 16	1/2	64 MHz ± 2%
1	0	1	0	0	0	1	8 MHz ± 2%	8 MHz ± 2%	× 10	1/1	80 MHz ± 2%
0	0	0	1	1	0	0	16 MHz ± 2%	8 MHz ± 2%	× 12	1/4	24 MHz ± 2%
0	1	0	1	1	0	0	16 MHz ± 2%	8 MHz ± 2%	× 16	1/4	32 MHz ± 2%
1	0	0	0	1	0	0	16 MHz ± 2%	8 MHz ± 2%	× 10	1/2	40 MHz ± 2%
0	0	1	0	1	0	0	16 MHz ± 2%	8 MHz ± 2%	× 12	1/2	48 MHz ± 2%
0	1	1	0	1	0	0	16 MHz ± 2%	8 MHz ± 2%	× 16	1/2	64 MHz ± 2%
1	0	1	0	1	0	1	16 MHz ± 2%	8 MHz ± 2%	× 10	1/1	80 MHz ± 2%
0	1	1	0	1	1	1	20 MHz ± 2%	5 MHz ± 2%	× 16	1/1	80 MHz ± 2%
0	1	0	0	1	1	0	20 MHz ± 2%	5 MHz ± 2%	× 16	1/2	40 MHz ± 2%
Other than above							Setting prohibited				

Note These accuracy are design target.

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/040C2H), set the PLLDIV1 bit to 0 ($f_{PLL} \leq 40$ MHz).**
13. **To change the SELPLL bit from 1 to 0 while PLLDIV1 = 1 ($f_{PLL} > 40$ MHz), before changing the SELPLL bit, stop counting by the timer RDe (setting the TSTART0 and TSTART1 bits in the TRDSTR register to 0) and stop the CAN (setting the CAN0EN and CAN1EN bit in the PER2 register to 0).**
14. **Do not change the value of the LCKSEL2, LCKSEL1, LCKSEL0, PLLDIV1, PLLDIV0, PLLMULA, and PLLMUL bit while the PLLON bit is set to 1.**

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}) 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 5-18. Format of PLL Status Register (PLLSTS)

Address: F02C9H 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 5-19. Format of f_{MP} Clock Division Register (MDIV)

Address: F02CBH 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	f_{MP} clock division control
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/040C2H) 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 2 MHz to 40 MHz.**
2. **When FRQSEL4 = 1 in the user option byte (000C2H/040C2H), set the MDIV2 to MDIV0 bits to 001B (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} > 40 MHz), set the MDIV2 to MDIV0 bits to 001B (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 80 MHz or 64 MHz is selected as f_{IH}, the initial setting of the MDIV register is "division by 2" so that f_{CLK} is set to 40 MHz or 32 MHz, respectively.**

5.3.16 A/D Conversion Clock Control Register (ADCKS)

The ADCKS register is the register that sets the division ratio of the peripheral hardware clock frequency (f_{CLK}), which is the clock source of ADCLK. Set this register before starting A/D conversion.

This register can be set by an 8-bit memory manipulation instruction.

Writing to the ADCKS register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation sets this register to 00H.

Figure 5-20. Format of A/D Conversion Clock Control Register (ADCKS)

Address: F02C7H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADCKS	0	0	0	0	0	0	ADCK[1:0]	

ADCK[1:0]	Selects A/D conversion clock
00B	Non divided (f_{CLK})
01B	Divided by 2 (f_{CLK}/2)
10B	Divided by 4 (f_{CLK}/4)
11B	Divided by 8 (f_{CLK}/8)

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Cautions 1. Settings such that the frequency of the A/D conversion clock (ADCLK) is less than 2 MHz are prohibited.

2. When using the low-current conversion mode, set the A/D clock frequency in the range of 2 MHz to 8 MHz.

5.4 System Clock Oscillator

5.4.1 X1 Oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (2 MHz 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.

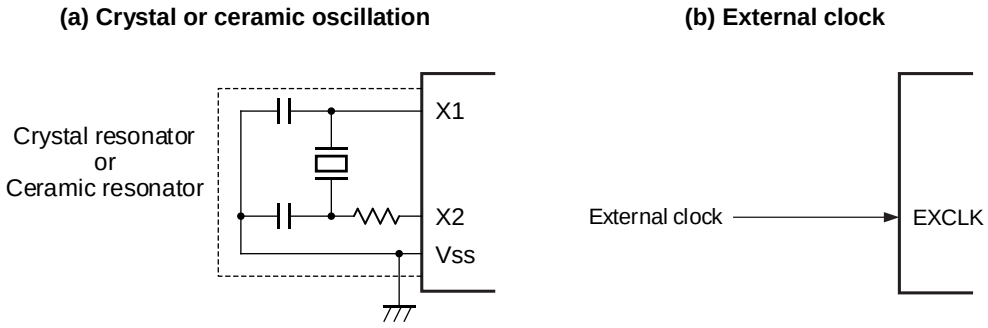
- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

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 Recommended Connection of Unused Pins**.

Figure 5-21 shows an example of the external circuit of the X1 oscillator.

Figure 5-21. Example of External Circuit of X1 Oscillator



(**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.

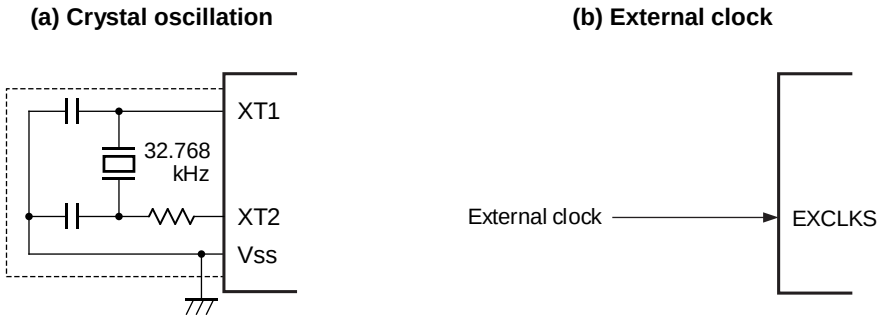
- Crystal oscillation: EXCLKS, OSCSELS = 0, 1
- External clock input: EXCLKS, OSCSELS = 1, 1

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 Recommended Connection of Unused Pins**.

Figure 5-22 shows an example of the external circuit of the XT1 oscillator.

Figure 5-22. Example of External Circuit of XT1 Oscillator



Caution When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in Figure 5-21 and Figure 5-22 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.**

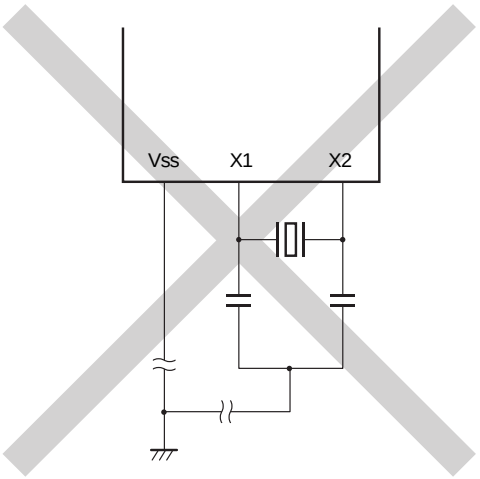
The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **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.**

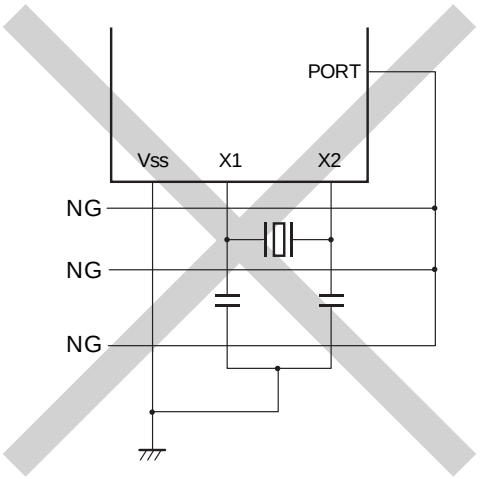
Figure 5-23 shows examples of incorrect resonator connection.

Figure 5-23. Examples of Incorrect Resonator Connection (1/2)

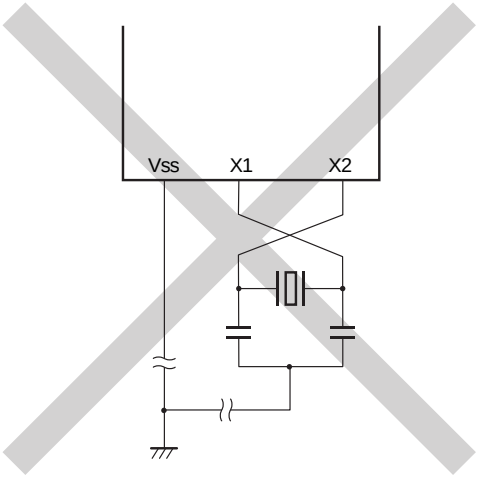
(a) Too long wiring



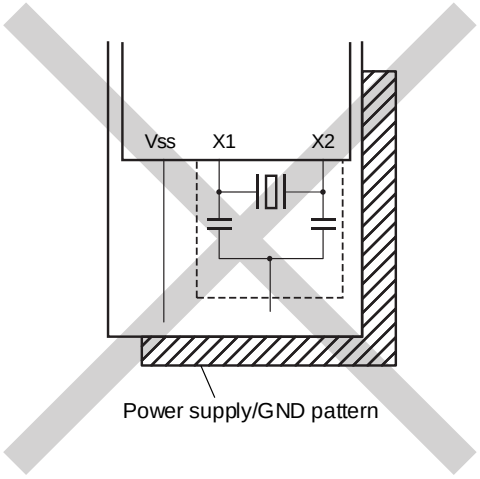
(b) Crossed signal line



(c) The X1 and X2 signal line wires cross.



(d) A power supply/GND pattern exists under the X1 and X2 wires.

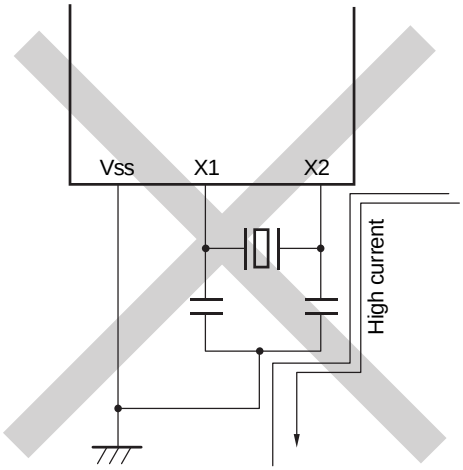


Caution Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board. Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

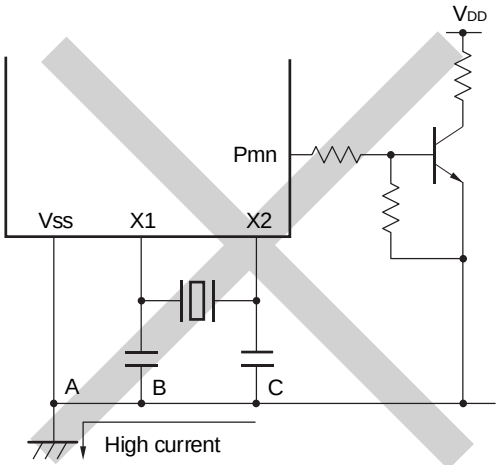
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 5-23. Examples of Incorrect Resonator Connection (2/2)

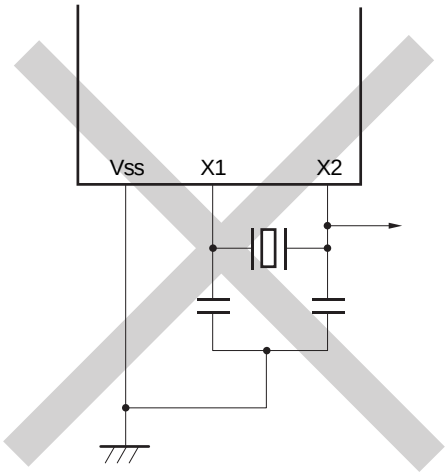
(e) Wiring near high alternating current



(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/F22 and RL78/F25. The frequency can be selected from among 80, 64, 40, 32, 20, 16, 8, 4, or 2 MHz by using the user option byte (000C2H/040C2H). When 80 MHz or 64 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/F22 and RL78/F25. 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/F22 and RL78/F25.

5.4.6 WDT-Dedicated Low-Speed On-chip Oscillator

The WDT-dedicated low-speed on-chip oscillator is incorporated in the RL78/F22 and RL78/F25.

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/040C0H) 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**).

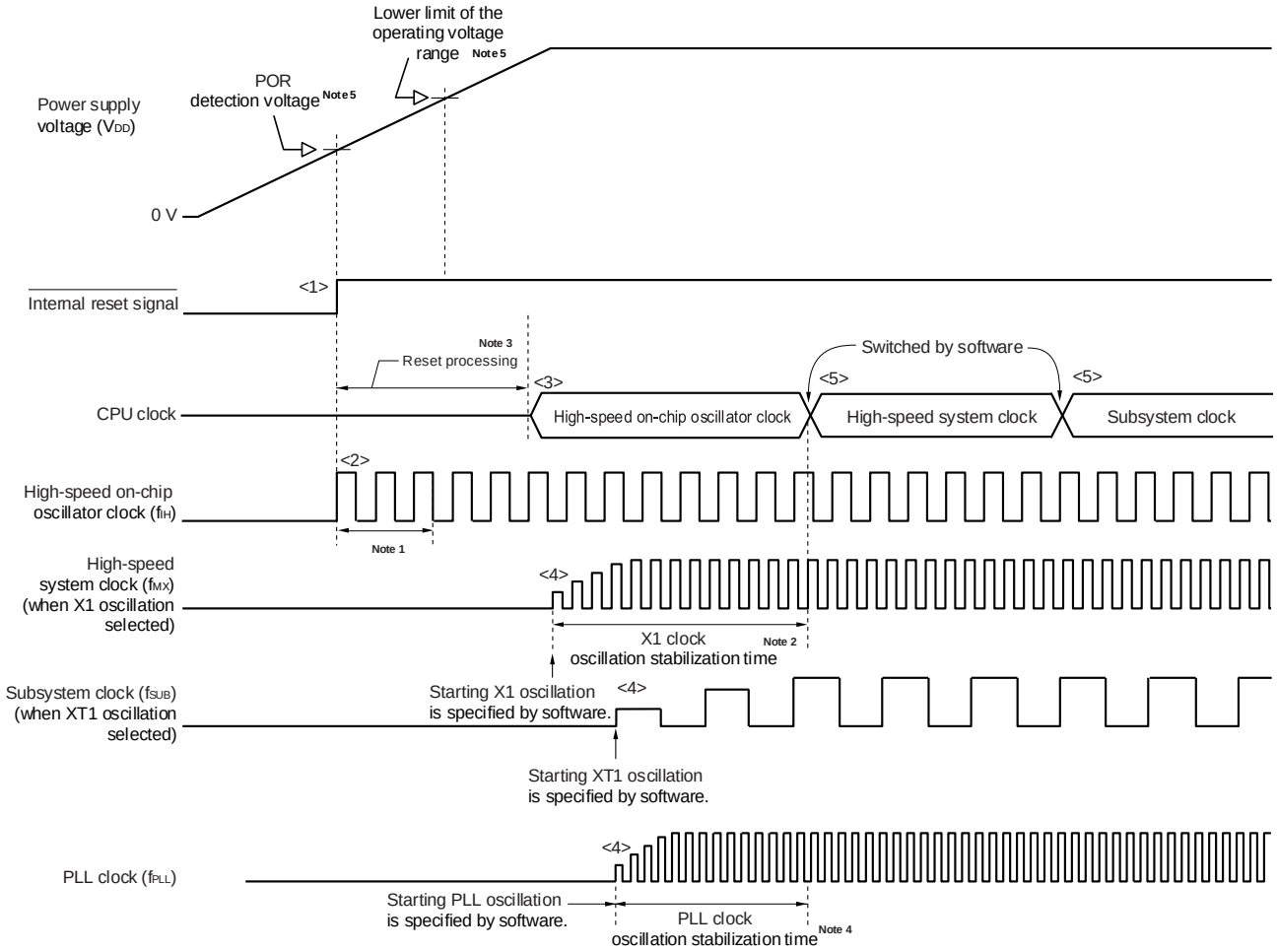
- Main system clock (f_{MAIN})
  - High-speed system clock (f_{MX})
    - X1 clock (f_X)
    - External main system clock (f_{EX})
  - High-speed on-chip oscillator clock (f_{IH})
- Subsystem clock (f_{SUB})
  - XT1 clock (f_{XT})
  - External subsystem clock (f_{EXS})
- PLL clock (f_{PLL})
- Low-speed 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/F22 and RL78/F25.

When the power supply voltage is turned on, the clock generator operation is shown in **Figure 5-24**.

Figure 5-24. 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 POR detection voltage **Note 5**, 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 by 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 26 POWER-ON-RESET (POR) CIRCUIT**.

4. When the PLL circuit starts operation, time is required so that the PLL circuit becomes locked (LOCK = 1).

5. See **CHAPTER 37** and **CHAPTER 38 ELECTRICAL SPECIFICATIONS** for details.

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 80, 64, 40, 32, 20, 16, 8, 4, and 2 MHz by using FRQSEL0 to FRQSEL4 of the user option byte (000C2H/040C2H). In addition, Oscillation can be changed by the internal high-speed on-chip oscillator frequency select register (HOCODIV).

[User Option Byte Setting]

Address: 000C2H/040C2H After reset: - (user setting value)

Symbol	7	6	5	4	3	2	1	0
User option byte	1	1	RESOUTB 0/1	FRQSEL4 0/1	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
					f_{IH}
1	1	0	0	0	80 MHz
1	0	0	0	0	64 MHz
0	1	0	0	0	40 MHz
0	0	0	0	0	32 MHz
0	1	0	0	1	20 MHz
0	0	0	0	1	16 MHz
0	0	0	1	0	8 MHz
0	0	0	1	1	4 MHz
0	0	1	0	0	2 MHz
Other than above					Setting prohibited

[High-speed On-chip Oscillator Frequency Select Register (HOCODIV) Setting]

Address: F00A8H After reset: value set with the FRQSEL2 to FRQSEL0 bits of the user option byte (000C2H/040C2H) 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			
			32 MHz base	40 MHz base	64 MHz base	80 MHz base
			FRQSEL4 = 0		FRQSEL4 = 1	
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 32 MHz	f_{IH} = 40 MHz	f_{IH} = 64 MHz	f_{IH} = 80 MHz
0	0	1	f_{IH} = 16 MHz	f_{IH} = 20 MHz	f_{IH} = 32 MHz	f_{IH} = 40 MHz
0	1	0	f_{IH} = 8 MHz	f_{IH} = 10 MHz	f_{IH} = 16 MHz	f_{IH} = 20 MHz
0	1	1	f_{IH} = 4 MHz	f_{IH} = 5 MHz	f_{IH} = 8 MHz	f_{IH} = 10 MHz
1	0	0	f_{IH} = 2 MHz	Setting prohibited	f_{IH} = 4 MHz	f_{IH} = 5 MHz
1	0	1	Setting prohibited	Setting prohibited	f_{IH} = 2 MHz	Setting prohibited
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 10 MHz, in such cases set the AMPH bit to 1, to operate the X1 oscillator.

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS		AMPHS1	AMPHS0	AMPH
	0	1	0	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 µs is set based on a 10 MHz resonator.

Symbol	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.

Symbol	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 µs is set based on a 10 MHz resonator.

Symbol	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> Set (1) the MCM0 bit of the CKC register to specify the X1 oscillation clock as the CPU/peripheral hardware clock.

Symbol	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).

Symbol	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 is stopped.

Symbol	7	6	5	4	3	2	1	0
OSMC	RTCLPC 0/1	0	0	WUTMMCK0 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.

Symbol	7	6	5	4	3	2	1	0
CKSEL	FPLLDIV 0	FMAINDIV1 0	FMAINDIV0 0	0	0	TRD_CKSEL 0	0	SELLOSC 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.

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 0	EXCLKS 0	OSCSELS 1	0	AMPHS1 0/1	AMPHS0 0/1	AMPH 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.

Symbol	7	6	5	4	3	2	1	0
CSC	MSTOP 0	XTSTOP 0	0	0	0	0	0	HIOSTOP 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).

Symbol	7	6	5	4	3	2	1	0
CKC	CLS 0	CSS 1	MCS 0	MCM0 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.

Symbol	7	6	5	4	3	2	1	0
CKC	CLS 1	CSS 1	MCS 0	MCM0 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 ≤ 40 MHz, clear the PLLDIV1 bit to 0. When PLL clock > 40 MHz, set the PLLDIV1 bit to 1.

Symbol	7	6	5	4	3	2	1	0
PLLCTL	LCKSEL1 0/1	LCKSEL0 0/1	PLLDIV1 0/1	PLLDIV0 0/1	PLLMULA 0/1	SELPLL 0	PLLMUL 0/1	PLLON 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 50 µs. When the PLL source clock (f_{MAIN}) is 4 MHz, set the LCKSEL1 and LCKSEL0 bits to 01B or 10B. When the PLL source clock (f_{MAIN}) is 8 MHz, set the LCKSEL1 and LCKSEL0 bits to 10B. When the PLL source clock (f_{MAIN}) is 16 MHz or 20 MHz, set the LCKSEL1 and LCKSEL0 bits to 11B.

<3> Select the frequency division of the PLL input clock with the FMAINDIV1 bit and the FMAINDIV0 bit of the CKSEL register.

When FMAINDIV1 = 1 and FMAINDIV0 = 0, the PLL input clock division ratio is 2.
When FMAINDIV1 = 1 and FMAINDIV0 = 1, the PLL input clock division ratio is 4.
When FMAINDIV1 = 0 and FMAINDIV0 = 0, the PLL input clock division ratio is not division.

<4> Select the frequency division of the PLL clock with the PLLDIV0 bit of the PLLCTL register and the FPLLDIV bit in the CKSEL register.

When PLLDIV0 = 0 and FPLLDIV = 0, the PLL division ratio is 2.
When PLLDIV0 = 1 and FPLLDIV = 0, the PLL division ratio is 4.
When FPLLDIV = 1, the PLL division ratio is not division.

<5> Select the multiplication value of the PLL clock with the PLLMUL bit and the PLLMULA bit of the PLLCTL register.

When PLLMULA = 0 and PLLMUL = 0, the PLL multiplication value is 12.
When PLLMULA = 0 and PLLMUL = 1, the PLL multiplication value is 16.
When PLLMULA = 1 and PLLMUL = 0, the PLL multiplication value is 10.
When PLLMULA = 1 and PLLMUL = 1, the PLL multiplication value is 20.

<6> Wait for the selection of the PLL multiplication value to become effective. After setting the PLLMUL bit and the PLLMULA bit, wait for at least 1 µs.

<7> Set the PLLON bit of the PLLCTL register to 1 to start oscillation of the PLL clock.

Symbol	7	6	5	4	3	2	1	0
PLLCTL	LCKSEL1 0/1	LCKSEL0 0/1	PLLDIV1 0/1	PLLDIV0 0/1	PLLMULA 0/1	SELPLL 0	PLLMUL 0/1	PLLON 1

<8> Confirm that the PLL circuit is locked (LOCK = 1) with the LOCK bit of the PLLSTS register.

Symbol	7	6	5	4	3	2	1	0
PLLSTS	LOCK 1	0	0	0	SELPLLS 0	0	0	0

<9> Set the PLL clock between 2 MHz and 40 MHz with the MDIV bits of the MDIV register.
Example: To select $f_{MP}/2$, set the following value.

Symbol	7	6	5	4	3	2	1	0
MDIV	0	0	0	0	0	MDIV2 0	MDIV1 0	MDIV0 1

<10> 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 ($f_{MP} = f_{PLL}$).

Symbol	7	6	5	4	3	2	1	0
PLLCTL	LCKSEL1 0/1	LCKSEL0 0/1	PLLDIV1 0/1	PLLDIV0 0/1	PLLMULA 0/1	SELPLL 1	PLLMUL 0/1	PLLON 1

<11> Confirm that the PLL-clock-selected mode is selected (SELPLLS = 1) with the SELPLLS bit of the PLLSTS register.

Symbol	7	6	5	4	3	2	1	0
PLLSTS	LOCK 1	0	0	0	SELPLLS 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 ($f_{PLL} = f_{MAIN}$).

Symbol	7	6	5	4	3	2	1	0
PLLCTL	LCKSEL1 0/1	LCKSEL0 0/1	PLLDIV1 0/1	PLLDIV0 0/1	PLLMULA 0/1	SELPLL 0	PLLMUL 0/1	PLLON 1

<2> Confirm that the clock through mode is selected (SELPLLS = 0) with the SELPLLS bit of the PLLSTS register.

Symbol	7	6	5	4	3	2	1	0
PLLSTS	LOCK 0/1	0	0	0	SELPLLS 0	0	0	0

<3> Clear the PLLON bit of the PLLCTL register to 0 to stop oscillation of the PLL clock.

Symbol	7	6	5	4	3	2	1	0
PLLCTL	LCKSEL1 0/1	LCKSEL0 0/1	PLLDIV1 0/1	PLLDIV0 0/1	PLLMULA 0/1	SELPLL 0	PLLMUL 0/1	PLLON 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.

Symbol	7	6	5	4	3	2	1	0
CKSEL	FPLLDIV	FMAINDIV1	FMAINDIV0			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).

Symbol	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).

Symbol	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0				
	0	1	0	1	0	0	0	0

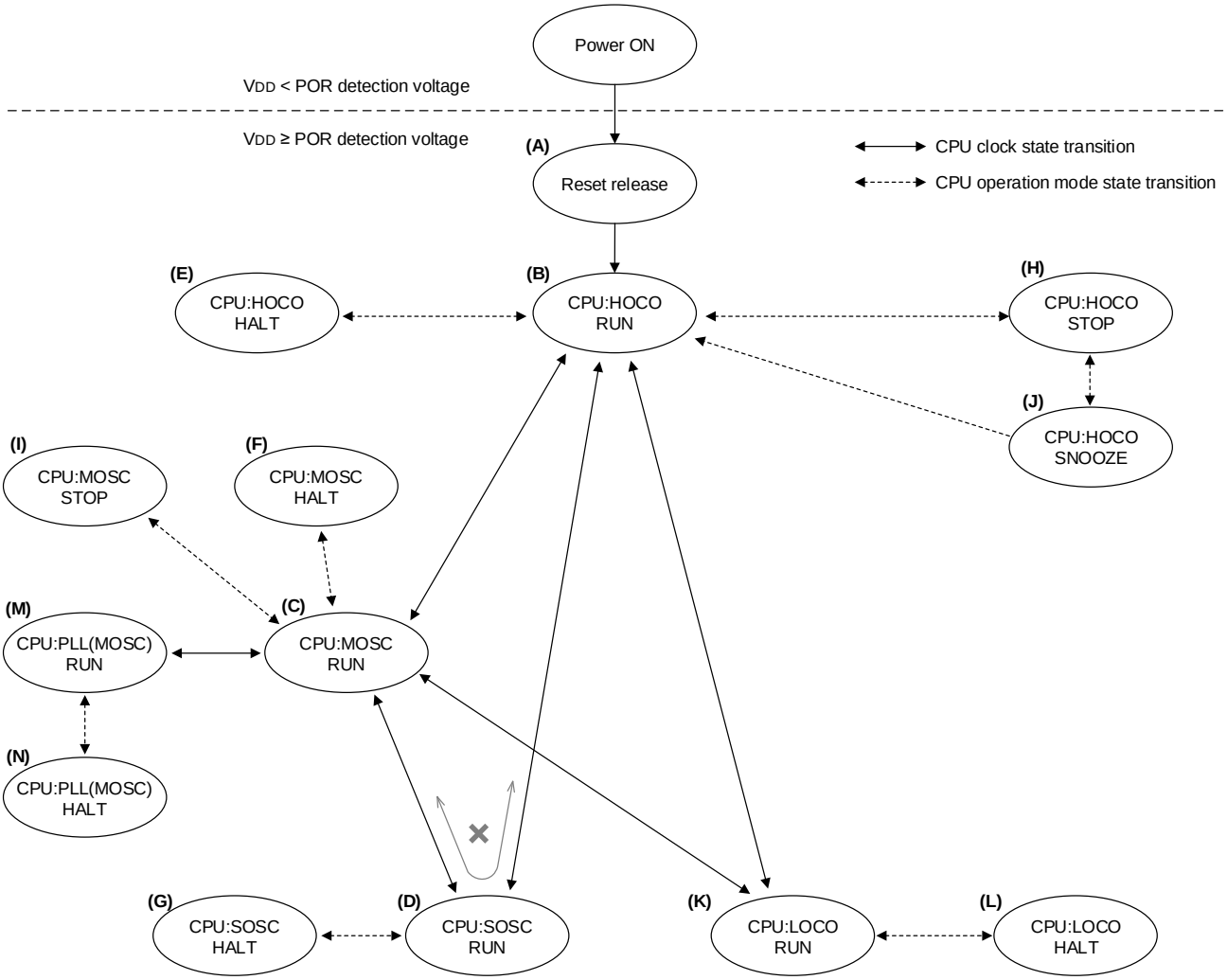
<4> Confirm that f_{SL} (low-speed on-chip oscillator) is selected as the CPU/peripheral hardware clock (CLS = 1) with the CLS bit of the CKC register.

Symbol	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 5-25 shows the CPU clock status transition diagram of this product.

Figure 5-25. CPU Clock Status Transition Diagram



Caution Transitions in the order of "(B)→(D)→(C)" or "(C)→(D)→(B)" are prohibited.

Remark HOCO: High-speed on-chip oscillator
LOCO: Low-speed on-chip oscillator
MOSC: Main oscillator (High-speed system clock oscillator **Note 1**)
SOSC: Sub oscillator (Subsystem clock oscillator **Note 1**)
PLL(xxxx): Clock generated by a PLL circuit using xxxx as the clock source **Note 2**.

Notes 1. See **Figure 5-1. Block Diagram of Clock Generator** for details on each clock generation circuit.
2. See **Figure 5-2. Block Diagram of PLL Circuit** for details of PLL circuit.

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 (K).

- 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 system clock (C) to operating with the PLL clock (M).

- Set the PLLCTL register (PLLDIV1 = x, LCKSEL[1:0] = xx, PLLDIV0 = x, PLLMUL = x, PLLMULA = x) and CKSEL register (FPLLDIV = x, FMAINDIV1 = x, FMAINDIV0 = x).
- Wait for the selection of the PLL multiplication value to become effective (After setting the PLLMUL bit and PLLMULA 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), wait at least 40 μs for oscillation accuracy stabilization time by software before changing the clock.

(7) Change the CPU from operating with the subsystem clock (D) or operating with the low-speed on-chip oscillator clock (K) 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), wait at least 40 μs for oscillation accuracy stabilization time by software before changing the clock.

(8) Change the CPU from operating with the PLL clock (M) to operating with the high-speed system clock (C).

- 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 (K) 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 (M) to HALT mode (N).
- The CPU changes from operating with the low-speed on-chip oscillator clock (K) to HALT mode (L).
  - 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 **24.3.3 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 24 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 5-4. 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. • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	The operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1) after checking that the CPU clock is changed.
	External main system clock	External clock input from the EXCLK pin is enabled. • 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. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • SELLOSC = 0 • 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. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0 • 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. • OSCSELS = 1, SELLOSC = 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**.

Table 5-4. 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. • HIOSTOP = 0	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	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. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • SELLOSC = 0 • After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	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. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0 • 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. • OSCSELS = 1, SELLOSC = 1	
	PLL clock	PLL oscillation is stable. • 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 5-4. 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. • HIOSTOP = 0	The external main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	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. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • SELLOSC = 0 • After elapse of oscillation stabilization time	The external main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	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. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0 • 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. • OSCSELS = 1, SELLOSC = 1	
	PLL clock	PLL oscillation is stable. • 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 5-4. 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. • HIOSTOP = 0, MCS = 0	XT1 oscillation can be stopped (XTSTOP = 1) after checking that the CPU clock is changed.
	X1 clock	X1 oscillation is stable, and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • 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. • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • 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 5-4. 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. • HIOSTOP = 0, MCS = 0	The external subsystem clock input can be disabled (XTSTOP = 1) after checking that the CPU clock is changed.
	X1 clock	X1 oscillation is stable, and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • 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. • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • 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 5-4. 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. • HIOSTOP = 0, MCS = 0	The low-speed on-chip oscillator can be stopped (SELLOSC = 0, WUTMMCK0 = 0) after checking that the CPU clock is changed.
	X1 clock	X1 oscillation is stable, and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • 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. • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • SELLOSC = 1, 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 5-4. Changing CPU Clock (7/7)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
PLL clock	X1 clock	X1 oscillation is stable, and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	The PLL clock can be stopped (PLLON = 0) after checking that the CPU clock is changed.
	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. • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • 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 5-5. 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 5-6)
f_{IH}	⟷	f_{MX}	Type 2 (Table 5-7, Table 5-8)
f_{MP}	⟷	f_{SL}	Type 3 (Table 5-9)
f_{MAIN}	⟷	f_{PLL}	Type 4 (Table 5-10)

Table 5-6. 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 5-7. Maximum Time Required for Type 2 (1) Note

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 For $f_{IH} \leq 40$ MHz

Table 5-8. Maximum Time Required for Type 2 (2) Note

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 For f_{IH} > 40 MHz

Table 5-9. 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 5-10. 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 Table 5-6 to Table 5-10 is the number of CPU clocks before switchover.

2. Calculate the number of clocks in Table 5-6 to Table 5-10 by rounding up the number after the decimal position.

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.

Before stopping the clock oscillation, check the conditions before the clock oscillation is stopped.

Table 5-11. 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 (The CPU is operating on a clock other than the high-speed on-chip oscillator clock.)	HIOSTOP = 1
X1 clock External main system clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
PLL clock	SELPLLS = 0 (The CPU is operating on a clock other than the PLL clock.)	PLLON = 0
XT1 clock External subsystem clock	CLS = 0 (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 (The CPU is operating on a clock other than the subsystem/low-speed on-chip oscillator clock.)	SELLOSC = 0 and 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

When the FRQSEL3 bit is set to 0 (high-speed on-chip oscillator = 64/32/16/8/4/2 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 40 MHz.

CHAPTER 6 TIMER ARRAY UNIT

The number of units or channels of the timer array unit differs, depending on the product.

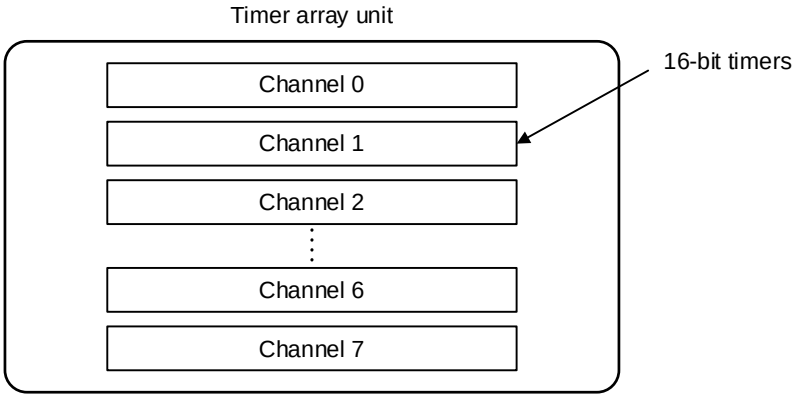
Units	Channels	RL78/F25	RL78/F22
Unit 0	Channel 0	√	√
	Channel 1	√	√
	Channel 2	√	√
	Channel 3	√	√
	Channel 4	√	√
	Channel 5	√	√
	Channel 6	√	√
	Channel 7	√	√
Unit 1	Channel 0	√	√
	Channel 1	√	√
	Channel 2	√	√
	Channel 3	√	√
	Channel 4	√	–
	Channel 5	√	–
	Channel 6	√	–
	Channel 7	√	–

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 RL78/F25 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**) • Square wave output (→ refer to **6.7.1**) • External event counter (→ refer to **6.7.2**) • Divider function (→ refer to **6.7.3**) • Input pulse interval measurement (→ refer to **6.7.4**) • Measurement of high-/low-level width of input signal (→ refer to **6.7.5**) • Delay counter (→ refer to **6.7.6**)	• One-shot pulse output (→ refer to **6.8.1**) • PWM output (→ refer to **6.8.2**) • 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

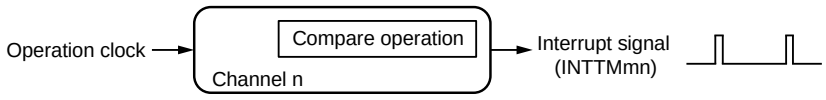
Timer array unit has the following functions.

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.

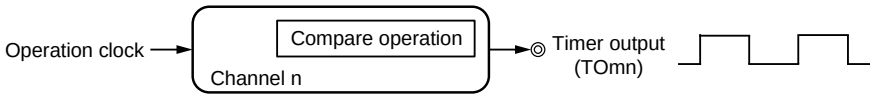
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



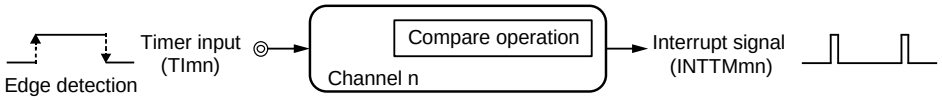
(2) Square wave output

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).



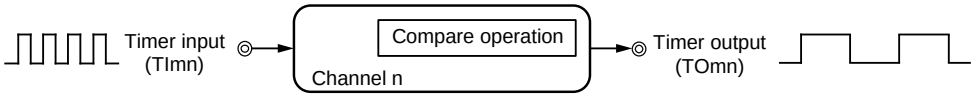
(3) External event counter

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

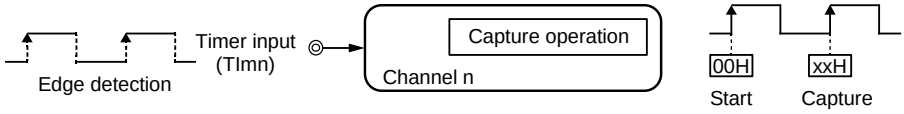
A clock input from a timer input pin (TImn) is divided and output from an output pin (TOmn).



Set the TImn and TOmn pins so that they are different from each other by the peripheral I/O redirection registers 0, 1, 2, 3, and 9 (PIOR0, PIOR1, PIOR2, PIOR3, and PIOR9).

(5) Input pulse interval measurement

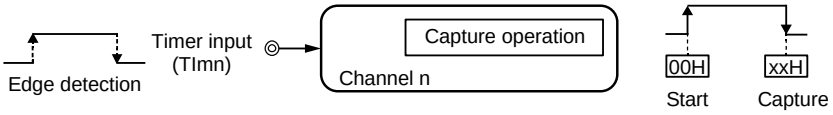
Counting is started by the valid edge of a pulse signal input to a timer input pin (TImn). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



(**Remarks** are listed on the next page.)

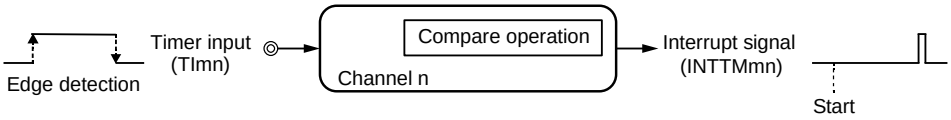
(6) Measurement of high-/low-level width of input signal

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)

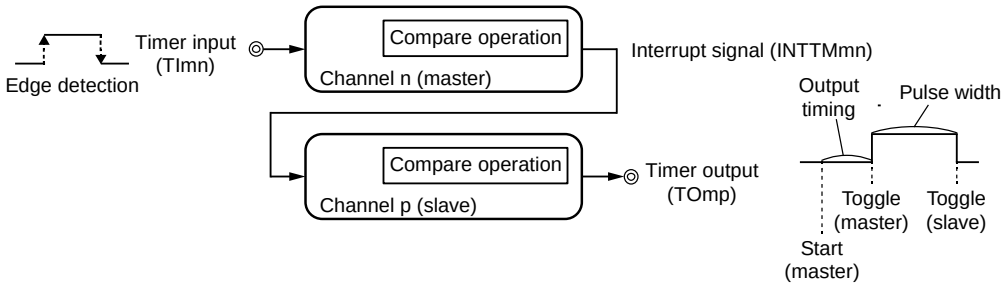
2. The presence or absence of timer I/O pins of channels 0 to 7 depends on the product. See **Table 6-2. Timer I/O Pins provided in Each Product** for details.

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.

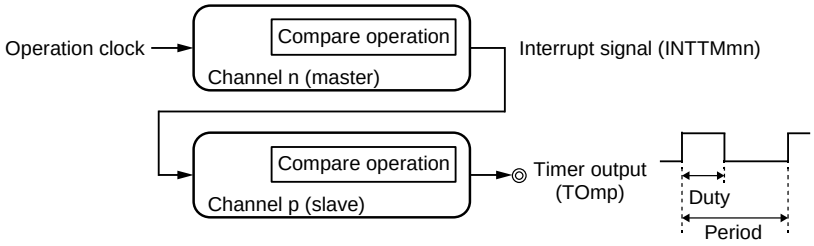
(1) One-shot pulse output

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

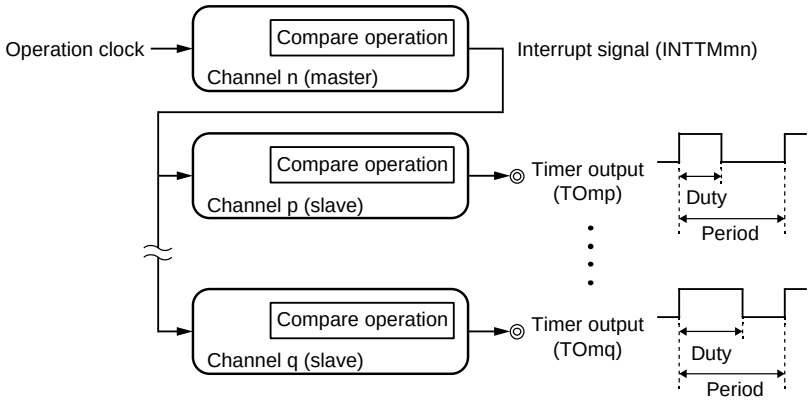
Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



(**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> • Peripheral enable register 0 (PER0) • Timer clock select register m (TPSm) • 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 3
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register mn (TSRmn) • Noise filter enable registers 1, 2 (NFEN1, NFEN2) • Port mode register (PMxx) Note 2 • 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 0, 1, 3, 4, 5, 6, 7, 12 (PM0, PM1, PM3, PM4, PM5, PM6, PM7, PM12)**.

3. Bit allocation differs depending on the product.

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 Note 3	
		RL78/F25	RL78/F22
Unit 0	Channel 0	P17/TI00/TO00, P66/(TI00)/(TO00)	P17/TI00/TO00
	Channel 1	P30/TI01/TO01, P126/(TI01)/(TO01), P60/(TO01)	P30/TI01/TO01, P60/(TO01) Note 1
	Channel 2	P16/TI02/TO02, P67/(TI02)/(TO02), P61/(TO02)	P16/TI02/TO02, P61/(TO02) Note 1
	Channel 3	P125/TI03/TO03, P127/(TI03)/(TO03), P62/(TO03)	P125/TI03/TO03 Note 1, P62/(TO03)
	Channel 4	P13/TI04/TO04, P01/(TI04)/(TO04)	P13/TI04/TO04
	Channel 5	P15/TI05/TO05, P00/(TI05)/(TO05)	P15/TI05/TO05, P00/(TI05)/(TO05) Note 2
	Channel 6	P14/TI06/TO06, P02/(TI06)/(TO06)	P14/TI06/TO06
	Channel 7	P120/TI07/TO07, P44/(TI07)/(TO07), P63/(TO07)	P120/TI07/TO07, P63/(TO07)
Unit 1	Channel 0	P41/TI10/TO10, P45/(TI10)/(TO10)	P41/TI10/TO10 Note 1
	Channel 1	P12/TI11/TO11, P54/(TI11)/(TO11)	P12/TI11/TO11
	Channel 2	P11/TI12/TO12, P46/(TI12)/(TO12)	P11/TI12/TO12
	Channel 3	P10/TI13/TO13, P55/(TI13)/(TO13), P120/(TO13)	P10/TI13/TO13 Note 1, P120/(TO13)
	Channel 4	P31/TI14/TO14, P64/(TI14)/(TO14)	×
	Channel 5	P70/TI15/TO15, P56/(TI15)/(TO15)	×
	Channel 6	P32/TI16/TO16, P65/(TI16)/(TO16)	×
	Channel 7	P71/TI17/TO17, P57/(TI17)/(TO17)	×

Notes 1. TAU functions using these terminals are not available in RL78/F22 24-pin products. Therefore, functions that use these pins cannot be used with the RL78/F22 24-pin products.

2. TAU functions using these terminals are not available in RL78/F22 32- and 24-pin products. Therefore, functions that use these pins cannot be used with the RL78/F22 32- and 24-pin products.

3. The I/O ports to which TImn and TOmn are assigned can be selected by setting the peripheral I/O redirection registers 0, 1, 2, 3, 9 (PIOR0, 1, 2, 3, 9).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

See the section below for more information on the PIOR0, 1, 2, 3, 9 registers.

4.3.9 Peripheral I/O redirection register 0 (PIOR0)
4.3.10 Peripheral I/O redirection register 1 (PIOR1)
4.3.11 Peripheral I/O redirection register 2 (PIOR2)
4.3.12 Peripheral I/O redirection register 3 (PIOR3)
4.3.18 Peripheral I/O redirection register 9 (PIOR9)

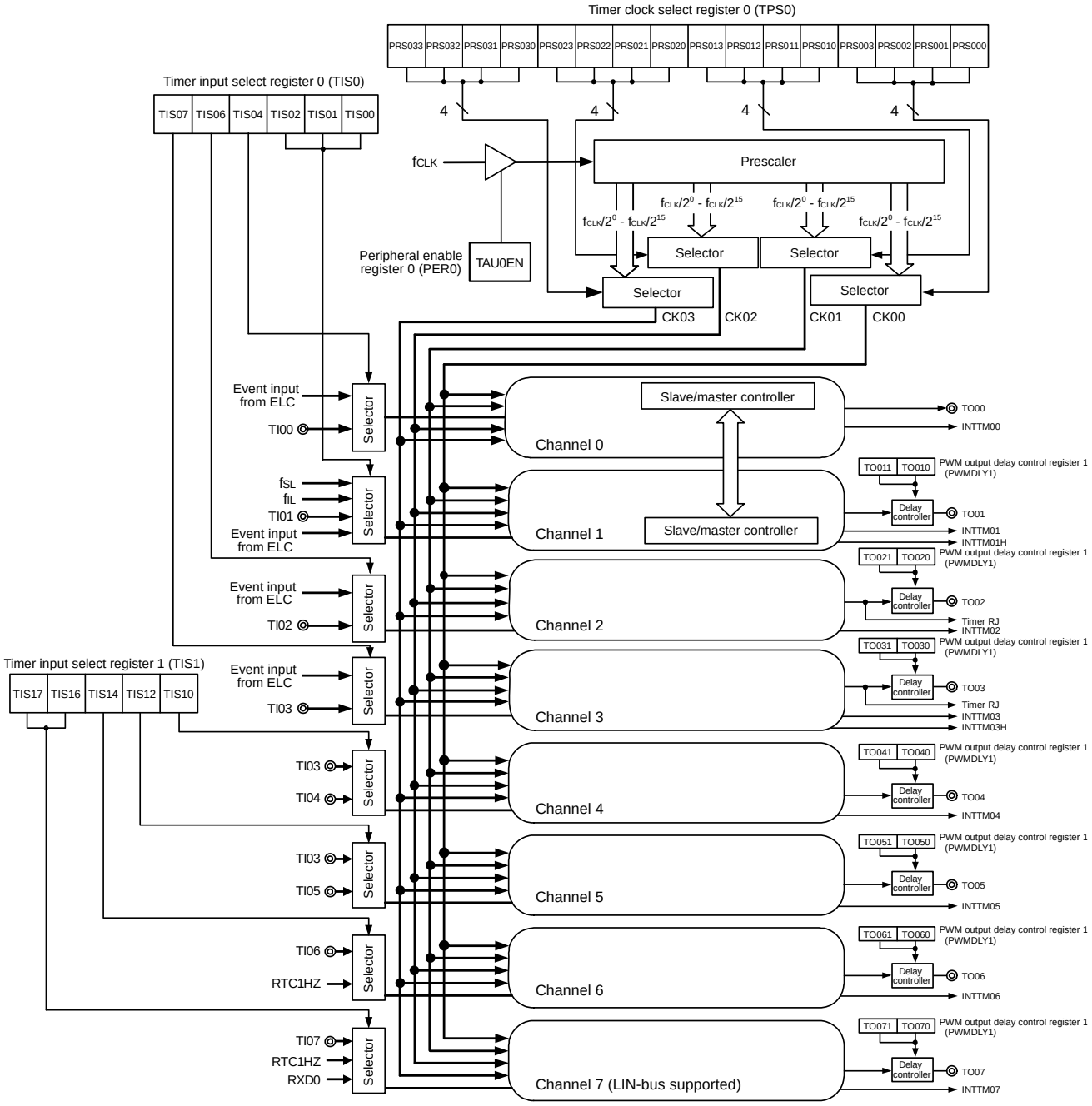
Remarks 1. Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection registers (PIORx).

2. When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

3. ×: The channel is not available.

Figure 6-1 to Figure 6-3 show the block diagrams of the timer array unit.

Figure 6-1. Entire Configuration of Timer Array Unit 0 (Example: RL78/F25 100-pin products)



Remark f_{SL}: Sub/low-speed on-chip oscillator clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 6-2. Entire Configuration of Timer Array Unit 1 (Example: RL78/F22 48-pin products)

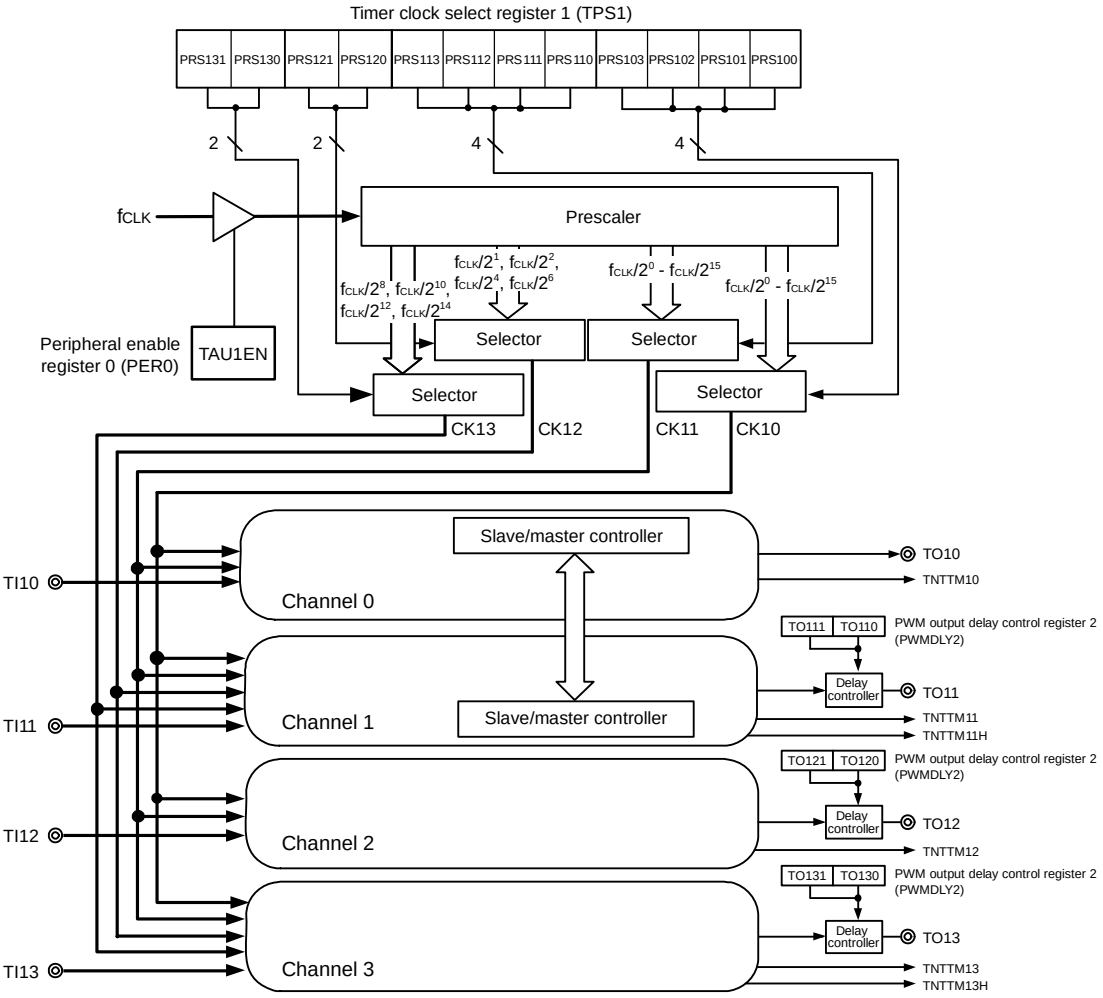


Figure 6-3. Entire Configuration of Timer Array Unit 1 (Example: RL78/F25 100-pin products)

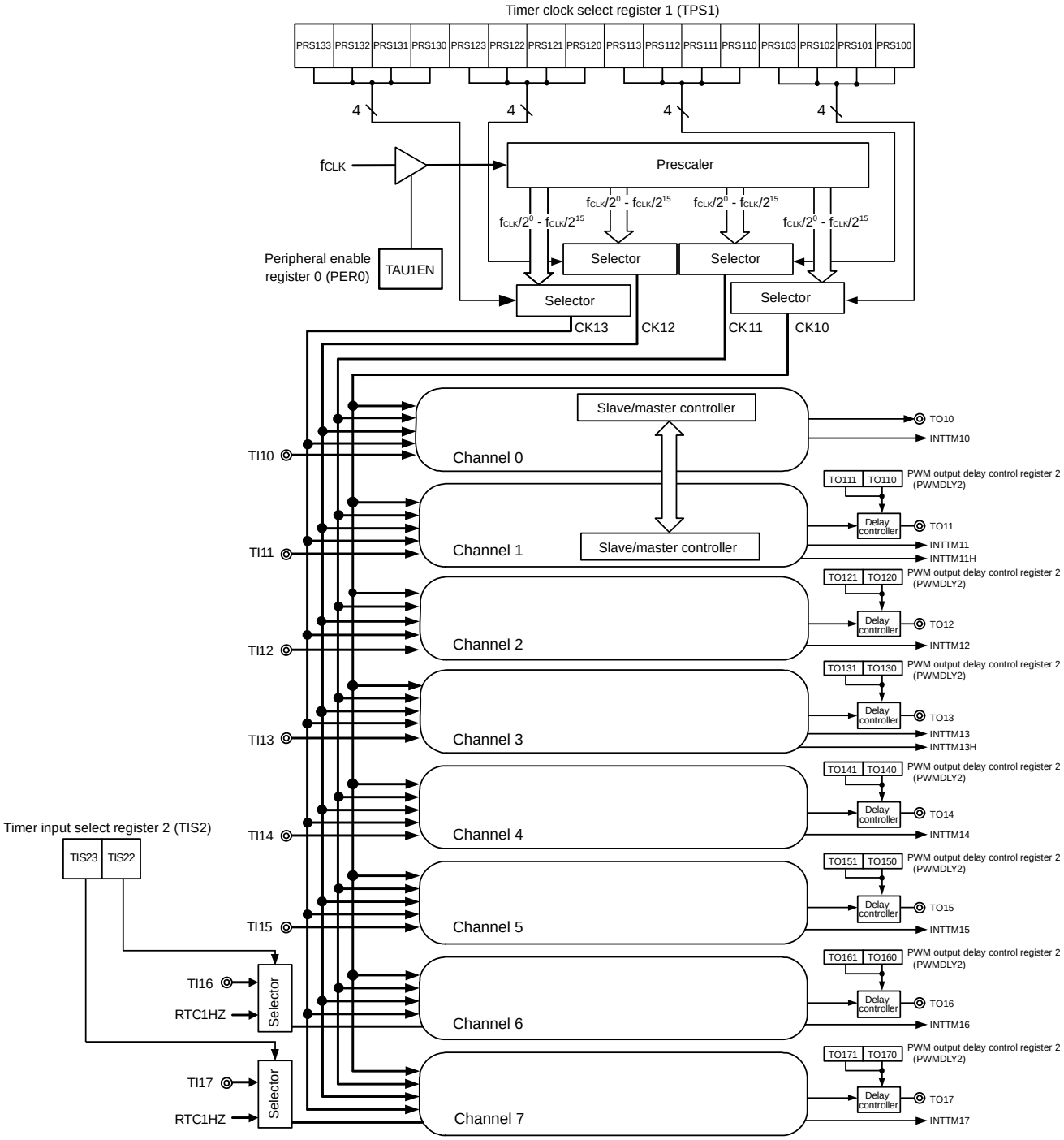
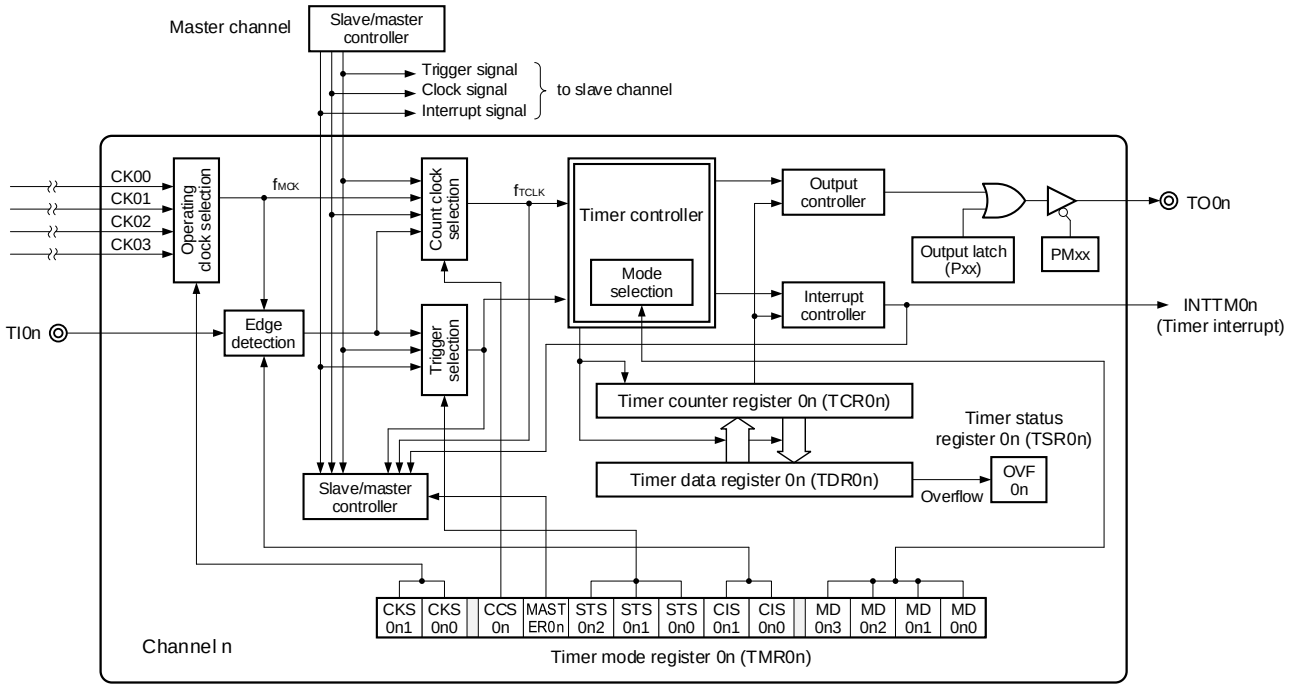
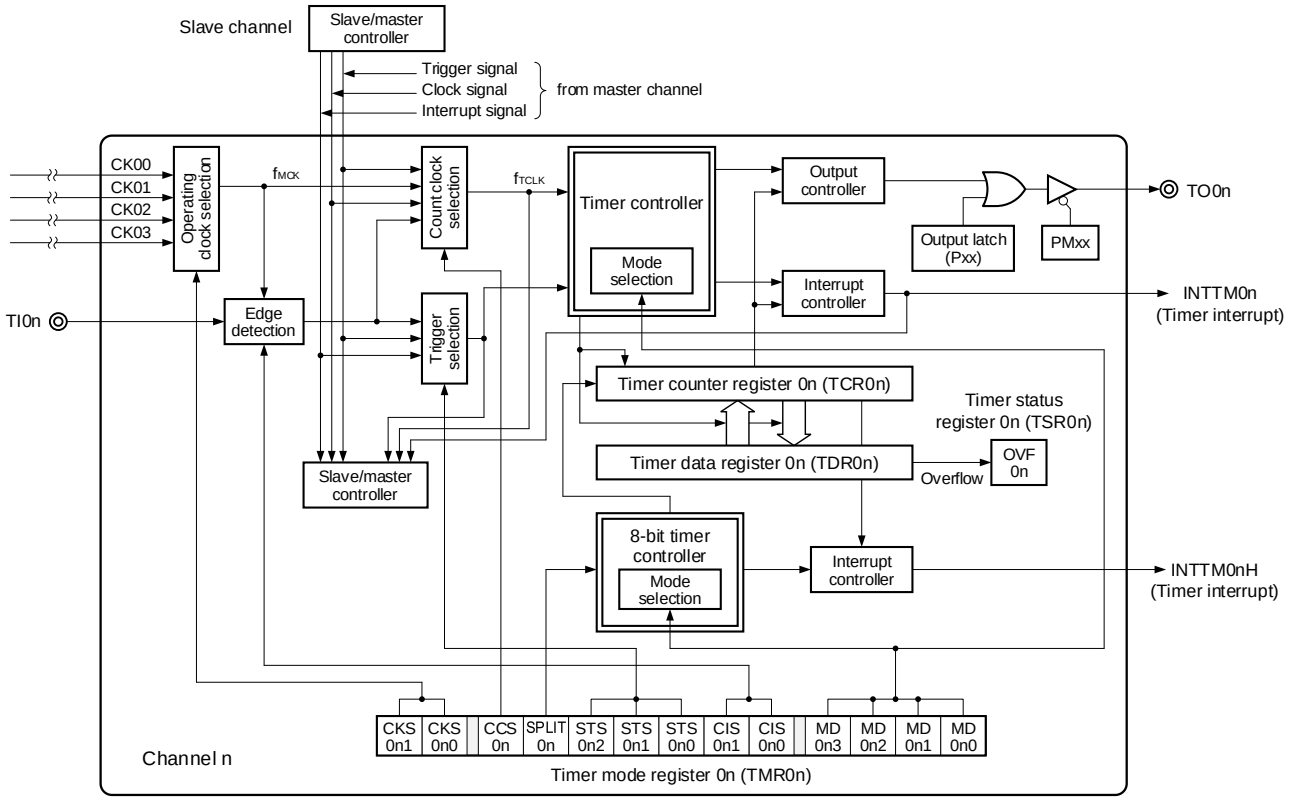


Figure 6-4. Internal Block Diagram of Channels 0, 2, 4, 6 of Timer Array Unit 0



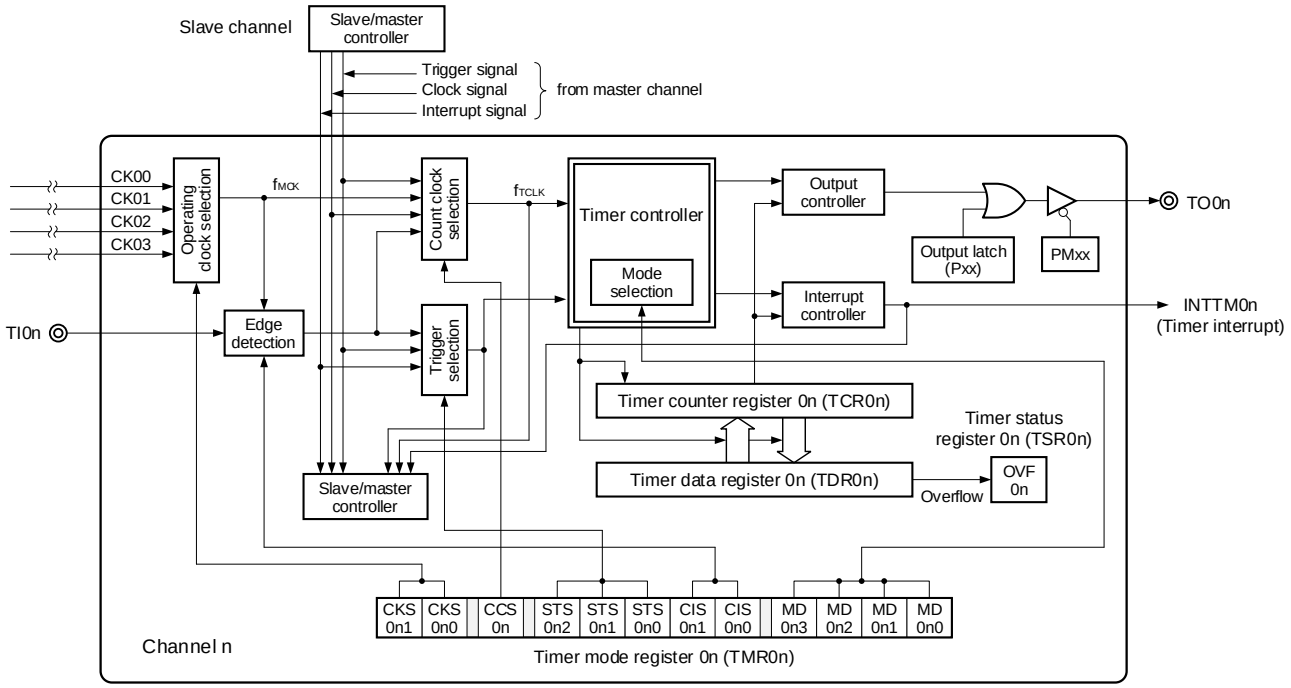
Remark n = 0, 2, 4, 6

Figure 6-5. Internal Block Diagram of Channels 1, 3 of Timer Array Unit 0



Remark n = 1, 3

Figure 6-6. Internal Block Diagram of Channels 5, 7 of Timer Array Unit 0



Remark n = 5, 7

6.2.1 Timer Count Register mn (TCRmn)

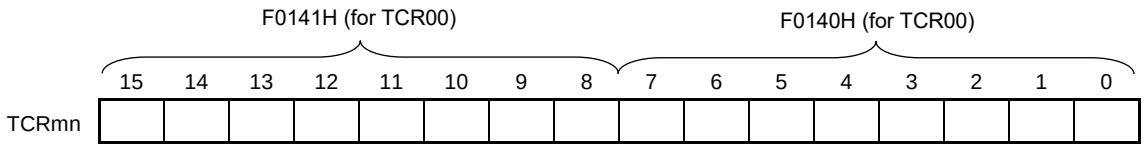
The TCRmn register is a 16-bit read-only register and is used to count clocks.

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: F0140H, F0141H (TCRm0) to F014EH, F014FH (TCRm7), After reset: FFFFH R



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).

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 SPLITm1, SPLITm3 bits of timer mode registers m1 and m3 (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), FFF74H, FFF75H (TDR02), After reset: 0000H R/W
FFF78H, FFF79H (TDR04) to FFF7EH, FFF7FH (TDR07)
FFF80H, FFF81H (TDR10), FFF84H, FFF85H (TDR12)
FFF88H, FFF89H (TDR14) to FFF8EH, FFF8FH (TDR17)

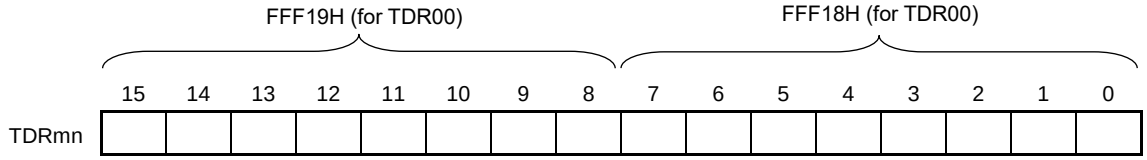
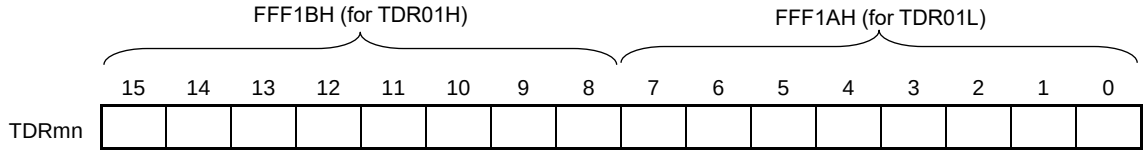


Figure 6-9. Format of Timer Data Register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF76H, FFF77H (TDR03), After reset: 0000H R/W
FFF82H, FFF83H (TDR11), FFF86H, FFF87H (TDR13)



(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

Caution When accessing a register of TAUm (m = 0, 1), select the target unit number m with the UTSEL register before accessing.

Table 6-4. Timer Array Unit Register Configuration (1/2)

Address	Register Name (m = 0, 1)	Symbol (m = 0, 1)		After Reset	Access Size
F0071H	Noise filter enable register 1	NFEN1		00H	1, 8
F0072H	Noise filter enable register 2	NFEN2		00H	1, 8
F0074H	Timer input select register 0	TIS0		00H	8
F0075H	Timer input select register 1	TIS1		00H	8
F007AH	Timer input select register 2	TIS2		00H	8
F02C0H	Peripheral enable register 0	PER0		00H	1, 8
F01F0H	Unit select register	UTSEL		00H	1, 8
F0140H	Timer count register m0	TCRm0		FFFFH	16
F0142H	Timer count register m1	TCRm1		FFFFH	16
F0144H	Timer count register m2	TCRm2		FFFFH	16
F0146H	Timer count register m3	TCRm3		FFFFH	16
F0148H	Timer count register m4	TCRm4		FFFFH	16
F014AH	Timer count register m5	TCRm5		FFFFH	16
F014CH	Timer count register m6	TCRm6		FFFFH	16
F014EH	Timer count register m7	TCRm7		FFFFH	16
F0150H	Timer mode register m0	TMRm0		0000H	16
F0152H	Timer mode register m1	TMRm1		0000H	16
F0154H	Timer mode register m2	TMRm2		0000H	16
F0156H	Timer mode register m3	TMRm3		0000H	16
F0158H	Timer mode register m4	TMRm4		0000H	16
F015AH	Timer mode register m5	TMRm5		0000H	16
F015CH	Timer mode register m6	TMRm6		0000H	16
F015EH	Timer mode register m7	TMRm7		0000H	16
F0160H	Timer status register m0	TSRm0L	TSRm0	0000H	8, 16
F0161H		–			
F0162H	Timer status register m1	TSRm1L	TSRm1	0000H	8, 16
F0163H		–			
F0164H	Timer status register m2	TSRm2L	TSRm2	0000H	8, 16
F0165H		–			
F0166H	Timer status register m3	TSRm3L	TSRm3	0000H	8, 16
F0167H		–			
F0168H	Timer status register m4	TSRm4L	TSRm4	0000H	8, 16
F0169H		–			
F016AH	Timer status register m5	TSRm5L	TSRm5	0000H	8, 16
F016BH		–			
F016CH	Timer status register m6	TSRm6L	TSRm6	0000H	8, 16
F016DH		–			
F016EH	Timer status register m7	TSRm7L	TSRm7	0000H	8, 16
F016FH		–			

(**Remarks** are listed on the next page.)

Table 6-4. Timer Array Unit Register Configuration (2/2)

Address	Register Name (m = 0, 1)	Symbol (m = 0, 1)		After Reset	Access Size
F0170H	Timer channel enable status register m	TEmL	TEm	0000H	1, 8, 16
F0171H		–			
F0172H	Timer channel start register m	TSmL	TSm	0000H	1, 8, 16
F0173H		–			
F0174H	Timer channel stop register m	TTmL	TTm	0000H	1, 8, 16
F0175H		–			
F0176H	Timer clock select register m	TPSm		0000H	16
F0178H	Timer output register m	TOmL	TOm	0000H	8, 16
F0179H		–			
F017AH	Timer output enable register m	TOEmL	TOEm	0000H	1, 8, 16
F017BH		–			
F017CH	Timer output level register m	TOLmL	TOLm	0000H	8, 16
F017DH		–			
F017EH	Timer output mode register m	TOMmL	TOMm	0000H	8, 16
F017FH		–			
F022AH	PWM output delay control register 1	PWMDLY1		0000H	16
F022CH	PWM output delay control register 2	PWMDLY2		0000H	16
FFF18H	Timer data register 00	TDR00		0000H	16
FFF1AH	Timer data register 01	TDR01L	TDR01	0000H	8, 16
FFF1BH		TDR01H			
FFF74H	Timer data register 02	TDR02		0000H	16
FFF76H	Timer data register 03	TDR03L	TDR03	0000H	8, 16
FFF77H		TDR03H			
FFF78H	Timer data register 04	TDR04		0000H	16
FFF7AH	Timer data register 05	TDR05		0000H	16
FFF7CH	Timer data register 06	TDR06		0000H	16
FFF7EH	Timer data register 07	TDR07		0000H	16
FFF80H	Timer data register 10	TDR10		0000H	16
FFF82H	Timer data register 11	TDR11L	TDR11	0000H	8, 16
FFF83H		TDR11H			
FFF84H	Timer data register 12	TDR12		0000H	16
FFF86H	Timer data register 13	TDR13L	TDR13	0000H	8, 16
FFF87H		TDR13H			
FFF88H	Timer data register 14	TDR14		0000H	16
FFF8AH	Timer data register 15	TDR15		0000H	16
FFF8CH	Timer data register 16	TDR16		0000H	16
FFF8EH	Timer data register 17	TDR17		0000H	16

Remarks 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 0, 1, 3, 4, 5, 6, 7, 12 (PM0, PM1, PM3, PM4, PM5, PM6, PM7, PM12)**.

2. m: Unit number (m = 0, 1), The unit number is selected by the TSEL0 bit of the UTSEL register.

3. n: Channel number (n = 0 to 7), Channel numbers 4 to 7 (n = 4 to 7) of unit 1 are not provided in RL78/F22 products.

6.3.1 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 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: F02C0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

TAU1EN	Control of timer array unit 1 input clock supply
0	Stops input clock supply. • SFR used by the timer array unit 1 cannot be written. • The timer array unit 1 is in the reset status.
1	Enables input clock supply. • SFR used by the timer array unit 1 can be read/written.

TAU0EN	Control of timer array unit 0 input clock supply
0	Stops input clock supply. • SFR used by the timer array unit 0 cannot be written. • The timer array unit 0 is in the reset status.
1	Enables input clock supply. • SFR used by the timer array unit 0 can be read/written.

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 0, 1, 3, 4, 5, 6, 7, 12 (PM0, PM1, PM3, PM4, PM5, PM6, PM7, PM12), port registers 0, 1, 3, 4, 5, 6, 7, 12 (P0, P1, P3, P4, P5, P6, P7, P12), and PWM output delay control registers 1, 2 (PWMDLY1, PWMDLY2)).

2. Bit 6 must be set to 0.

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 RL78/F22 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 RL78/F22 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 RL78/F25 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: F0176H, F0177H (TPSm) 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	f_{CLK} = 40 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz	40 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz	20 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz	10 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz	5 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz	2 MHz	2.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz	1 MHz	1.25 MHz
0	1	1	0	$f_{CLK}/2^6$	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz	500 kHz	625 kHz
0	1	1	1	$f_{CLK}/2^7$	15.62 kHz	39.1 kHz	78.1 kHz	156.2 kHz	250 kHz	312.5 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz	156.2 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.76 kHz	19.5 kHz	39.1 kHz	62.5 kHz	78.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz	31.25 kHz	39.1 kHz
1	0	1	1	$f_{CLK}/2^{11}$	976 Hz	2.44 kHz	4.88 kHz	9.76 kHz	15.63 kHz	19.5 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz	9.76 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz	4.88 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz	2.44 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	976 Hz	1.22 kHz

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. This format cannot be applied to unit 1 of the RL78/F22 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: F0176H, F0177H 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	f_{CLK} = 40 MHz
0	0	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz	20 MHz
0	1	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz	10 MHz
1	0	$f_{CLK}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz	2 MHz	2.5 MHz
1	1	$f_{CLK}/2^6$	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz	500 kHz	625 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	f_{CLK} = 40 MHz
0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz	156.2 kHz
0	1	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz	31.25 kHz	39.1 kHz
1	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz	9.76 kHz
1	1	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz	2.44 kHz

Note The above format is applied to the TPS1 of the RL78/F22 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 unit 0 for RL78/F22 and RL78/F25 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 6-5 can be achieved by using the interval timer function.

Table 6-5. Interval Times Available for Operation Clock CKSm2 or CKSm3

Clock		Interval time (f_{CLK} = 40 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: F0150H, F0151H (TMRm0) to F015EH, F015FH (TMRm7), 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

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 When using unit 0: In channels 0 to 3, Valid edge of input signal selected by TIS0 In channels 4 to 7, Valid edge of input signal selected by TIS1 When using unit 1: In channels 6 and 7, Valid edge of input signal selected by TIS2
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 RL78/F22 products.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

2. TMR17 to TMR14 are not provided in the RL78/F22 products.

Figure 6-13. Format of Timer Mode Register mn (TMRmn) (2/4)

Address: F0150H, F0151H (TMRm0) to F015EH, F015FH (TMRm7), 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	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). 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). 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. (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. TMR17 to TMR14 are not provided in the RL78/F22 products.

Figure 6-13. Format of Timer Mode Register mn (TMRmn) (3/4)

Address: F0150H, F0151H (TMRm0) to F015EH, F015FH (TMRm7), 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) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) 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. TMR17 to TMR14 are not provided in the RL78/F22 products.

Figure 6-13. Format of Timer Mode Register mn (TMRmn) (4/4)

Address: F0150H, F0151H (TMRm0) to F015EH, F015FH (TMRm7), 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	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 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) • 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 not 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** and **Remarks** are listed on the next page.)

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. TMR17 to TMR14 are not provided in the RL78/F22 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 **Figure 6-6** 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 read 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: F0160H, F0161H (TSRm0) to F016EH, F016FH (TSRm7), After reset: 0000H R

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. TSR17 to TSR14 are not provided in the RL78/F22 products.

Table 6-6. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVF bit	Set/clear conditions
• Capture mode • Capture & one-count mode	clear	When no overflow has occurred upon capturing
	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • 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 read 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: F0170H, F0171H 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 RL78/F22 products.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

2. TE17 to TE14 are not provided in the RL78/F22 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: F0172H, F0173H7 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. The TCRm3 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-7** 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. The TCRm1 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-7** 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. The TCRmn register count operation start in the count operation enabled state varies depending on each operation mode (see **Table 6-7** in **6.5.2 Start Timing of Counter**).. 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 RL78/F22 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. TS17 to TS14 are not provided in the RL78/F22 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: F0174H, F0175H 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	TEHm3 bit is cleared to 0 and the count operation is stopped.

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	TEHm1 bit is cleared to 0 and the count operation is stopped.

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	TEmn bit is cleared to 0 and the count operation is stopped. 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 RL78/F22 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. TT17 to TT14 are not provided in the RL78/F22 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 1	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 2

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 2

TIS04 Note 1	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 2

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 2
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 RL78/F25 products. Write “0” when writing to the timer input select register 0 (TIS0) of RL78/F22 products.

2. Provided only in RL78/F25 products. Do not set in the RL78/F22 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.

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.

Reset signal generation clears this register to 00H.

This function is valid only for the RL78/F25 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.

3. This register is not provided in the RL78/F22 products.

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: F017AH, F017BH 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. Timer operation is not applied to the TOmn bit and the output is fixed. Writing to the TOmn bit is enabled.
1	Timer output is enabled. Timer operation is applied to the TOmn bit and an output waveform is generated. Writing to the TOmn bit is ignored.

Caution Be sure to clear bits 15 to 8 to “0”. (unit 1, 0: RL78/F25 products)
Be sure to clear bits 15 to 8 of unit 0 and bits 15 to 4 of unit 1 to “0”. (RL78/F22 products)

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

2. TOE17 to TOE14 are not provided in the RL78/F22 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 of 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: F0178H, F0179H 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: RL78/F25 products)
Be sure to clear bits 15 to 8 of unit 0 and bits 15 to 4 of unit 1 to “0”. (RL78/F22 products)

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

2. TO17 to TO14 are not provided in the RL78/F22 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: F017CH, F017DH 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: RL78/F25 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”. (RL78/F22 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. TOL17 to TOL14 are not provided in the RL78/F22 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: F017EH, F017FH 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 (INTTMmp) of the slave channel)

Caution Be sure to clear bits 15 to 8 and 0 to “0”. (unit 1, 0: RL78/F25 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”. (RL78/F22 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. TOM17 to TOM14 are not provided in the RL78/F22 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, 2 (NFEN1, NFEN2) (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 1, 2 (NFEN1, 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. Bits 7 to 4 are not provided in the RL78/F22 products.

6.3.16 Port Mode Registers 0, 1, 3, 4, 5, 6, 7, 12 (PM0, PM1, PM3, PM4, PM5, PM6, PM7, PM12)

These registers set input/output of ports 0, 1, 3, 4, 5, 6, 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.

RL78/F25 products: PM0, PM1, PM3, PM4, PM5, PM6, PM7, PM12
RL78/F22 products: PM0, PM1, PM3, PM4, PM5, PM6, 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 PM0, PM1, PM3, PM4, PM5, PM6, 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 (PMm) (RL78/F25 100-pin products)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	1	PM03	PM02	PM01	PM00

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	1	1	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: 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 = 0, 1, 3, 4, 5, 6, 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 0, 1, 3, 4, 5, 6, 7, and 12 of the RL78/F25 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.

Figure 6-27. Format of PWM Output Delay Control Register 1 (PWMDLY1)

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	TO031 Note	TO030 Note	TO021 Note	TO020 Note	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.

Figure 6-28. Format of PWM Output Delay Control Register 2 (PWMDLY2)

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 8 are not provided in the RL78/F22 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).

6.3.19 Unit Select Register (UTSEL)

The UTSEL register is used to select the target unit for accessing the serial array unit (SAU) and timer array unit (TAU) registers.

Set the UTSEL register by a 1-bit or 8-bit memory manipulation instruction.

The TAU0 and TAU1 registers are assigned to the same address.

Before accessing the TAU0 or TAU1 registers, select the unit to be accessed with the TSEL0 bit of the UTSEL register.

Figure 6-29. Format of Unit Select Register (UTSEL)

Address: F01F0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
UTSEL	0	0	0	0	0	SSEL0	0	TSEL0

SSEL0	Serial array unit selection
0	SAU0 is selected.
1	SAU1 is selected.

TSEL0	Timer array unit selection
0	TAU0 is selected.
1	TAU1 is selected.

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.

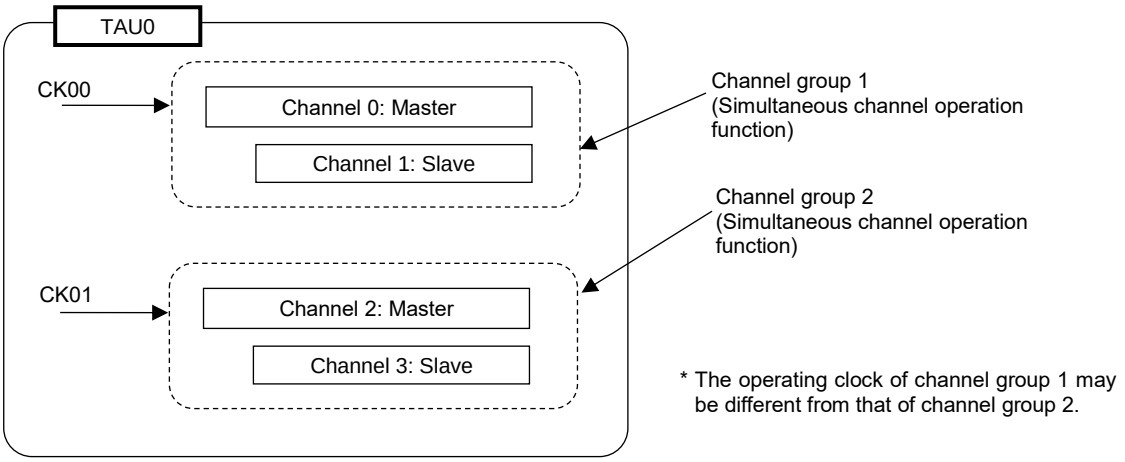
(1) Only an even channel (channel 0, 2, 4, 6) can be set as a master channel.
(2) Any channel, except channel 0, can be set as a slave channel.
(3) The slave channel must be lower than the master channel.
Example: If channel 2 is set as a master channel, channel 3 or those that follow (channels 3, 4, 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) CKm2/CKm3 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.
(14) 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).

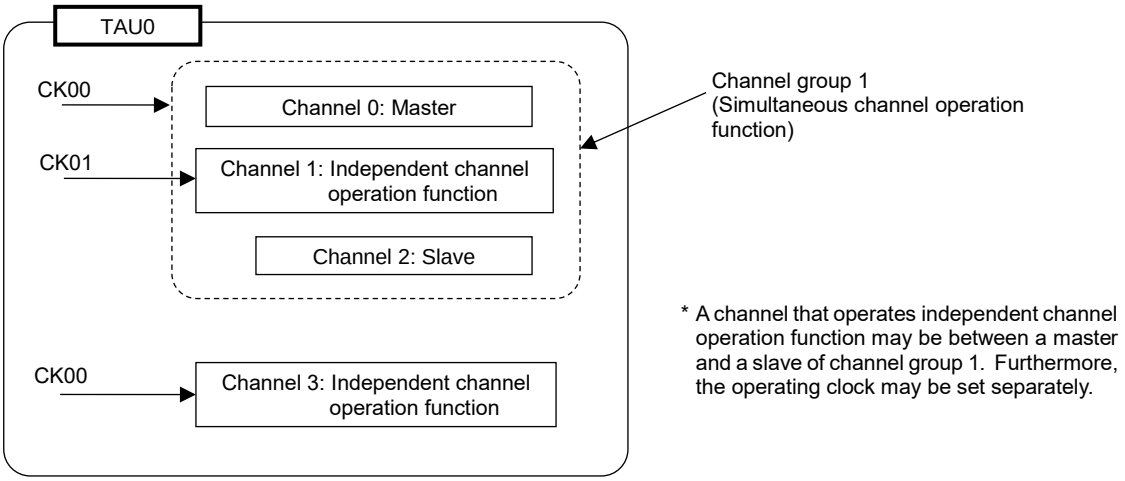
If two or more channel groups that do not operate in combination are specified, the basic rules described above do not apply to the channel groups.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)
2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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 SPLITmn 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.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3)

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 the 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

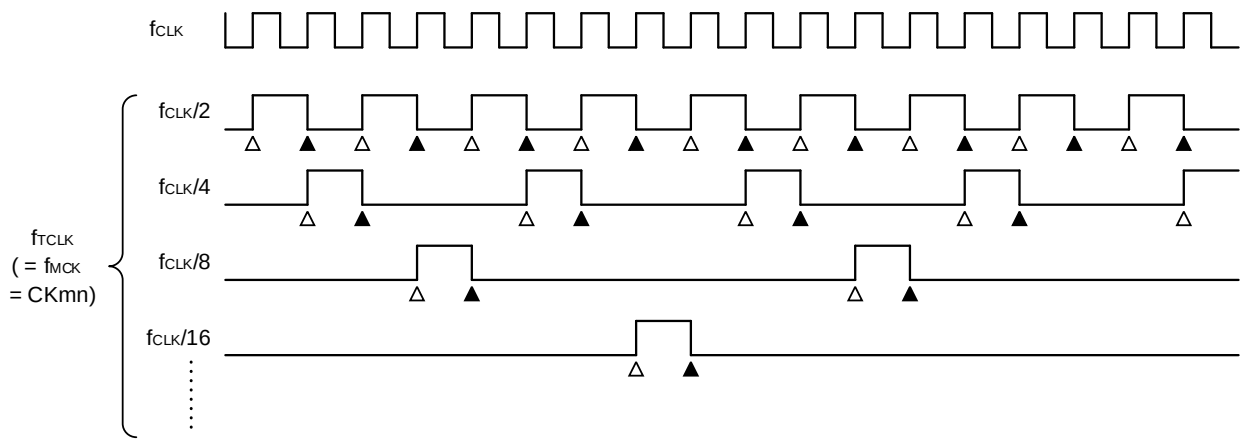
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

(1) When operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 0)

The count clock (f_{TCLK}) is set to between f_{CLK} to $f_{CLK}/2^{15}$ by setting the 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) is delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. This is described as “counting at rising edge of the count clock” as a matter of convenience.

Figure 6-30. Timing of f_{CLK} and Count Clock (f_{TCLK}) (When CCSmn = 0)



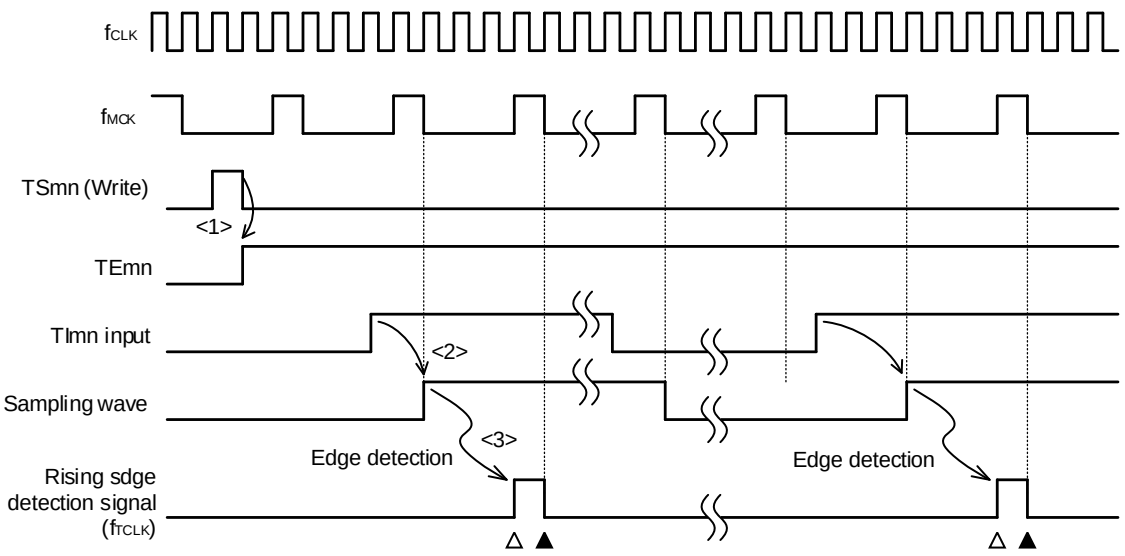
Remarks 1. △: Rising edge of the count clock
▲: Synchronization, increment/decrement of counter
2. f_{CLK}: CPU/peripheral hardware clock

(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 the valid edge of input signal via the TImn pin and synchronizes the next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 periods of f_{MCK} from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 to 4 clocks).

Counting of timer count register mn (TCRmn) is delayed by one period of f_{CLK} from the rising edge of the count clock, because of synchronization with f_{CLK}. This is described as “counting at valid edge of input signal via the TImn pin” as a matter of convenience.

Figure 6-31. 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, the delay counter, and the one-shot pulse output are the same as that shown in Figure 6-31.

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 6-7.

Table 6-7. 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. 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) Operation of interval timer mode**).
Event counter mode	Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register. The subsequent count clock performs count down operation. 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) Operation of event counter mode**).
Capture mode	No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3(3) Operation of capture mode (input pulse interval measurement)**).
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). No operation is carried out from start trigger detection until count clock generation. 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) Operation of 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). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3(5) Operation of capture & one-count mode (high-level width measurement)**).

6.5.3 Operation of Counter

Here, the counter operation in each mode is explained.

(1) Operation of 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.

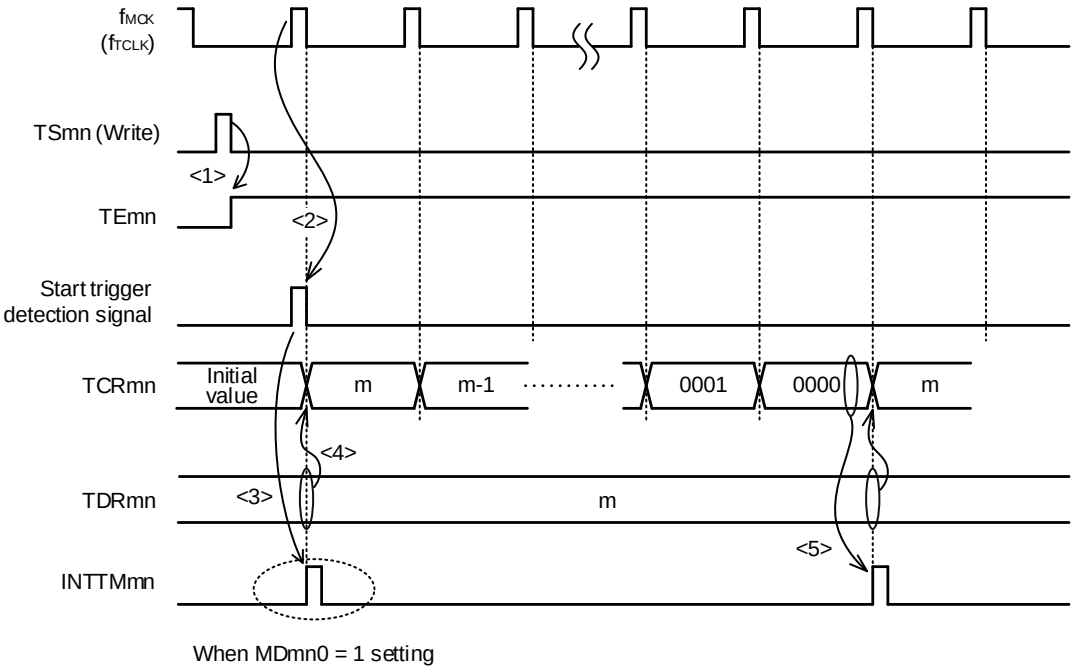
<2> A start trigger is generated at the first count clock after operation is enabled.

<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 6-32. Operation Timing (In Interval Timer Mode)



When MDmn0 = 1 setting

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) Operation of event counter mode

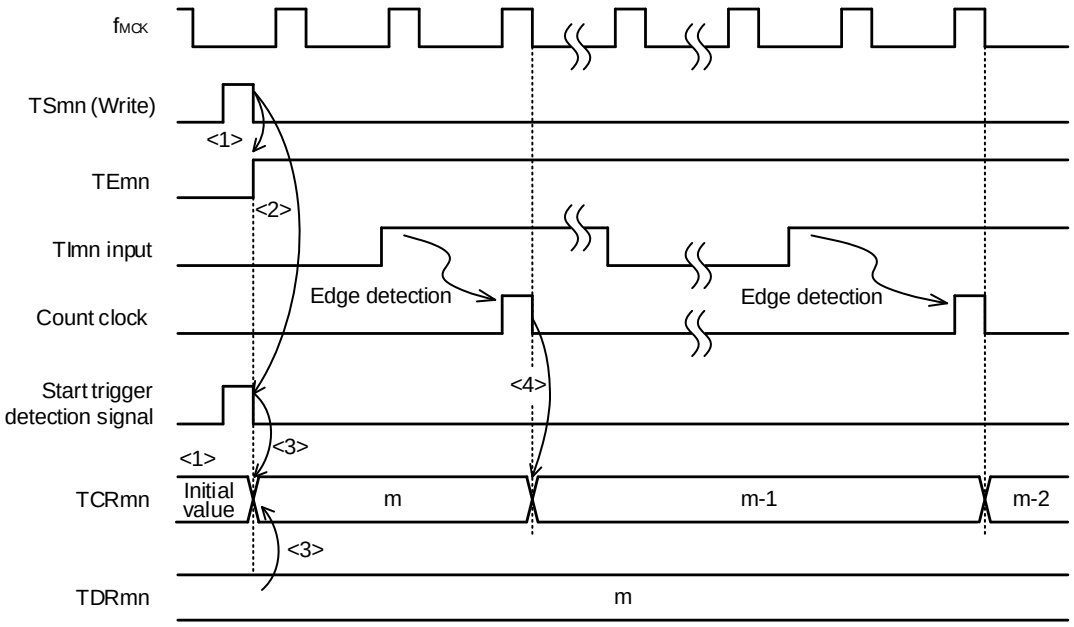
<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).

<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 6-33. Operation 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) Operation of capture mode (input pulse interval measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

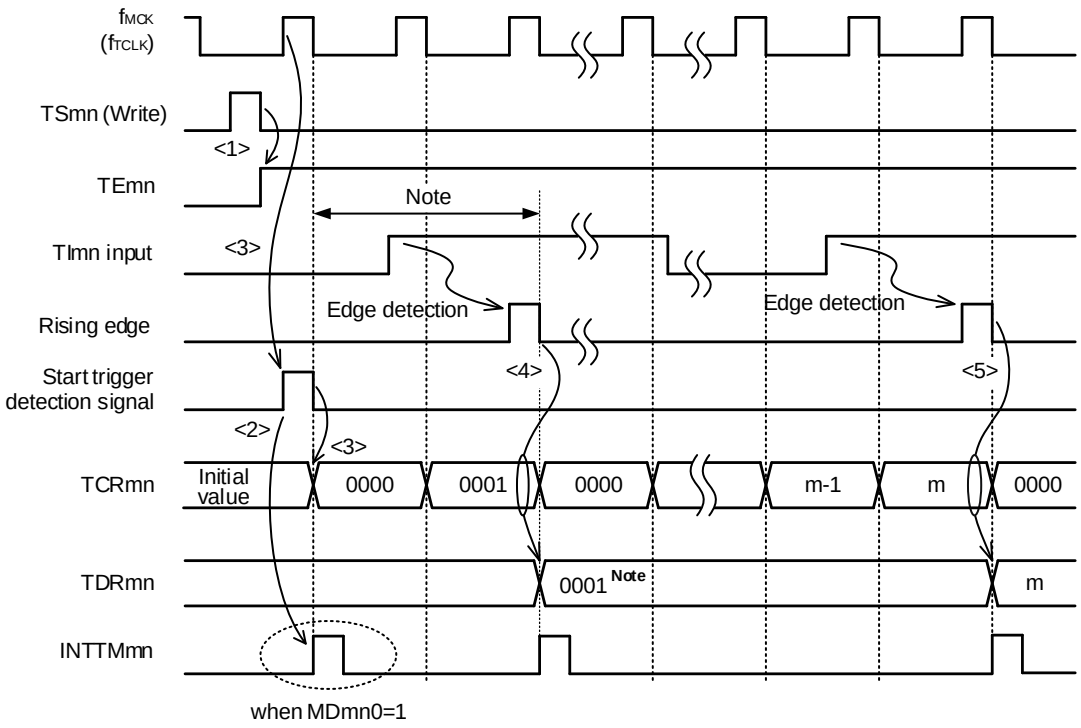
<2> Timer count register mn (TCRmn) holds the initial value until count clock generation.

<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 6-34. 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 Figure 6-34 shows the timing when 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).

(4) Operation of one-count mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

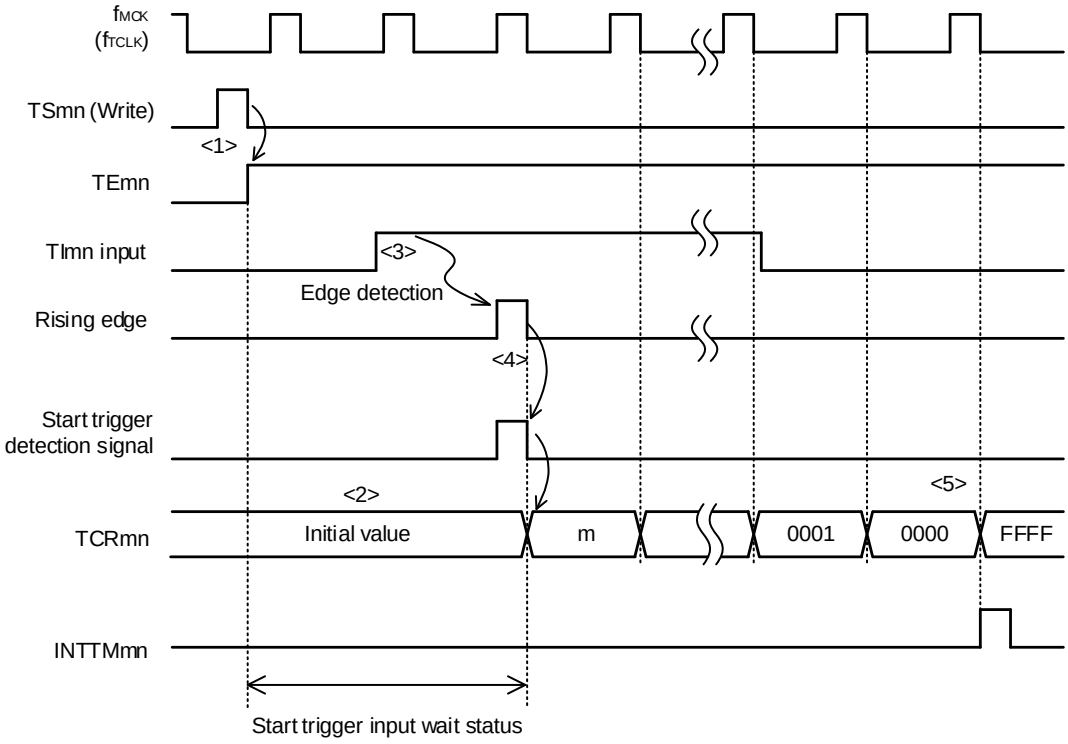
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<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 6-35. Operation Timing (In One-count Mode)



Remark The timing is shown in Figure 6-35 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. The error per one period occurs because of the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

(5) Operation of capture & one-count mode (high-level width measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit of timer channel start register m (TSm).

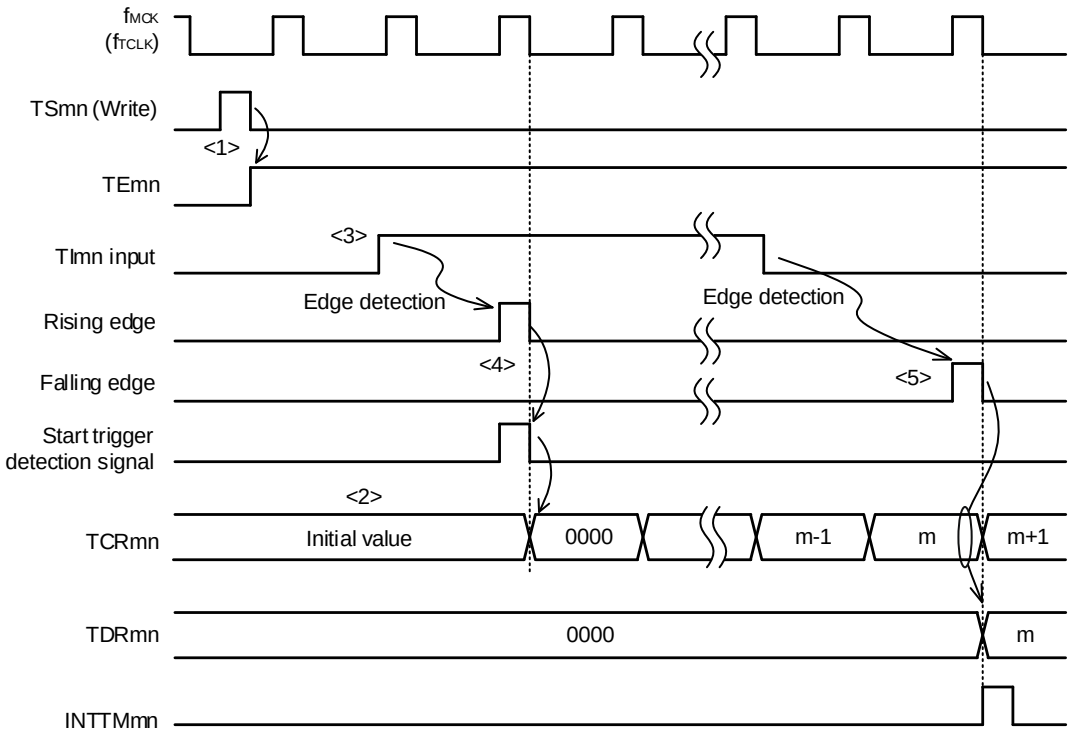
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<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 6-36. Operation Timing (in Capture & One-count Mode: High-level Width Measurement)

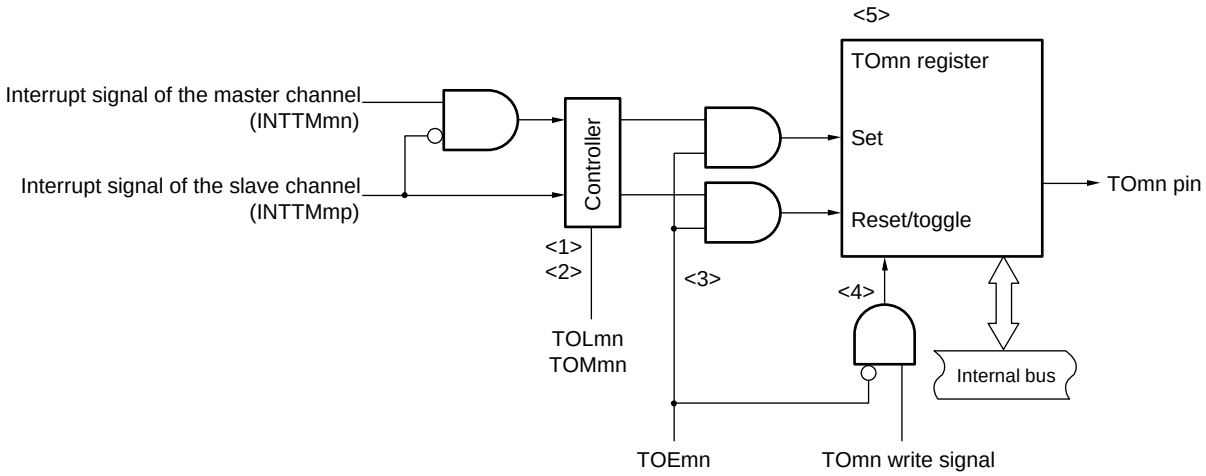


Remark The timing is shown in Figure 6-36 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs because of the asynchronous relationship between the period of the TImn input and that of the count clock (f_{MCK}).

6.6 Channel Output (TOmn pin) Control

6.6.1 TOmn Pin Output Circuit Configuration

Figure 6-37. 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 INTTMmp (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 INTTMmp (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, INTTMmp → reset)
When TOLmn = 1: Reverse operation (INTTMmn → reset, INTTMmp → set)

When INTTMmn and INTTMmp are simultaneously generated, (0% output of PWM), INTTMmp (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTMmp (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 INTTMmp (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.

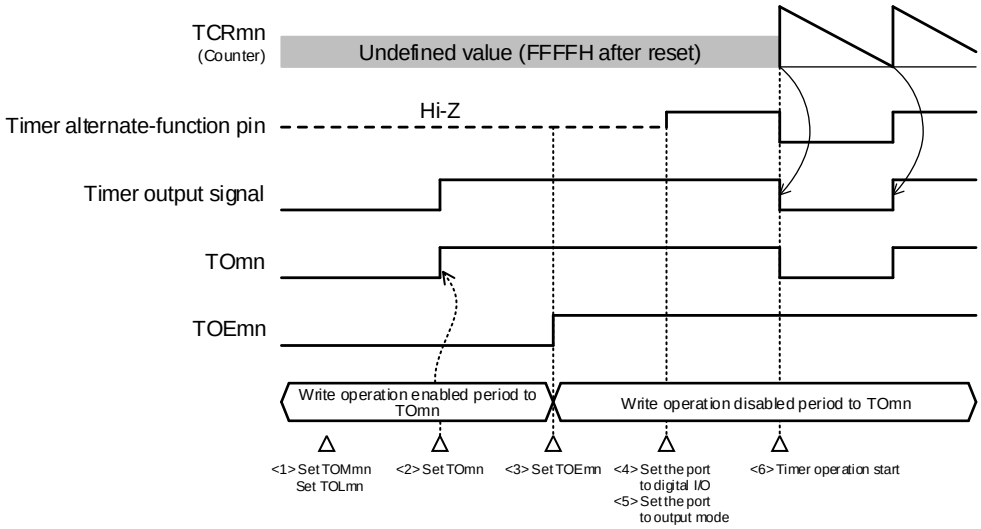
Remarks 1. m: Unit number (m = 0, 1)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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 6-38. 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 (PMCm)**).

<5> The port I/O setting is set to output (see **6.3.16 Port Mode Registers 0, 1, 3, 4, 5, 6, 7, 12 (PM0, PM1, PM3, PM4, PM5, PM6, 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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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 in **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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

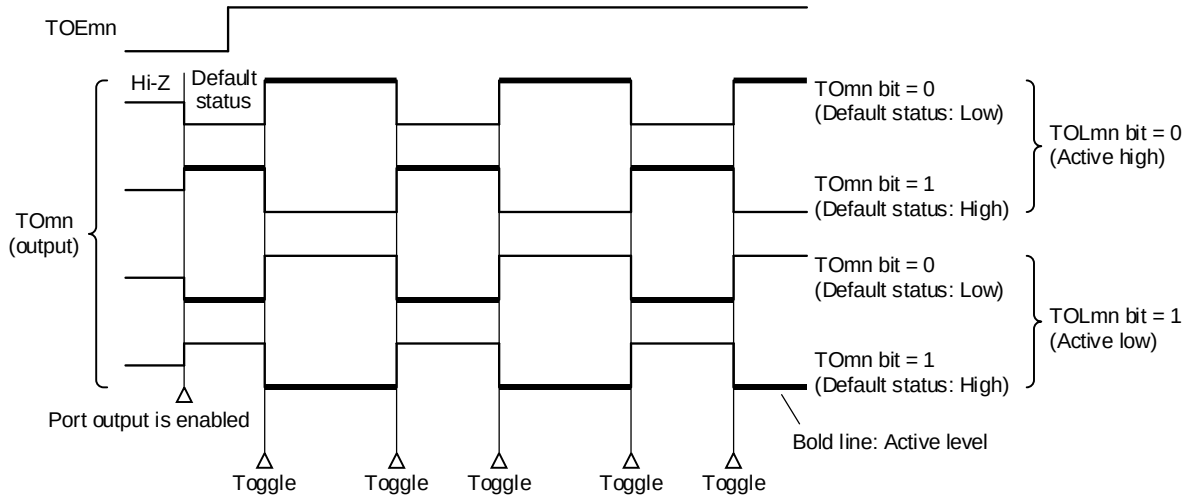
(2) Default level of TOmn pin and output level after timer operation start

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, as 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 6-39. TOmn Pin Output Status at Toggle Output (TOMmn = 0)



Remarks 1. Toggle: Reverse TOmn pin output status

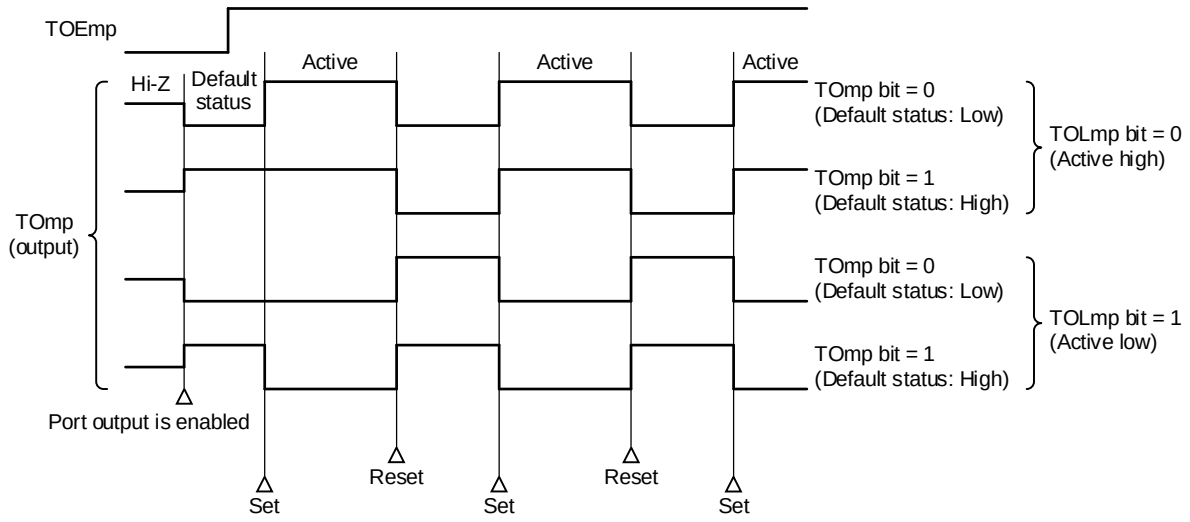
2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

3. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

(b) When operation starts with slave channel output mode (TOMmp = 1) setting (PWM output))

When slave channel output mode (TOMmp = 1), the active level is determined by timer output level register m (TOLm) setting.

Figure 6-40. TOmp Pin Output Status at PWM Output (TOMmp = 1)



Remarks 1. Set: The output signal of the TOmp pin changes from inactive level to active level.
Reset: The output signal of the TOmp pin changes from active level to inactive level.

2. m: Unit number (m = 0, 1), p: Channel number (p = 1 to 7)

3. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

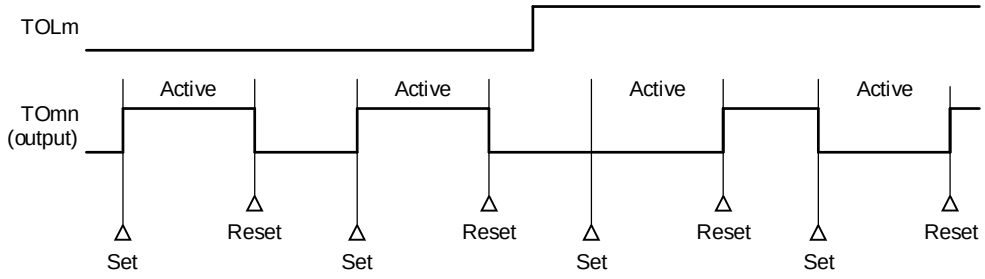
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(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 6-41. 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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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.

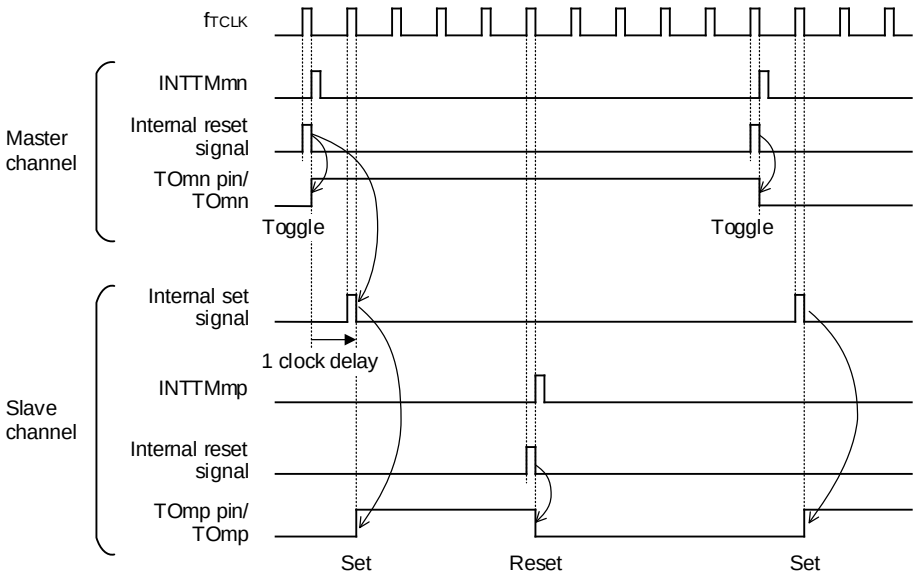
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-42 shows the set/reset operating statuses where the master/slave channels are set as follows.

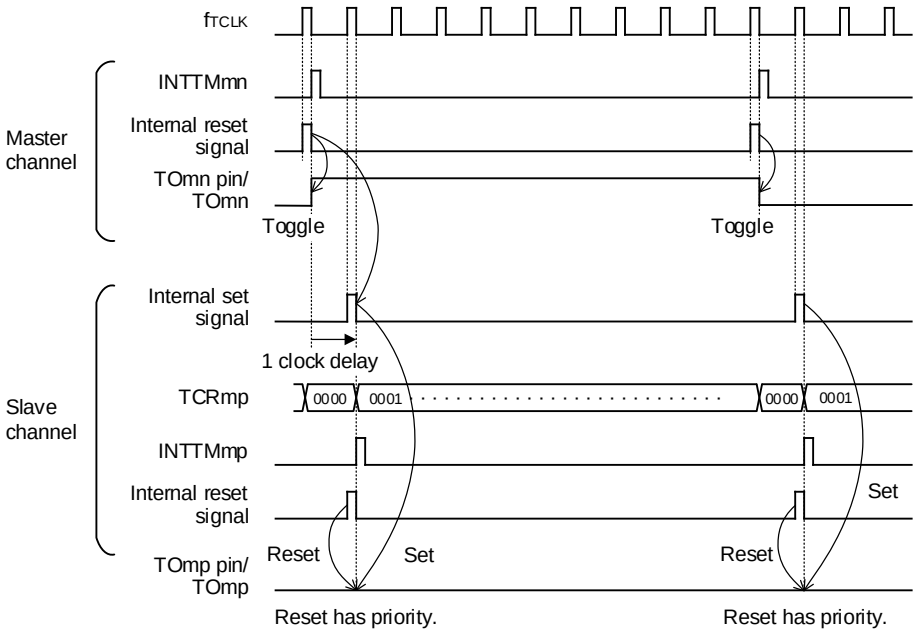
Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 6-42. 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

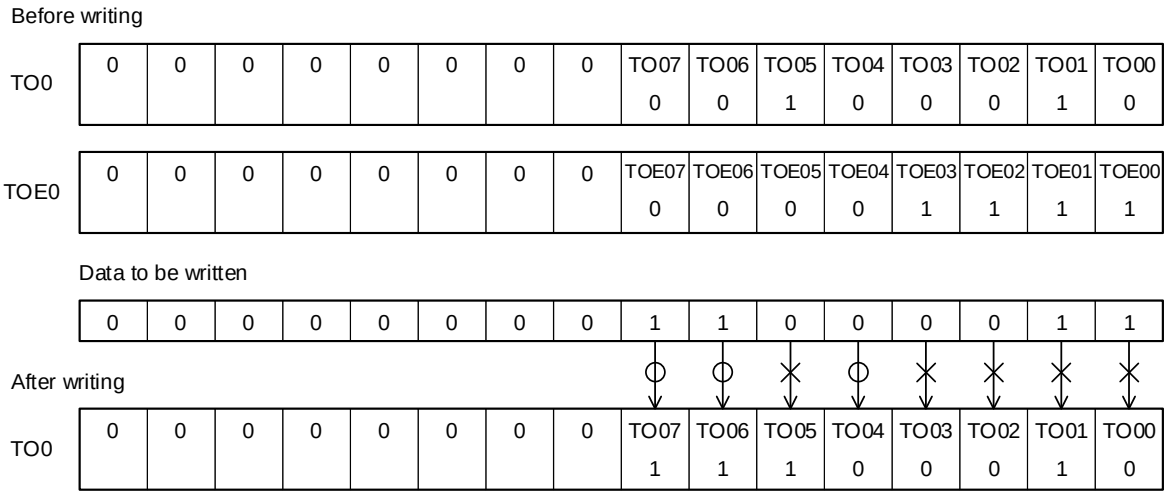
3. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

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).

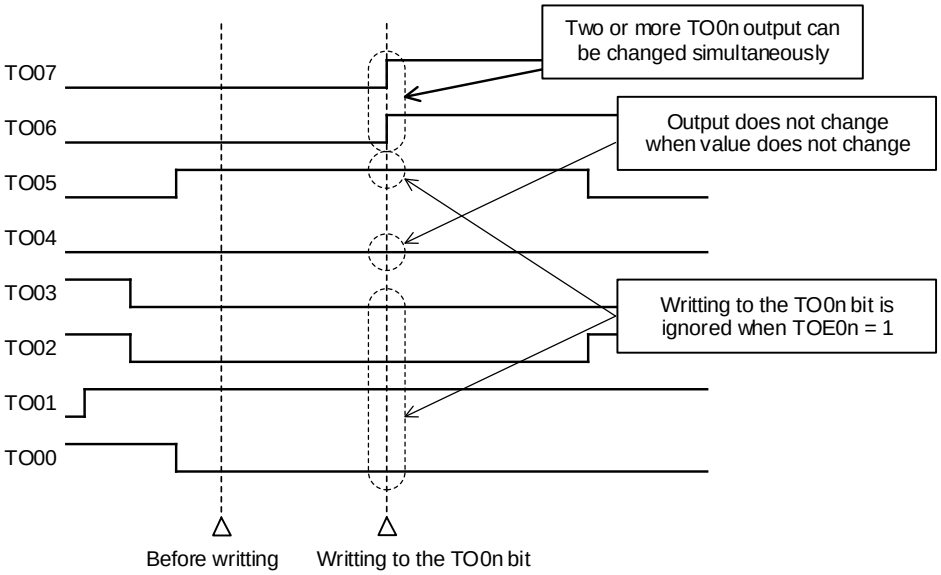
Figure 6-43. Example of TO0n Bit Collective Manipulation



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 6-44. TO0n Pin Statuses by Collective Manipulation of TO0n Bit



Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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.

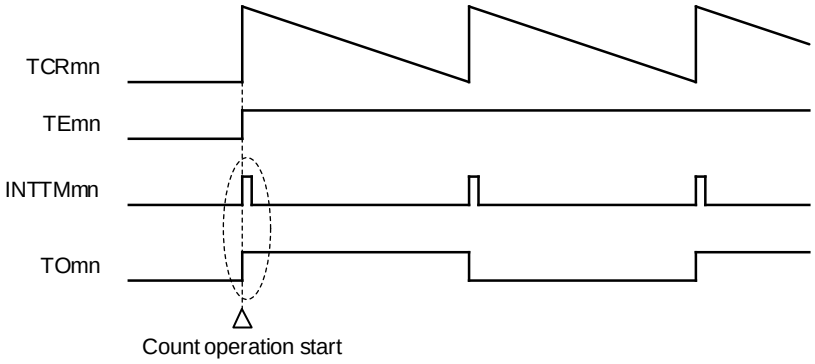
When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

In the other modes, neither timer interrupt at count operation start nor TOmn output is controlled.

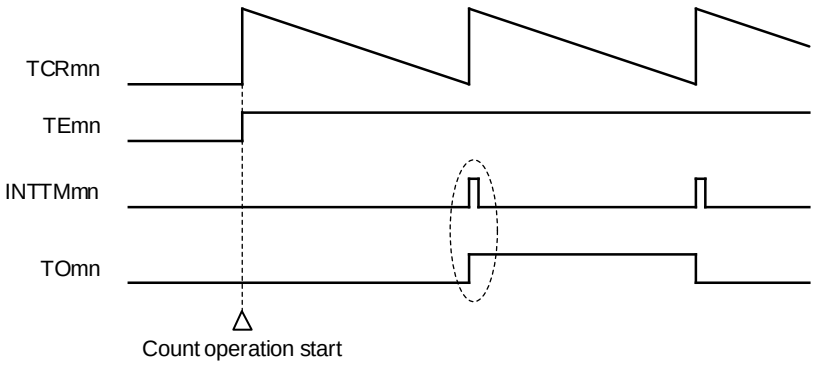
Figure 6-45 show operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 6-45. 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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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.

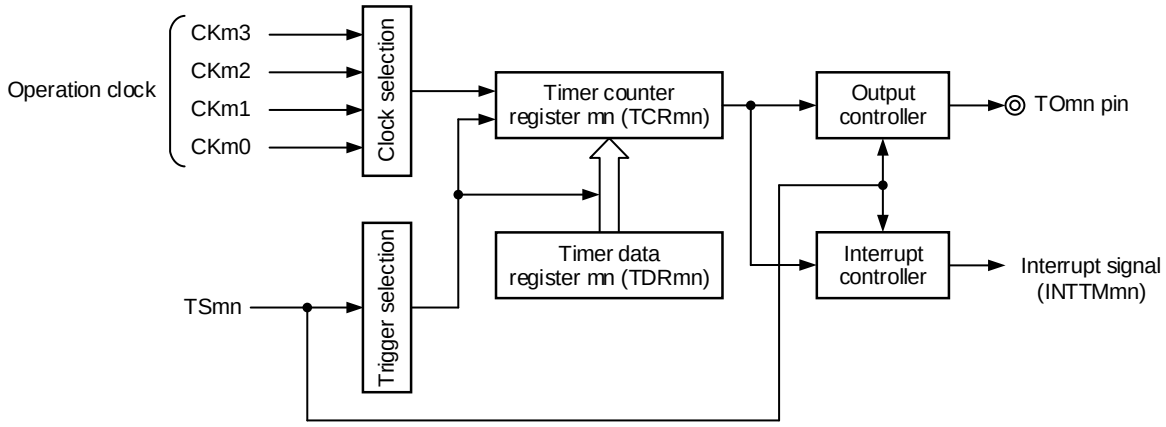
When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, the TCRmn register loads the value of the TDRmn register again. After that, the same operation is repeated.

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)

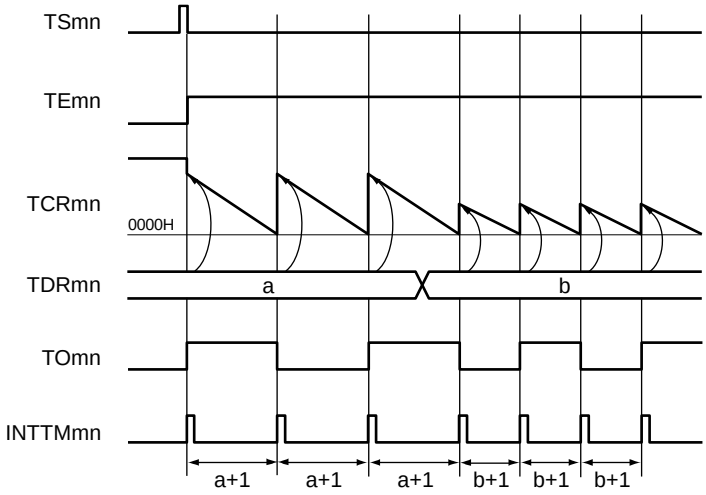
2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-46. Block Diagram of Operation as Interval Timer/Square Wave Output



Remark In case of channels 0 and 2 of unit 1 in RL78/F22 products, the clock cannot be selected CK12 and CK13 (see **Figure 6-2** and **Figure 6-12**).

Figure 6-47. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)



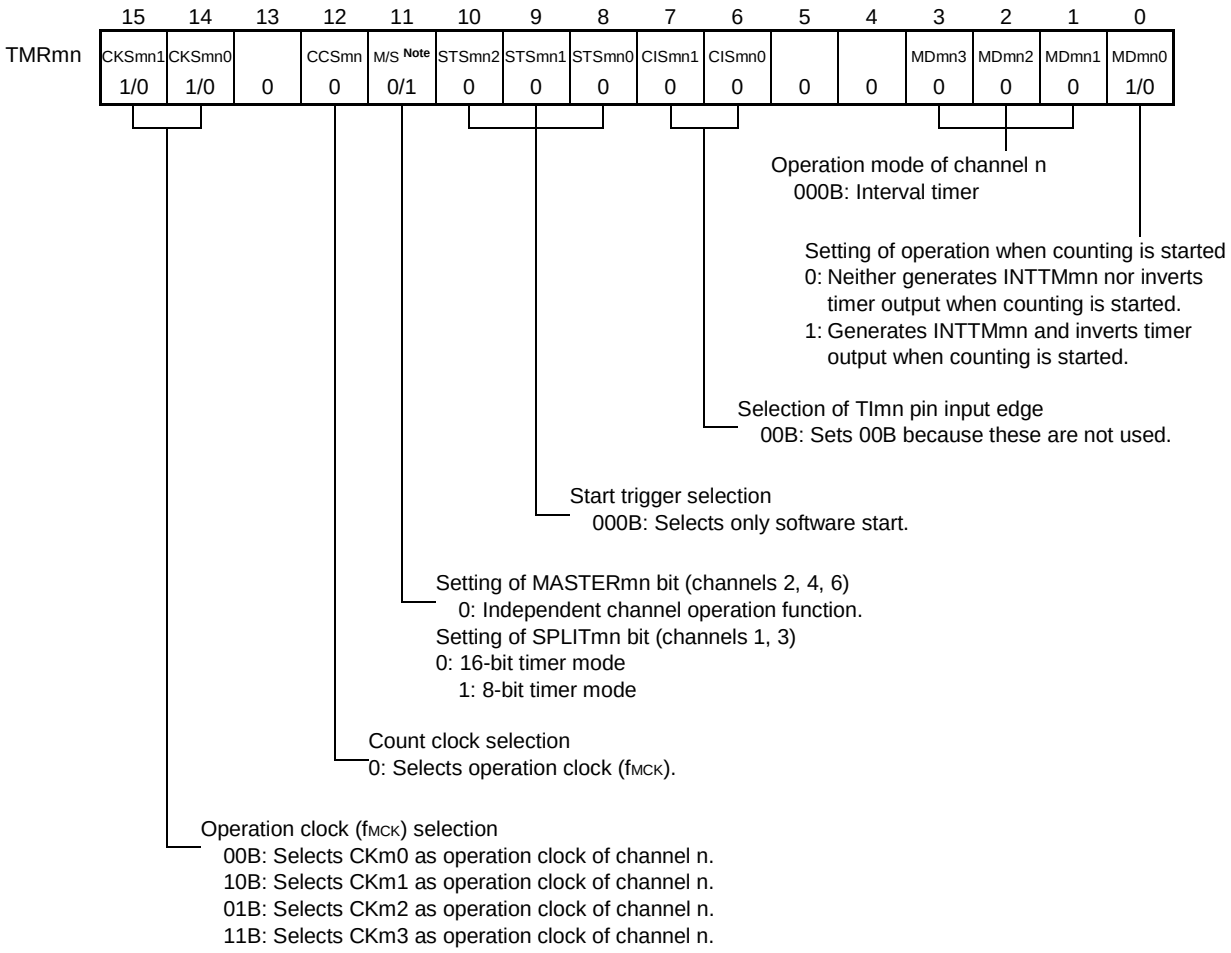
Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

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
TDRmn: Timer data register mn
TOmn: TOmn pin output signal

3. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-48. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output

(a) Timer mode register mn (TMRmn)



(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.

(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.

(**Note** and **Remarks** are on the next page.)

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-49. Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (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. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets interval (period) value to timer data register mn (TDRmn).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOmn output Clears the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output. →	The TOmn pin goes into Hi-Z output state. 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.). Sets the TSmn (TSHm1, TSHm3) bit to 1. → The TSmn (TSHm1, TSHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3) = 1, and count operation starts. 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. Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. 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. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn (TTHm1, TTHm3) bit is set to 1. → The TTmn (TTHm1, TTHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3), and count operation stops. The TCRmn register holds count value and stops. 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 6-49. Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOmn pin output level: 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: Setting not required.	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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.

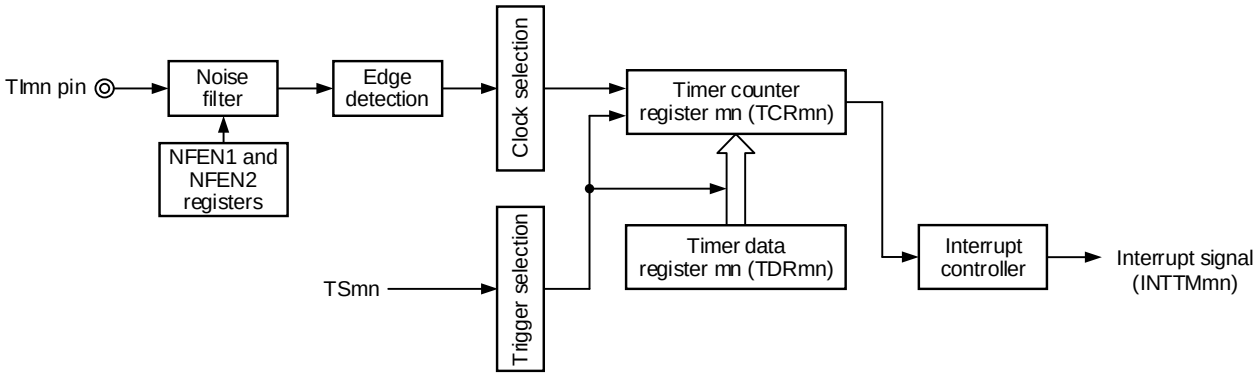
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.

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.

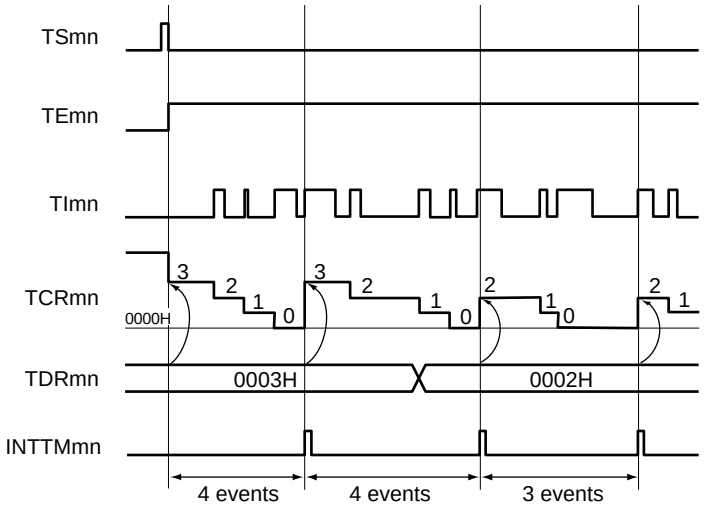
Figure 6-50. Block Diagram of Operation as External Event Counter



Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-51. Example of Basic Timing of Operation as External Event Counter



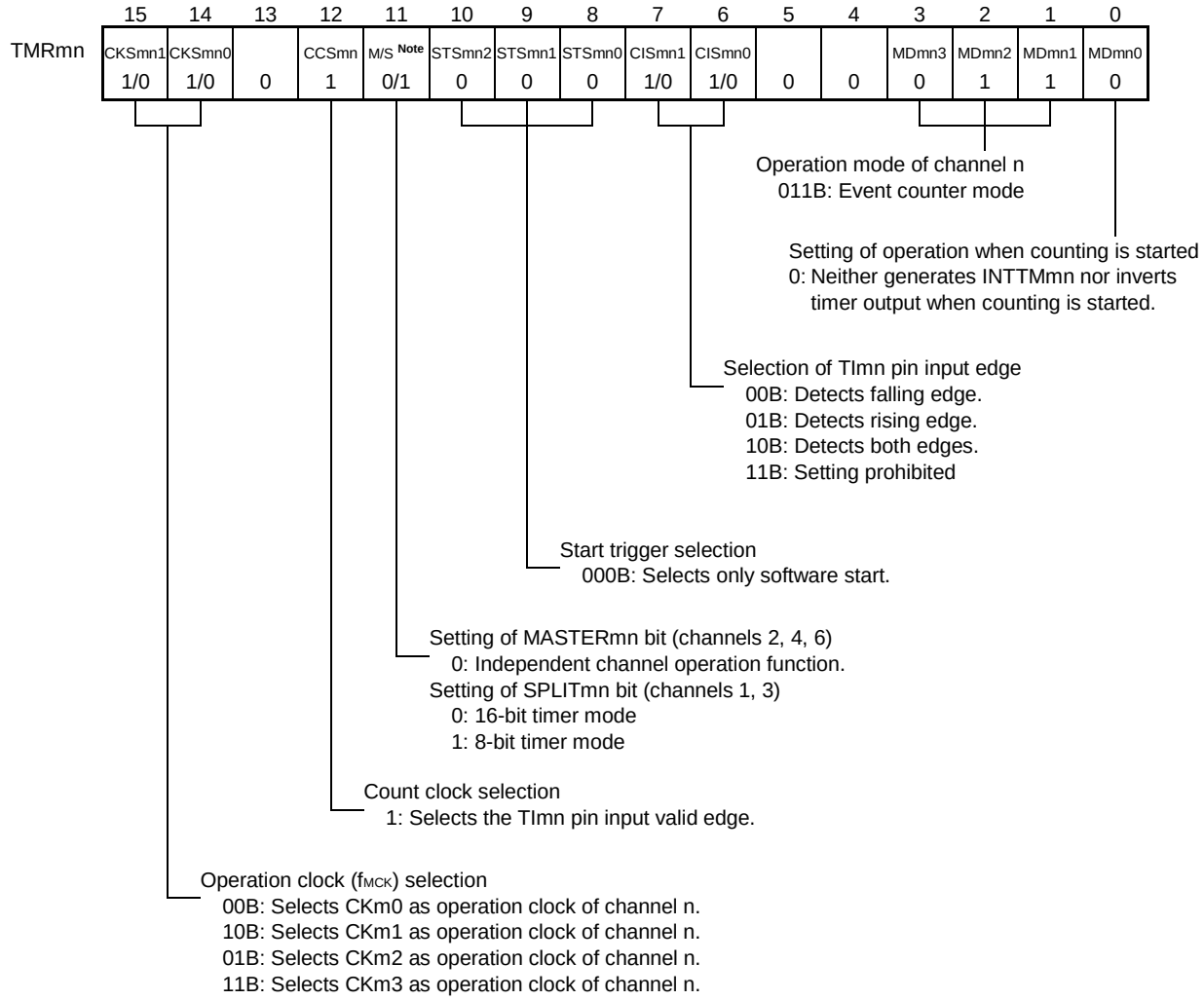
Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

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
TDRmn: Timer data register mn

3. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-52. Example of Set Contents of Registers in External Event Counter Mode

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn	0: Outputs 0 from TOmn.
	0	

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn	0: Stops the TOmn output operation by counting operation.
	0	

(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	

(**Note** and **Remarks** are on the next page.)

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-53. Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets number of counts to timer data register mn (TDRmn). Clears the TOEmn bit of timer output enable register m (TOEm) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. 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. The TCRmn register can always be read. The TSRmn register is not used 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. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status 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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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, 3, and 9 (PIOR0, PIOR1, PIOR2, PIOR3, and PIOR9).

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 TDRmn + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≃ Input clock frequency / (Set value of TDRmn + 1)

Timer count register m (TCRm) operates as a down counter in the interval timer mode.

After the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TCRmn register loads the value of timer data register mn (TDRmn) when the TImn valid edge is detected.

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 timer mode register mn (TMRmn) is 1, INTTMmn is output and TOmn is toggled.

After that, the TCRmn register counts down at the valid edge of the TImn pin. When TCRmn = 0000H, it toggles TOmn. At the same time, the TCRmn register loads the value of the TDRmn register again, and continues counting.

If detection of both the edges of the TImn pin is selected, the duty factor error of the input clock affects the divided clock period of the TOmn output.

The period of the TOmn output clock includes a sampling error of one period of the operation clock.

- Clock period of TOmn output = Ideal TOmn output clock period ± Operation clock period (error)

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 6-54. Block Diagram of Operation as Frequency Divider

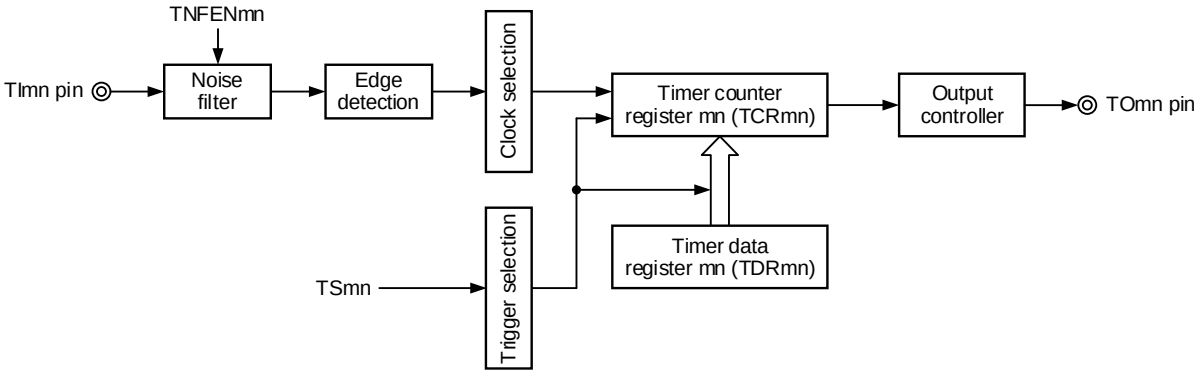
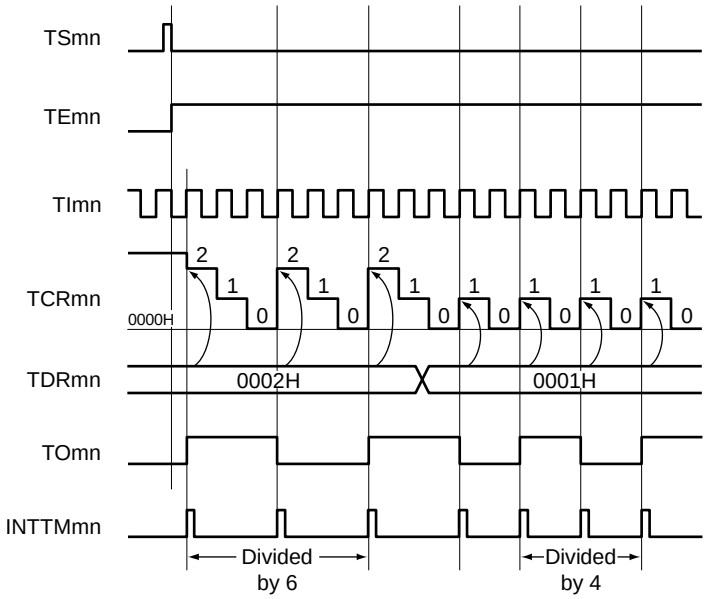


Figure 6-55. Example of Basic Timing of Operation as Frequency Divider (MDmn0 = 1)



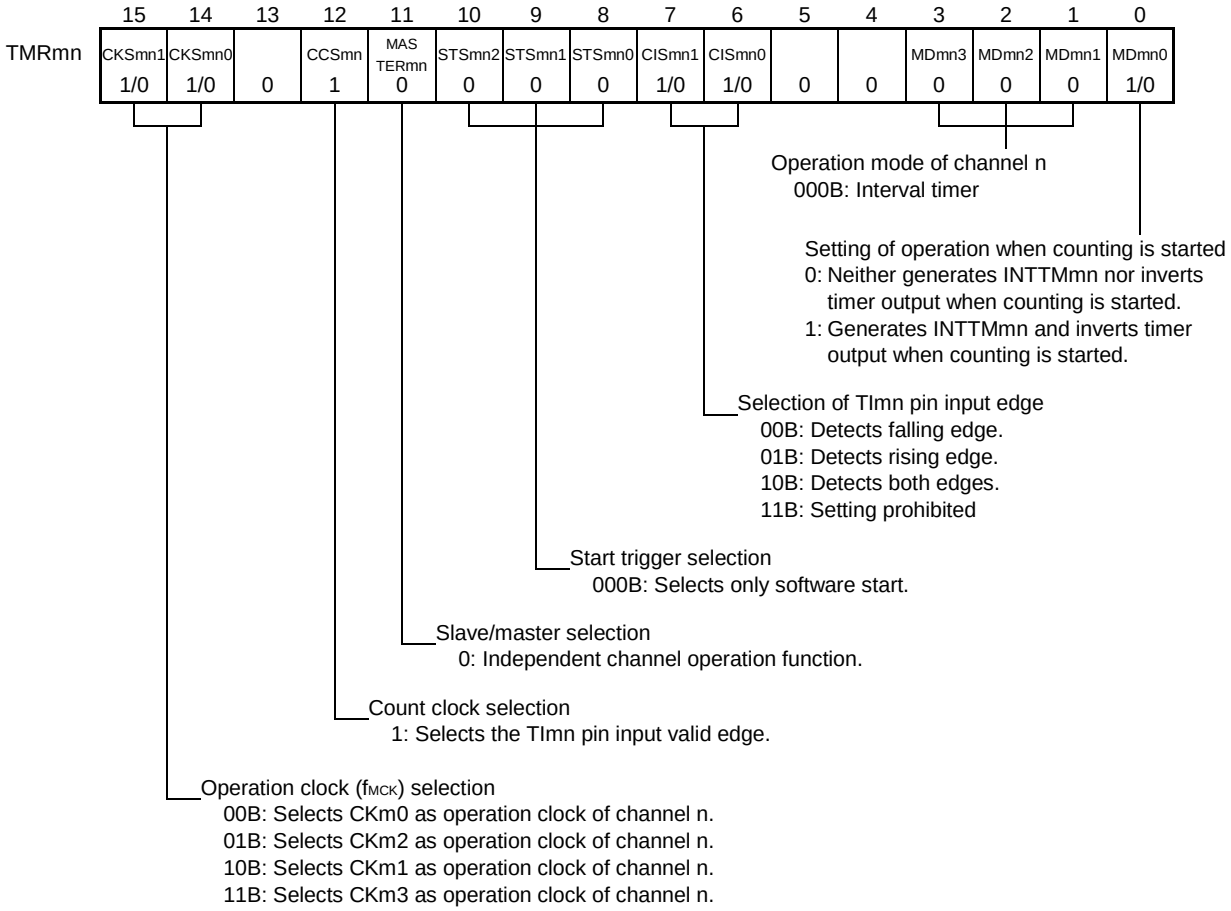
Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn
TDRmn: Timer data register mn
TOmn: TOmn pin output signal

3. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-56. Example of Set Contents of Registers During Operation as Frequency Divider

(a) Timer mode register mn (TMRmn)



(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.

(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: Channel number (n = 0 to 7)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-57. Operation Procedure When Frequency Divider Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel and selects the detection edge). Sets interval (period) value to timer data register mn (TDRmn).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output. →	The TOmn pin goes into Hi-Z output state. The TOmn default setting level is output when the port mode register is in 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 when operation is resumed). → Sets the TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. 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 value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOm and TOEm registers can be changed. Set values of the TMRmn register, TOMmn, and TOLmn bits cannot 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. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. 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 set level.
TAU stop	To hold the TOmn pin output level Clears the TOmn bit to 0 after the value to be held is set to the port register. → When holding the TOmn pin output level is not necessary Setting not required.	The TOmn pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode).

Operation is resumed.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

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.

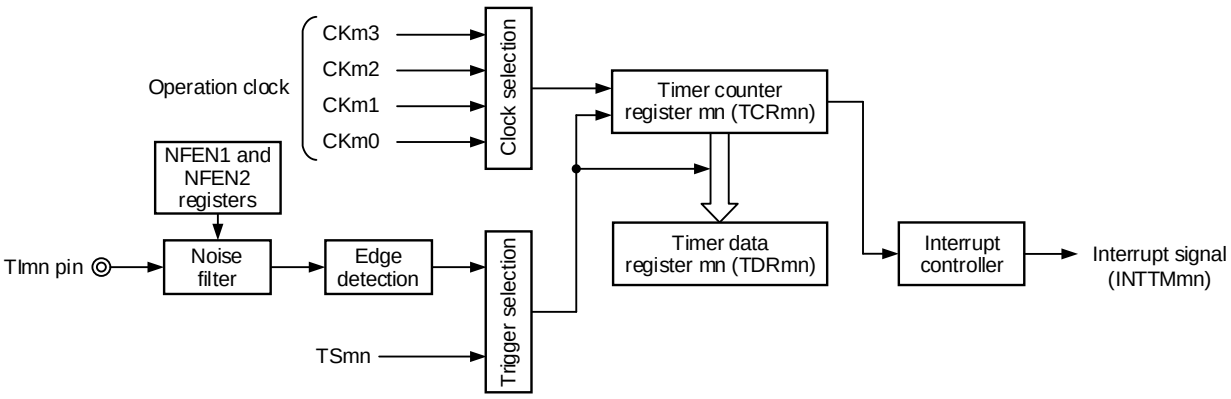
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

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 6-58. Block Diagram of Operation as Input Pulse Interval Measurement

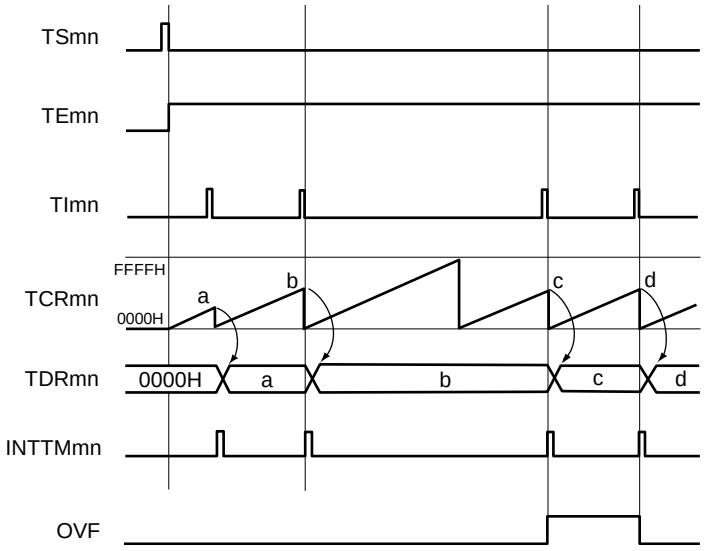


Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

3. In case of channels 0 and 2 of unit 1 in RL78/F22 products, the clock cannot be selected CK12 and CK13 (see **Figure 6-2** and **Figure 6-12**).

Figure 6-59. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)



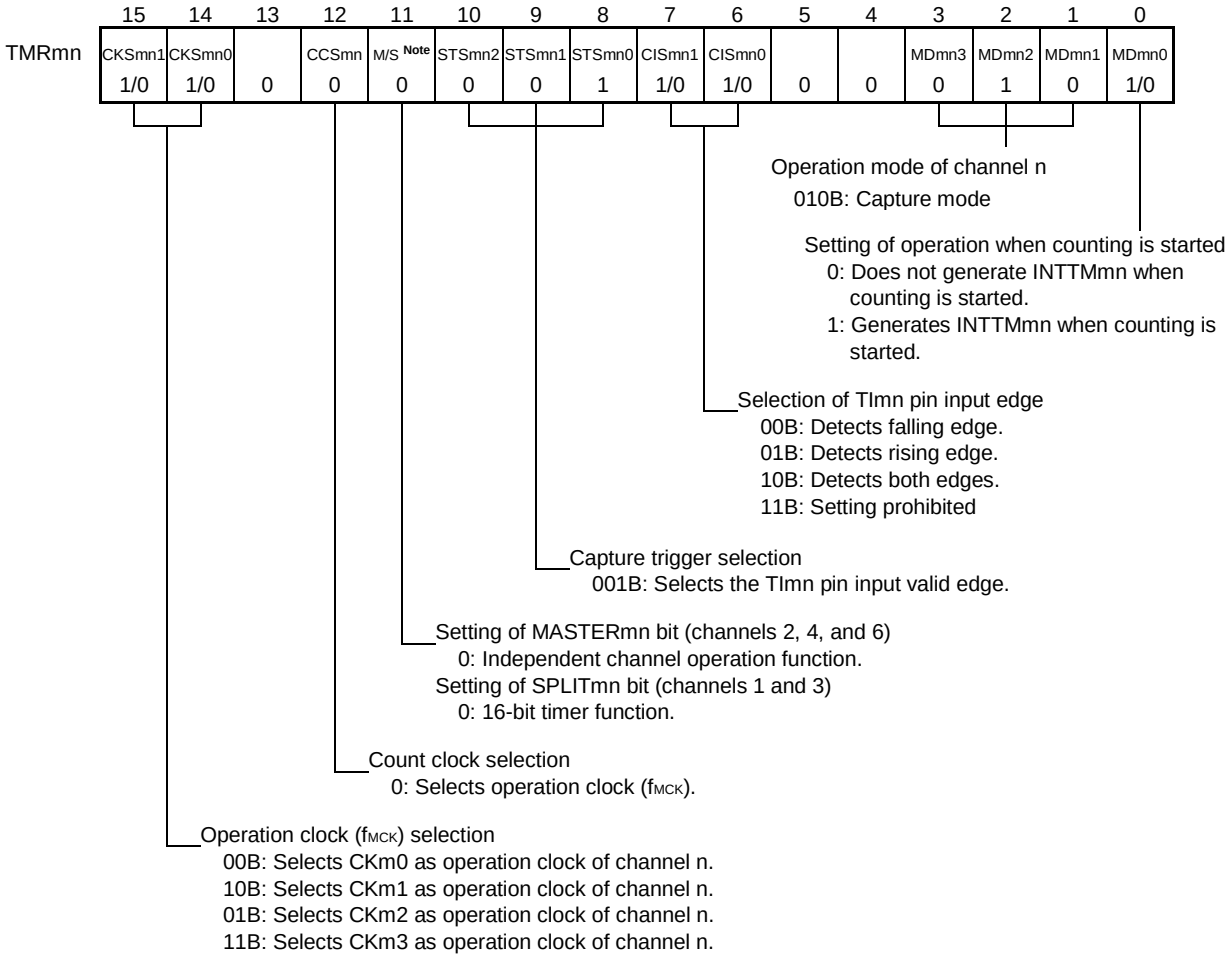
Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

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
TDRmn: Timer data register mn
OVF: Bit 0 of timer status register mn (TSRmn)

3. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-60. 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	0: Outputs 0 from TOmn.
	0	

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn	0: Stops TOmn output operation by counting operation.
	0	

(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	

(**Note** and **Remarks** are listed on the next page.)

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-61. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Timer count register mn (TCRmn) is cleared to 0000H at the count clock input. 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. The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. 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. 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. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. 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 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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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.

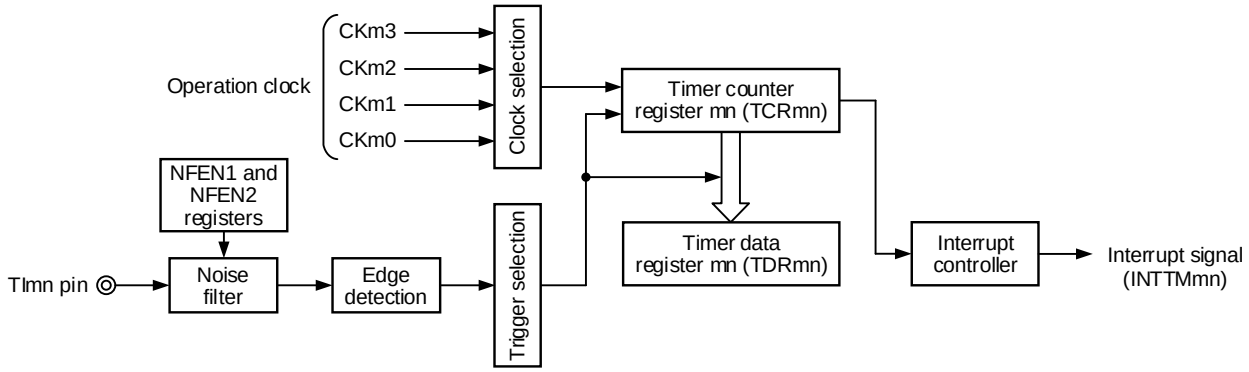
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.

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.

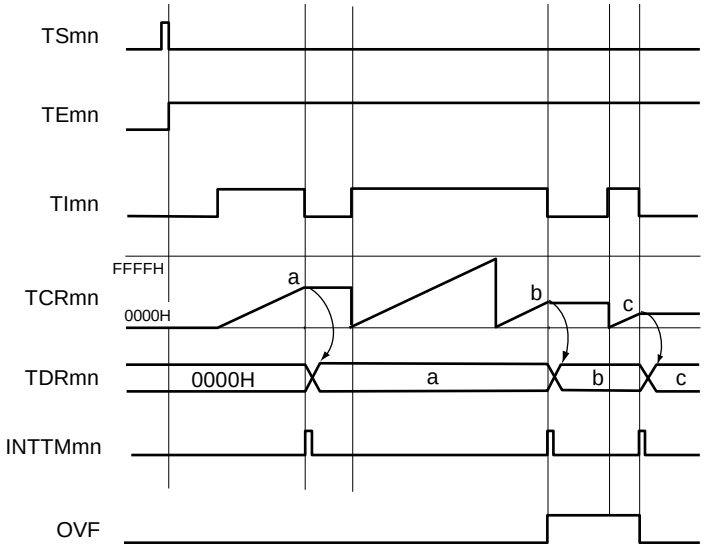
Figure 6-62. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

2. In case of channels 0 and 2 of unit 1 in RL78/F22 products, the clock cannot be selected CK12 and CK13 (see **Figure 6-2** and **Figure 6-12**).

Figure 6-63. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



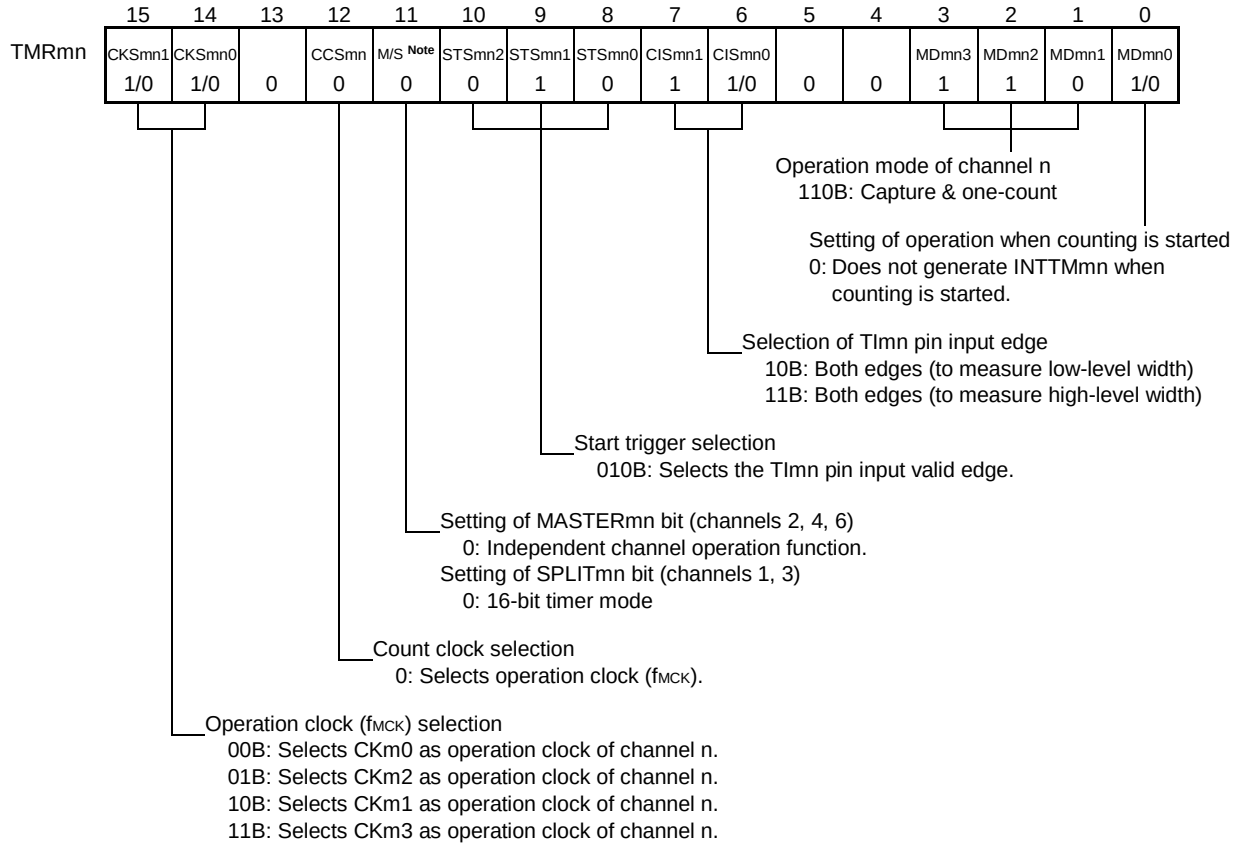
Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

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
TDRmn: Timer data register mn
OVF: Bit 0 of timer status register mn (TSRmn)

3. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-64. 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)

	Bit n	
TOm	TOmn	0: Outputs 0 from TOmn.
	0	

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn	0: Stops the TOmn output operation by counting operation.
	0	

(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	

(**Note** and **Remarks** are listed on the next page.)

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-65. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → 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. The TCRmn register can always be read. The TSRmn register can always be read. 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. 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. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. 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 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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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)

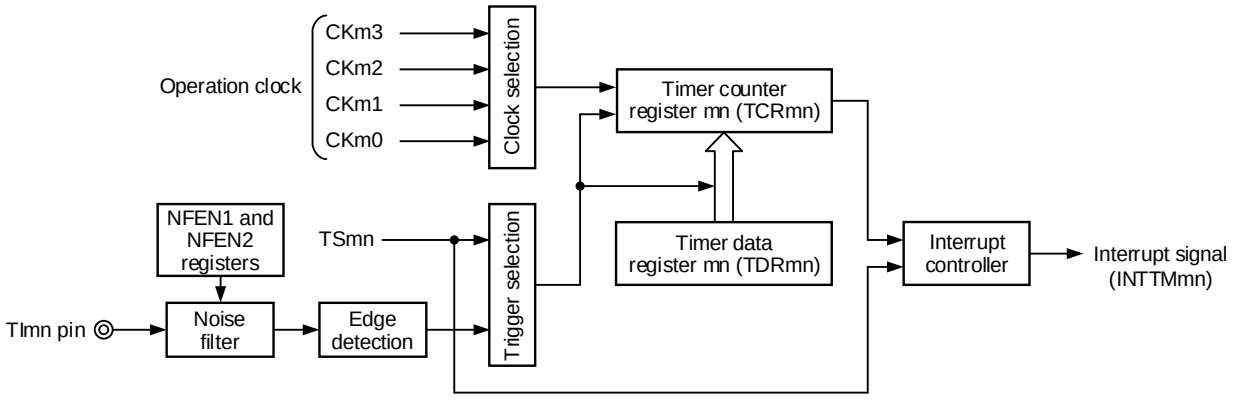
Timer count register mn (TCRmn) operates as a down counter in the one-count mode.

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 6-66. Block Diagram of Operation as Delay Counter

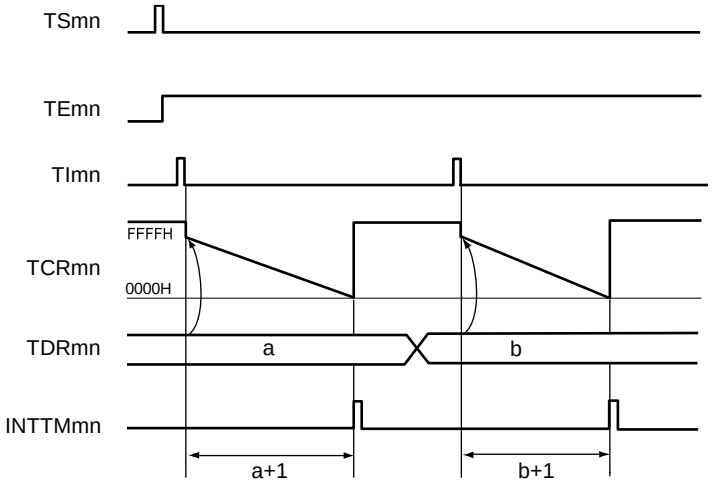


Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

3. In case of channels 0 and 2 of unit 1 in RL78/F22 products, the clock cannot be selected CK12 and CK13 (see **Figure 6-2** and **Figure 6-12**).

Figure 6-67. Example of Basic Timing of Operation as Delay Counter



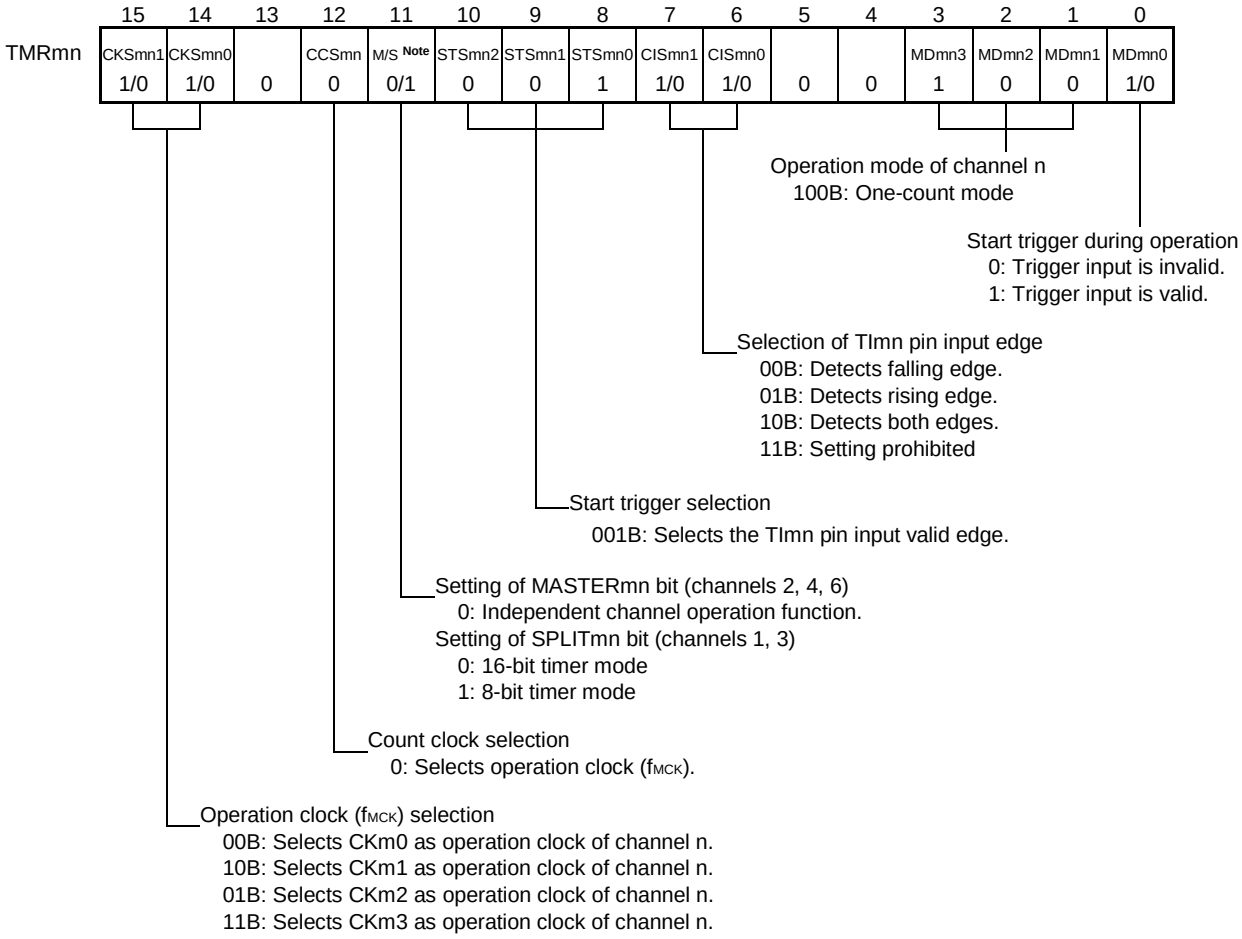
Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-68. Example of Set Contents of Registers to Delay Counter

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn	0: Outputs 0 from TOmn.
	0	

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn	0: Stops the TOmn output operation by counting operation.
	0	

(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	

(**Note** and **Remarks** are listed on the next page.)

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-69. Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (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. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). INTTMmn output delay is set to timer data register mn (TDRmn). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → 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. The TCRmn register can always be read. 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. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status 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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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.

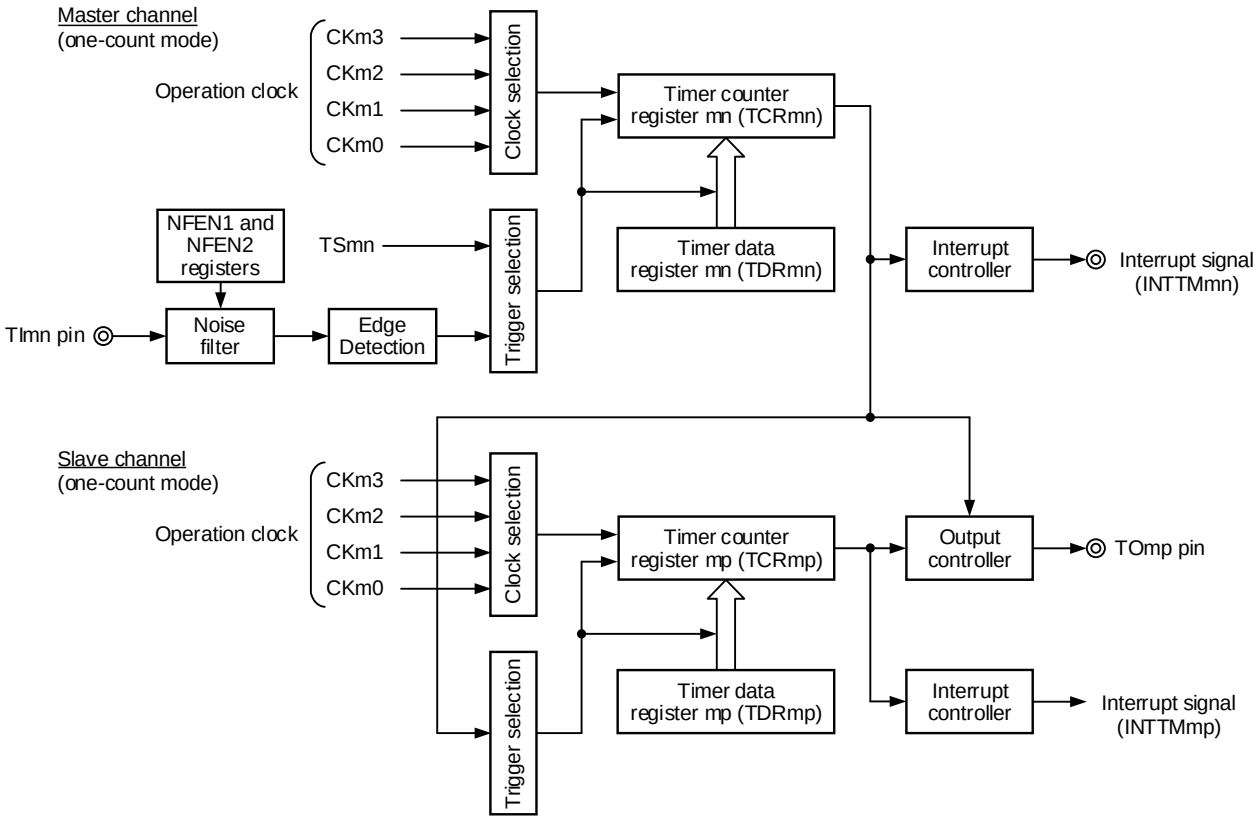
Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-70. Block Diagram of Operation as One-Shot Pulse Output Function

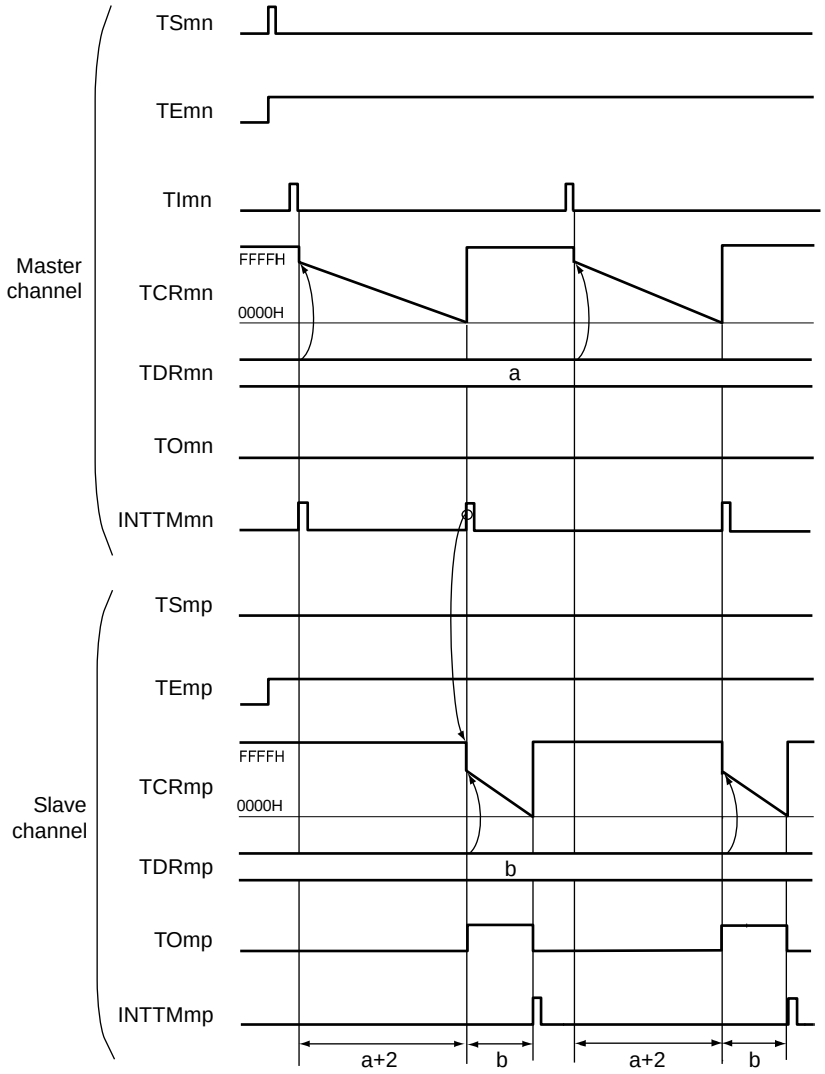


Remarks 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

3. In case of channels 0 and 2 of unit 1 in RL78/F22 products, the clock cannot be selected CK12 and CK13 (see **Figure 6-2** and **Figure 6-12**).

Figure 6-71. Example of Basic Timing of Operation as One-Shot Pulse Output Function



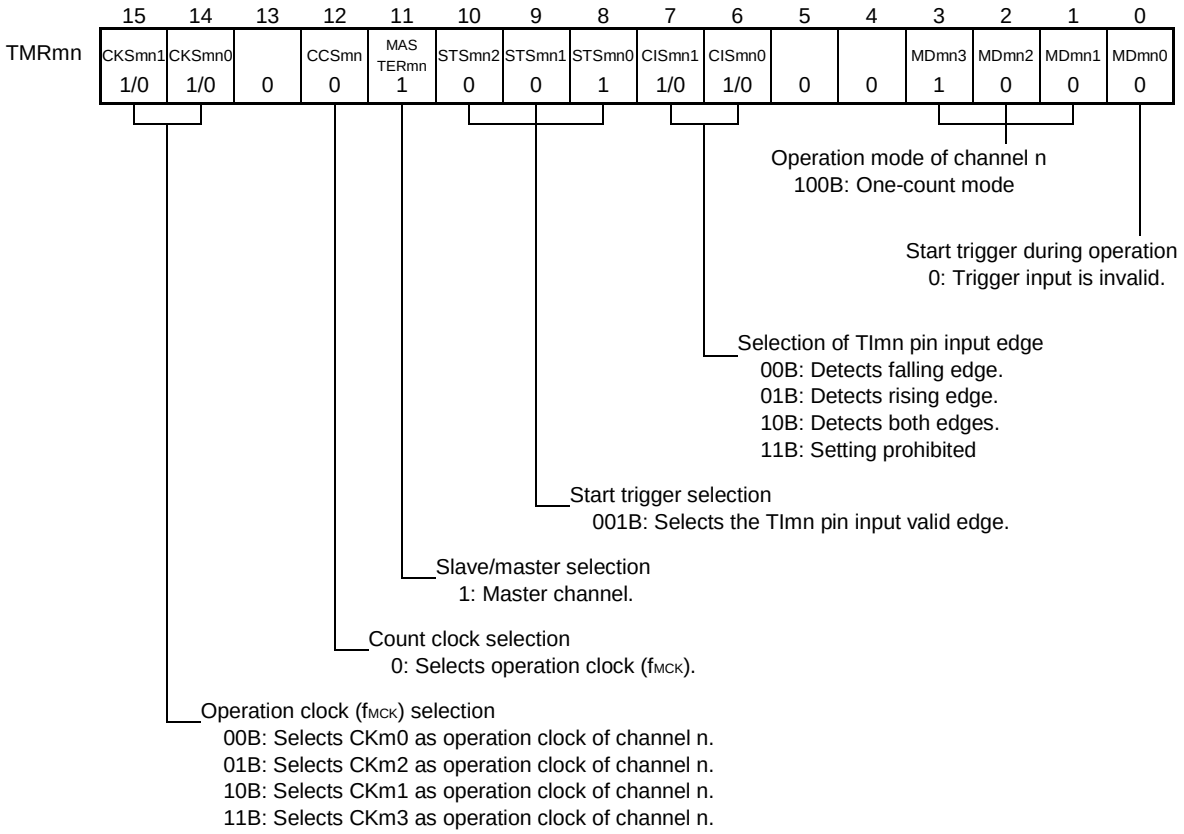
Remarks 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

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
TDRmn, TDRmp: Timer data registers mn, mp
TOmn, TOmp: TOmn and TOmp pins output signal

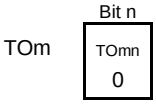
3. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-72. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register mn (TMRmn)

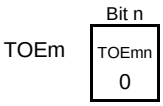


(b) Timer output register m (TOm)



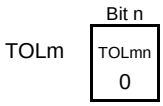
0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)



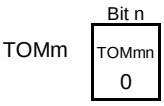
0: Stops the TOmn output operation by counting operation.

(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)



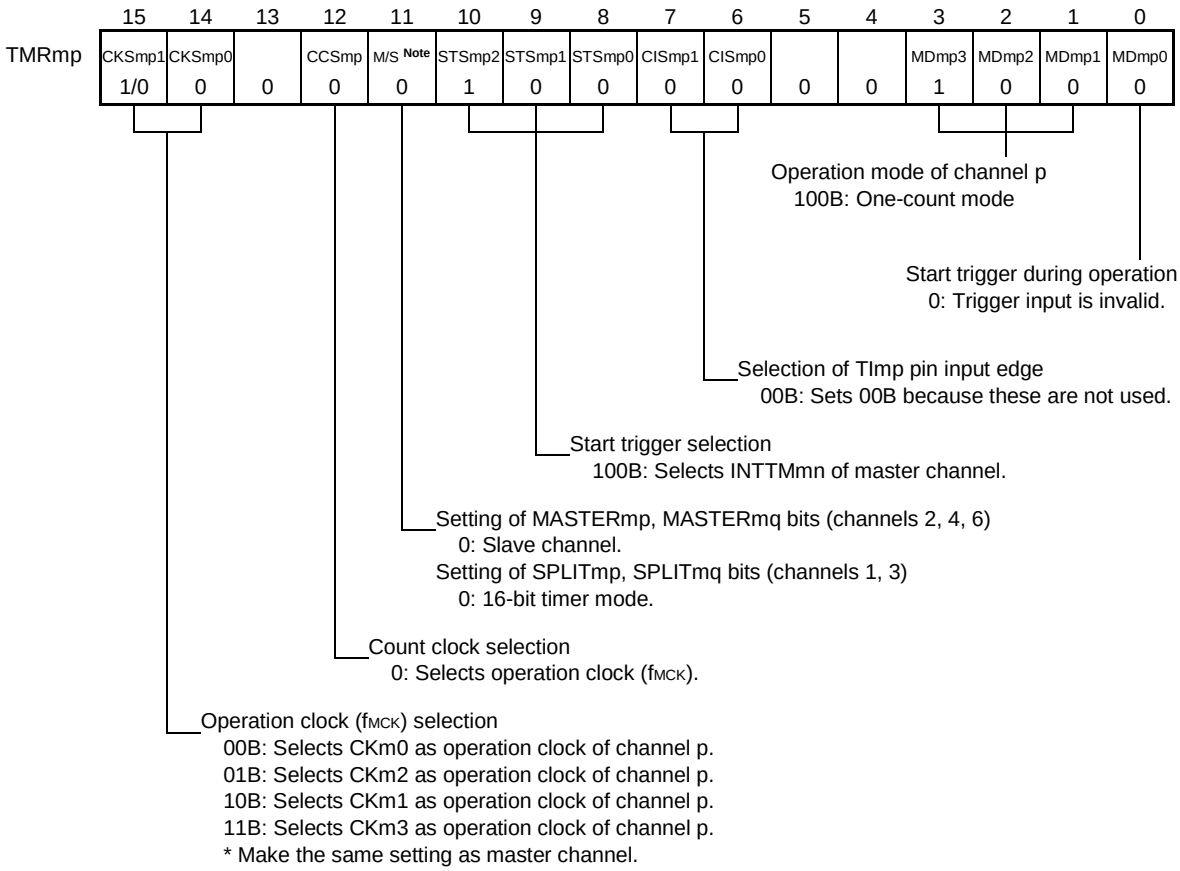
0: Sets master channel output mode.

Remarks 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-73. 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 1/0	0: Outputs 0 from TOmp. 1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p	
TOEm	TOEmp 1/0	0: Stops the TOmp output operation by counting operation. 1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p	
TOLm	TOLmp 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p	
TOMm	TOMmp 1	1: Sets the slave channel output mode.

(**Note** and **Remarks** are listed on the next page.)

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-74. Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (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. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 0 (off) or 1 (on). Sets timer mode register mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). 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. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. ——▶	The TOmp pin goes into Hi-Z output state. 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 6-74. Operation Procedure of One-Shot Pulse Output Function (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	The TEmn and TEmp bits are set to 1 and the master channel enters the TImn input edge detection wait status. Counter stops operating.
	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. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. 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. 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. 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. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	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 Clears the TOmp bit to 0 after the value to be held is set to the port register. → When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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.

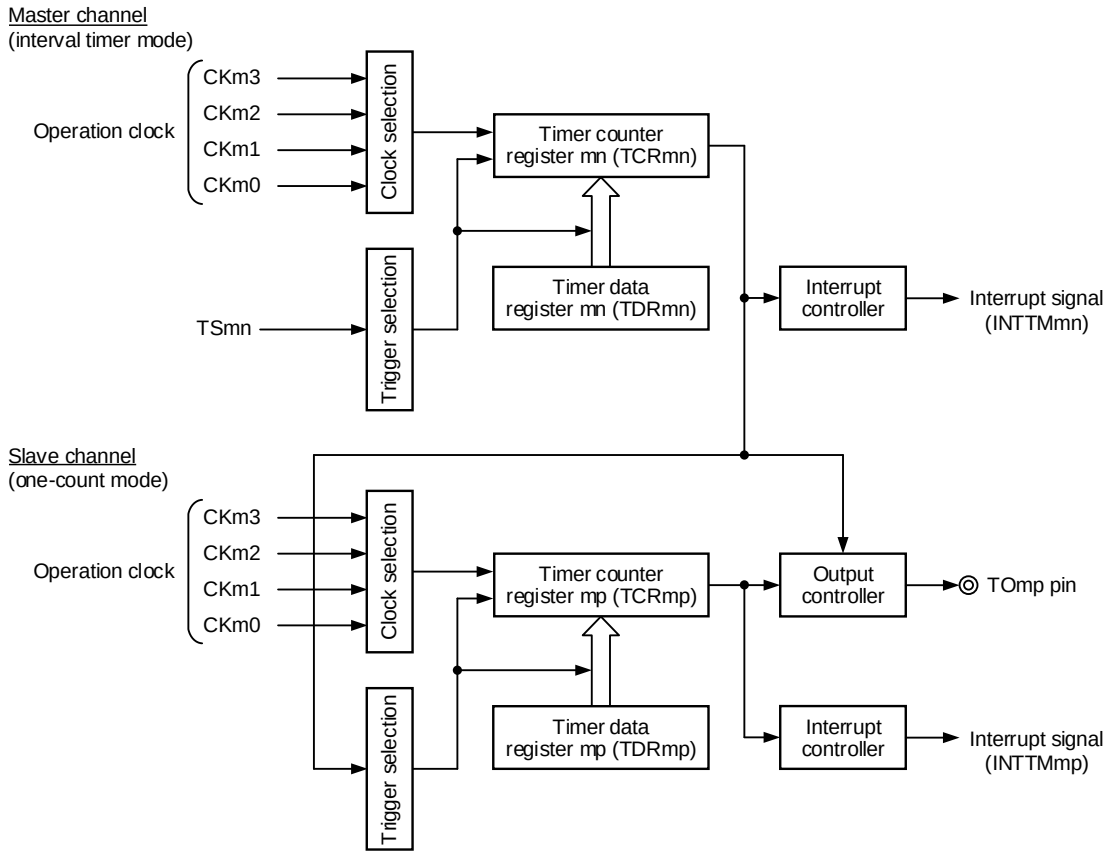
PWM output (TOmp) goes to the active level one clock after the master channel generates INTTMmn and goes to the inactive level when the TCRmp register of the slave channel becomes 0000H.

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-75. Block Diagram of Operation as PWM Function

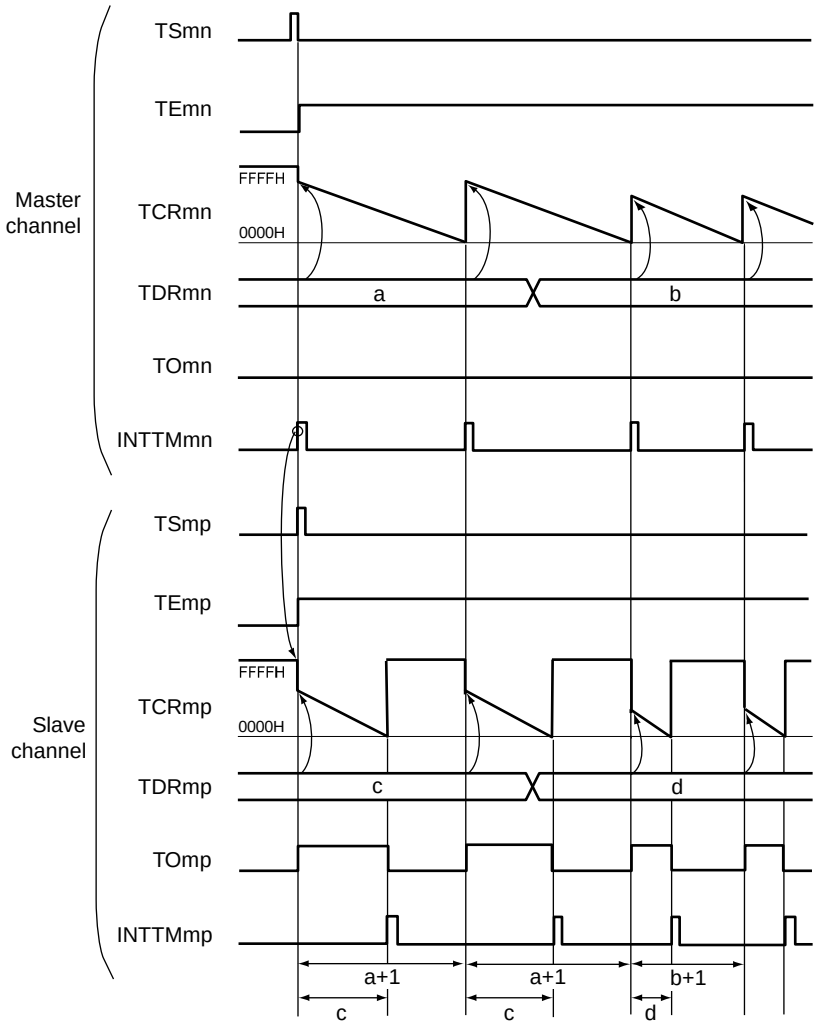


Remarks 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

3. In case of channels 0 and 2 of unit 1 in RL78/F22 products, the clock can not be selected CK12 and CK13 (see **Figure 6-2** and **Figure 6-12**).

Figure 6-76. Example of Basic Timing of Operation as PWM Function



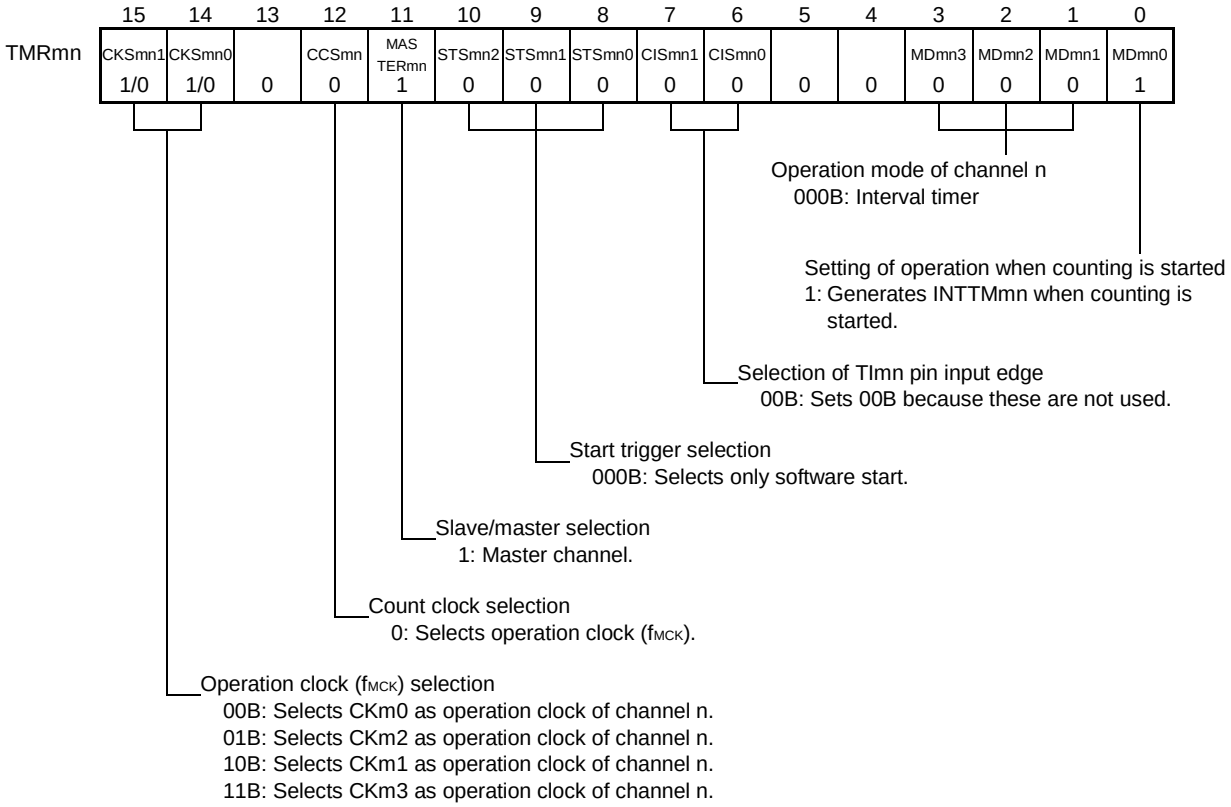
Remarks 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4 6)
p: Slave channel number (n < p ≤ 7)

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
TDRmn, TDRmp: Timer data registers mn, mp
TOmn, TOmp: TOmn and TOmp pins output signal

3. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-77. 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)

	Bit n	
TOm	TOmn	0: Outputs 0 from TOmn.
	0	

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn	0: Stops the TOmn output operation by counting operation.
	0	

(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)

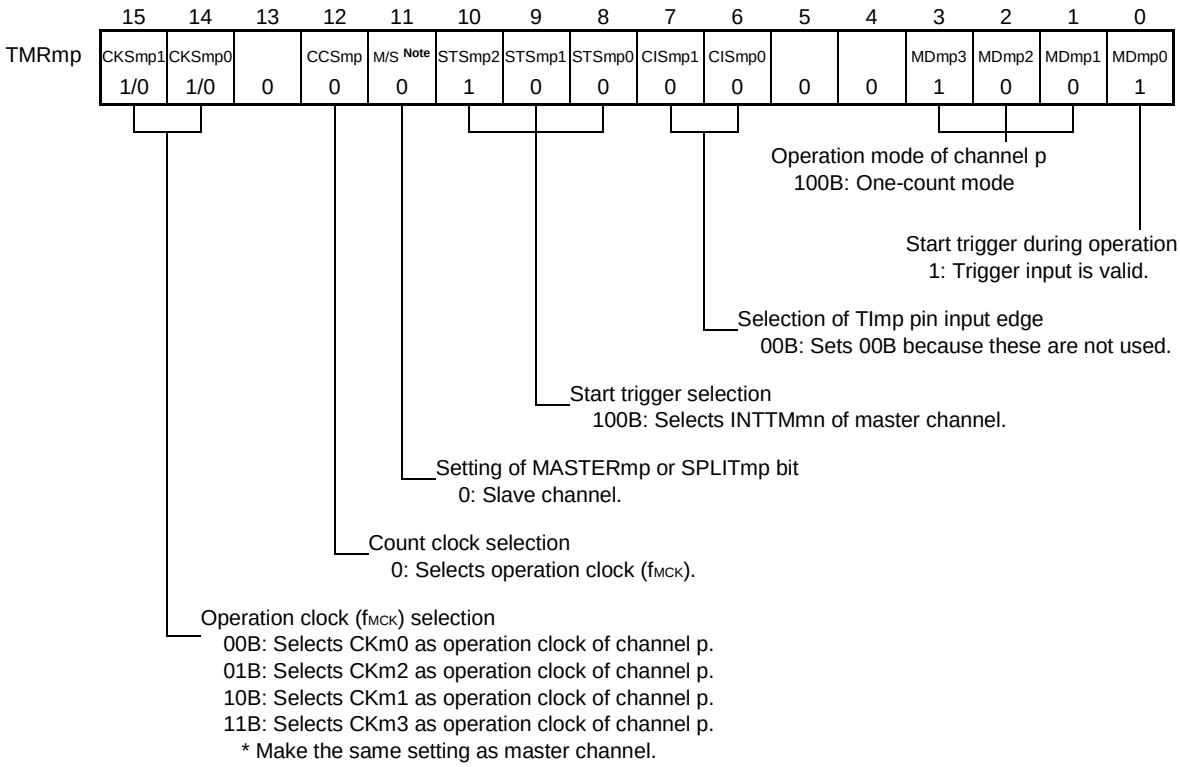
	Bit n	
TOMm	TOMmn	0: Sets master channel output mode.
	0	

Remarks 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-78. 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)

	Bit p	
TOm	TOmp 1/0	0: Outputs 0 from TOmp. 1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p	
TOEm	TOEmp 1/0	0: Stops the TOmp output operation by counting operation. 1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p	
TOLm	TOLmp 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p	
TOMm	TOMmp 1	1: Sets the slave channel output mode.

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-79. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (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. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). 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). 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. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. ——▶	The TOmp pin goes into Hi-Z output state. 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. ▶ Clears the port register and port mode register to 0. ——▶	TOmp does not change because channel stops operating. The TOmp pin outputs the TOmp set level.

(**Remarks** are listed on the next page.)

Figure 6-79. Operation Procedure When PWM Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp = 1 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 and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated. The TCRmn and TCRmp registers can always be read. 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. 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. 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. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	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 Clears the TOmp bit to 0 after the value to be held is set to the port register. → When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 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 In the following cases, the duty factor exceeds 100%, but it is aggregated to 100% output.
{Set value of TDRmp (slave 1)} > {set value of TDRmn (master) + 1} or
{Set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}

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.

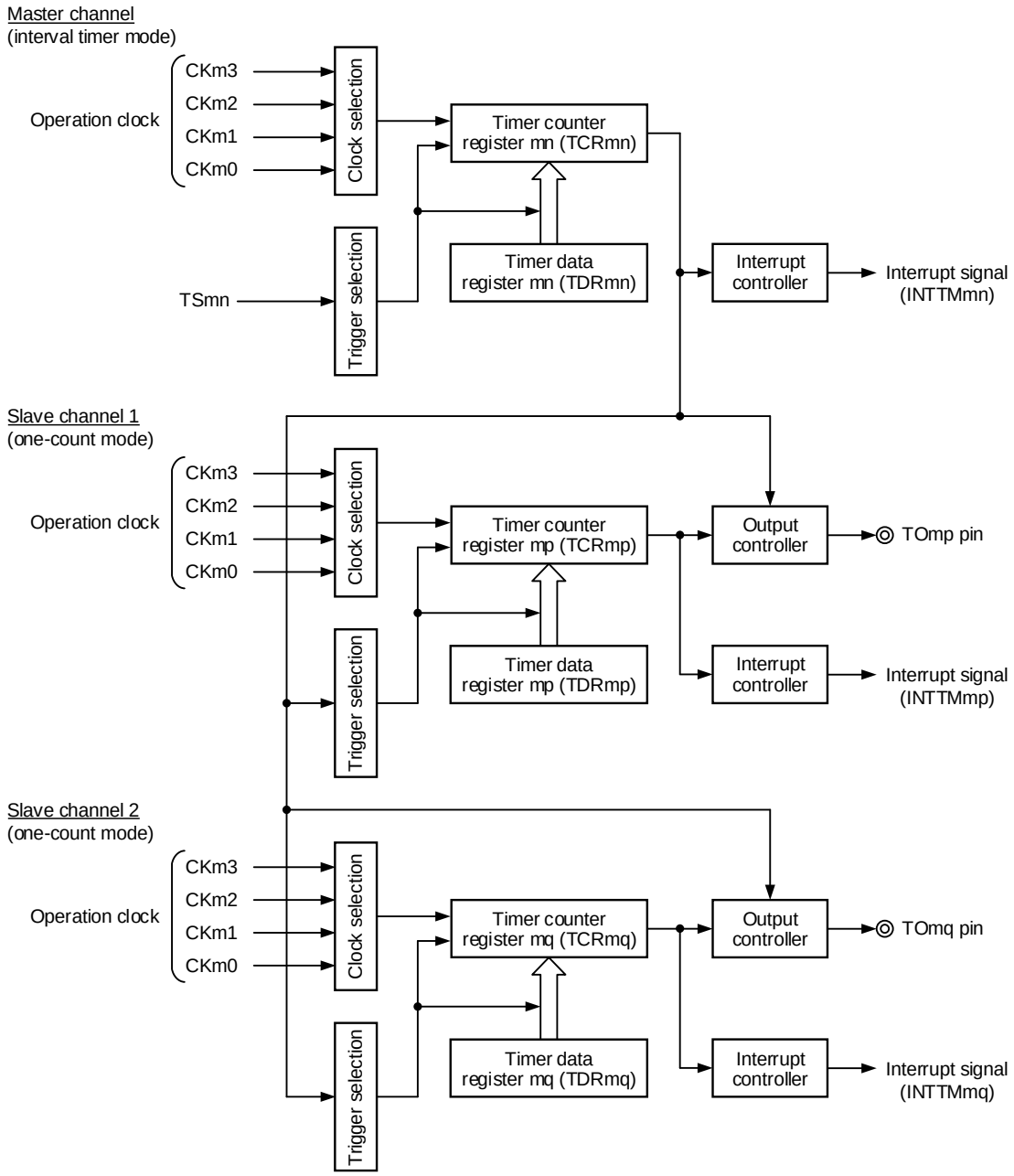
When channel 0 is used as the master channel as above, up to three types of PWM signals can be output at the same time.

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
$n < p < q \leq 7$ (Where p and q are integers greater than n)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-80. Block Diagram of Operation as Multiple PWM Output Function (Output Two Types of PWMs)

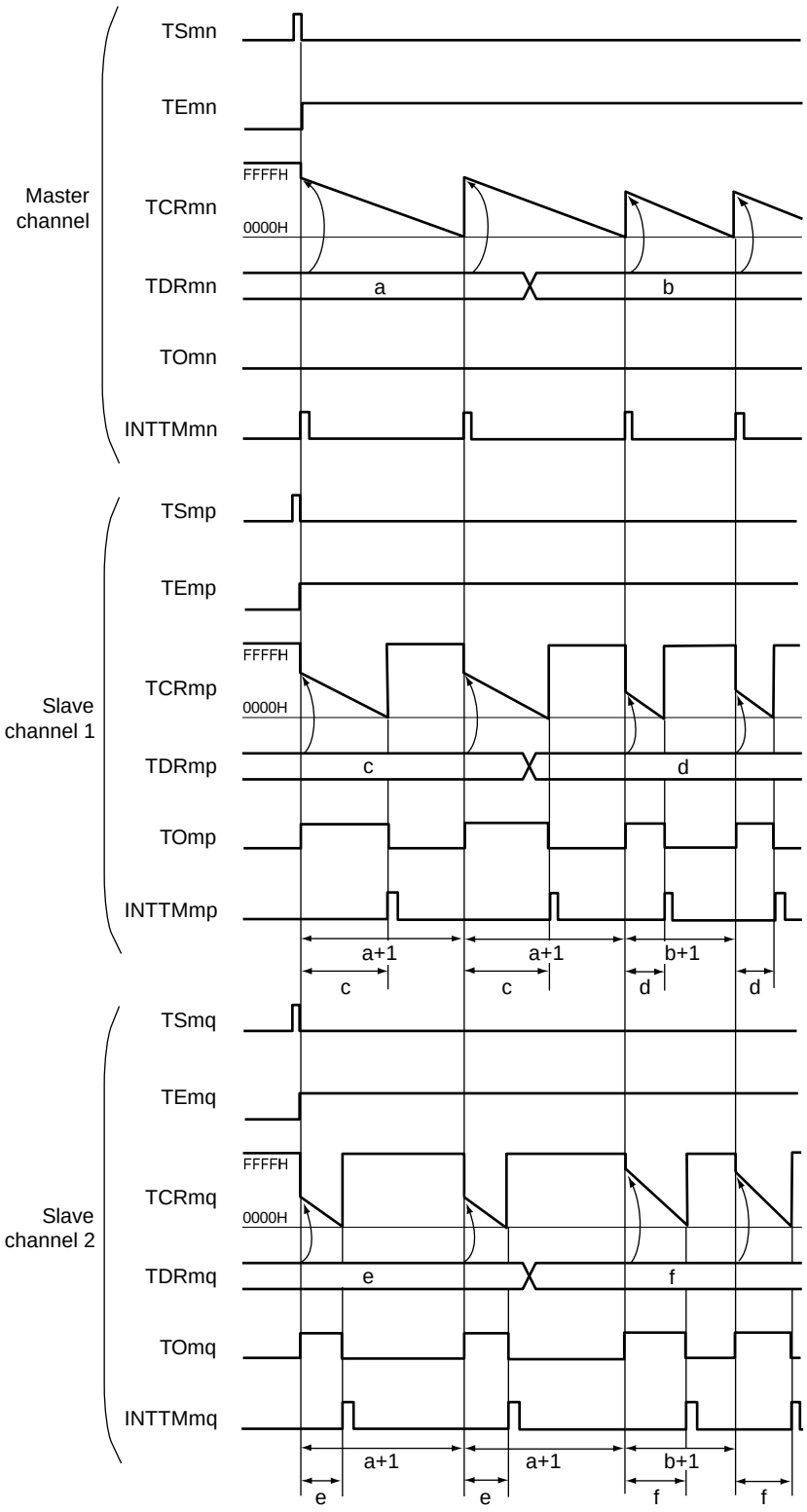


Remarks 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \leq 7$ (Where p and q are integers greater than n)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

3. In case of channels 0 and 2 of unit 1 in RL78/F22 products, the clock can not be selected CK12 and CK13 (see **Figure 6-2** and **Figure 6-12**).

Figure 6-81. Example of Basic Timing of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



(**Remarks** are listed on the next page.)

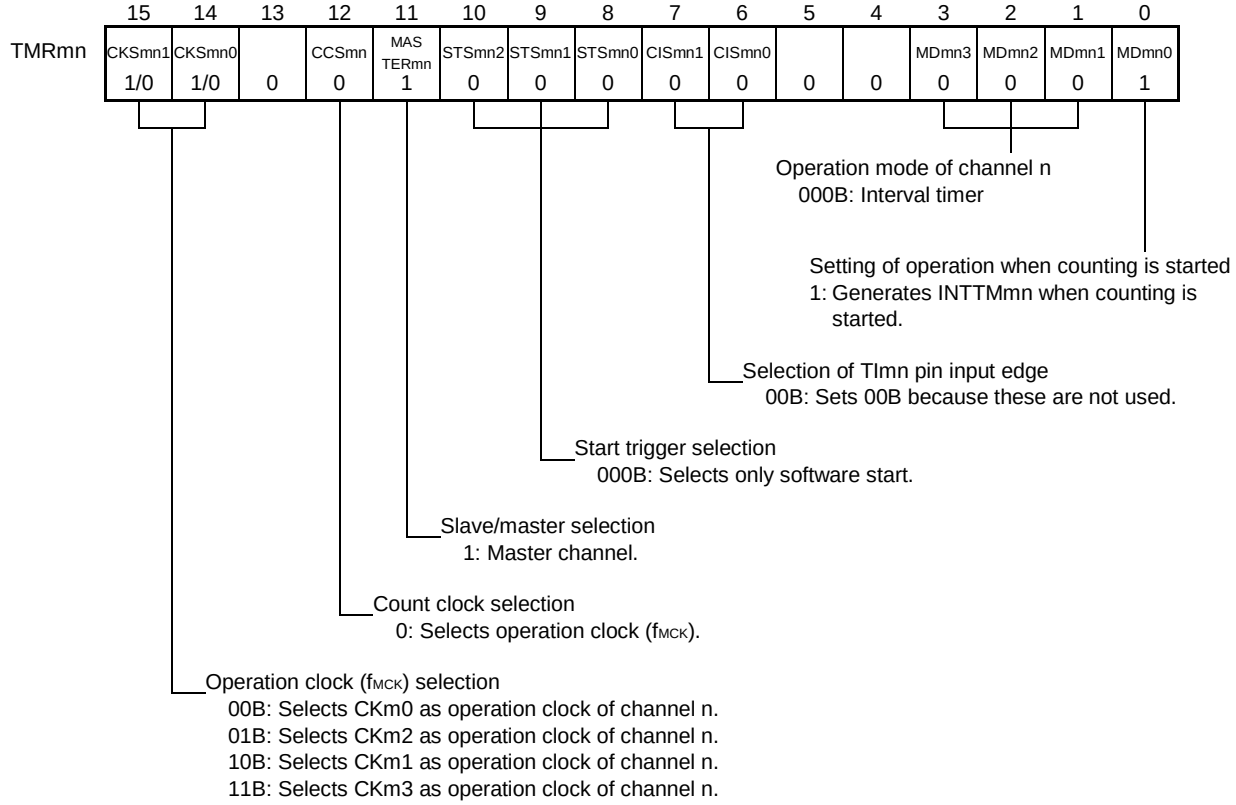
Remarks 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

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
TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq
TOmn, TOmp, TOmq: TOmn, TOmp, and TOmq pins output signal

3. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-82. Example of Set Contents of Registers
When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn	0: Outputs 0 from TOmn.
	0	

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn	0: Stops the TOmn output operation by counting operation.
	0	

(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: Master channel number (n = 0, 2, 4)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-83. 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** and **Remarks** are on the next page.)

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. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

Figure 6-84. 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 (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. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). 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). 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. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOMmp and TOMmq bits of timer output mode register m (TOMm) are set to 1 (slave channel output mode). Clears the TOLmp and TOLmq bits to 0. 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. 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 6-84. Operation Procedure When Multiple PWM Output Function Is Used (Output Two Types of PWMs) (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	(Sets the TOEmp and TOEmq (slave) bits to 1 only when resuming operation.) 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. →	TEmn = 1, TEmp, TEmq = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
	The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits.	
During operation	Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed. Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated. The TCRmn, TCRmp, and TCRmq registers can always be read. 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. 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. 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. 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. →	TEmn, TEmp, TEmq = 0, and count operation stops. The TCRmn, TCRmp, and TCRmq registers hold count value and stop. The TOmp and TOmq output are not initialized but hold current status.
	The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	
	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: Clears the TOmp and TOmq bits to 0 after the value to be held is set to the port register. →	The TOmp and TOmq pin output levels are held by port function.
	When holding the TOmp and TOmq pin output levels are not necessary: Setting not required	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.)

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 a consecutive integer greater than n)

2. Channels 7 to 4 of unit 1 are not provided in the RL78/F22 products.

6.9 Cautions When Using Timer Array Unit

6.9.1 Cautions When Using Timer Output

(1) When the f_{CLK} (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.

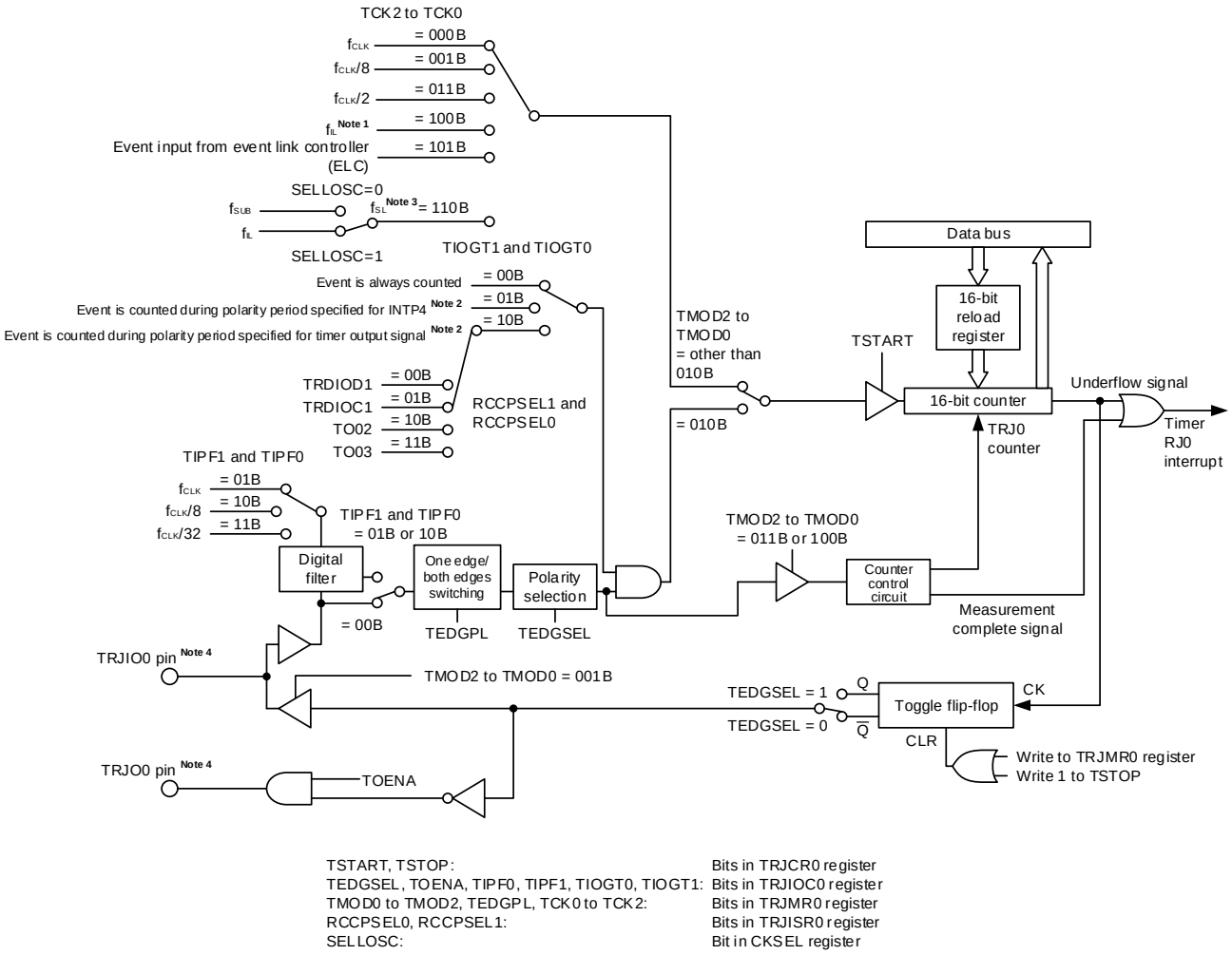
Table 7-1. Timer RJ Specifications

Item		Description
Operating modes	Timer mode	The count source is counted.
	Pulse output mode Note 1	The count source is counted and the output is inverted at each underflow of the timer.
	Event counter mode Note 1	• An external event is counted. • Operation is possible in STOP mode.
	Pulse width measurement mode Note 1	An external pulse width is measured.
	Pulse period measurement mode Note 1	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 Note 2) selectable
Interrupt		• When the counter underflows. • When the measurement of the active width of the external input (TRJIO0) is completed in pulse width measurement mode. • 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 Note 2). • Event input from the ELC is selectable as a count source. • Timer RJ0 interrupt (INTTRJ0) can be a event output to ELC.

Notes 1. The RL78/F22 24-pin products do not have TRJIO0 and TRJO0 pins. Therefore, pulse output mode, event counter mode, pulse width measurement mode, and pulse period measurement mode cannot be used.

2. The ELC is only available in the RL78/F25.

Figure 7-1. Timer RJ Block Diagram



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 RDe or the output clock for clock output/buzzer output.

4. The RL78/F22 24-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 RL78/F22 24-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

Address	Register Name	Symbol	After Reset	Access Size
F00F3H	Operation speed mode control register	OSMC	00H	8
F0240H	Timer RJ control register 0	TRJCR0	00H	8
F0241H	Timer RJ I/O control register 0 Note 2	TRJIOC0	00H	1, 8
F0242H	Timer RJ mode register 0	TRJMR0	00H	1, 8
F0243H	Timer RJ event pin select register 0 Note 2	TRJISR0	00H	1, 8
F02C1H	Peripheral enable register 1	PER1	00H	1, 8
F02CAH	Clock select register	CKSEL	00H	1, 8
F06F0H	Timer RJ counter register 0 Note 1	TRJ0	FFFFH	16

Remark See **7.3.9 Port Mode Registers 1, 4 (PM1, PM4)** for the port mode registers (PMm) and port registers (Pm).

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 RL78/F22 24-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: F02C1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER1	DACEN Note	CTSUEN	CMPEN Note	TRD0EN	DTCEN	PWMOPEN	0	TRJ0EN

TRJ0EN	Control of timer RJ0 input clock supply
0	Stops input clock supply. • SFR used by timer RJ0 cannot be written. • Timer RJ0 is in the reset status.
1	Enables input clock supply. • SFR used by timer RJ0 can be read and written.

Note Only on the RL78/F25.

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/F22: bits 1, 5, and 7
RL78/F25: bit 1

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 stopped
1	Low-speed on-chip oscillator operating

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: F02CAH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	<2>	1	<0>
CKSEL	FPLLDIV	FMAINDIV1	FMAINDIV0	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 and stopping the low-speed on-chip oscillator.
1	Selects f_{IL} Note 2 and running the low-speed on-chip oscillator.

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 RL78/F22 24-, 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.

The ELC is only available in the RL78/F25.

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)

The TRJCR0 register starts or stops count operation and indicates the status of timer RJ.

The TRJCR0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

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	R/W
0	No underflow	R/W
1	Underflow	
[Condition for setting to 0] • When 0 is written to this bit by a program. [Condition for setting to 1] • When the counter underflows.		

TEDGF	Active edge judgement flag Note 3	R/W
0	No active edge received	R/W
1	Active edge received	
[Condition for setting to 0] • When 0 is written to this bit by a program. [Conditions for setting to 1] • When the measurement of the active width of the external input (TRJIO0) is completed in pulse width measurement mode. • 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 1	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 2	R/W
0	Count stops	R
1	Count in progress	
[Conditions for setting to 0] • When 0 is written to the TSTART bit (the TCSTF bit is set to 0 in synchronization with the count source). • When 1 is written to the TSTOP bit. [Condition for setting to 1] • 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 2	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. 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.

2. For notes on using the TSTART and TCSTF bits, see **7.5.1 Count Operation Start and Stop Control.**

3. RL78/F22 24-pin products do not have the active edge judgement flag (TEDGF).

7.3.6 Timer RJ I/O Control Register 0 (TRJIOC0)

The TRJIOC0 register sets the input/output of timer RJ.

The TRJIOC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

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 RL78/F22 24-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) 1: Output is started at low (Initialization level: Low)
Event counter mode	0: Count at rising edge 1: Count at falling edge
Pulse width measurement mode	0: Low-level width is measured 1: High-level width is measured
Pulse period measurement mode	0: Measure from one rising edge to the next rising edge 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) 1: Output is started at high (Initialization level: High)

7.3.7 Timer RJ Mode Register 0 (TRJMR0)

The TRJMR0 register sets the operating mode of timer RJ.

The TRJMR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

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	
0	1	0	Event counter mode	
0	1	1	Pulse width measurement mode	
1	0	0	Pulse period measurement mode	
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/F25. Do not make the setting in other products.

6. The TEDGPL bit is enabled only in event counter mode.

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)

The TRJISR0 register selects the timer for controlling the event count period and sets the polarity in event counter mode.

The TRJISR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

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	RCCPSEL1	RCCPSEL0

Bits 7 to 3	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	R

RCCPSEL2 **Note**	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 **Note**	RCCPSEL0 **Note**	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 RL78/F22 24-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 (PMmn) bit and the port register (Pmn) 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 (PMmn) bit corresponding to each port to 1. At this time, the port register (Pmn) 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) (RL78/F25 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 RL78/F25 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.**

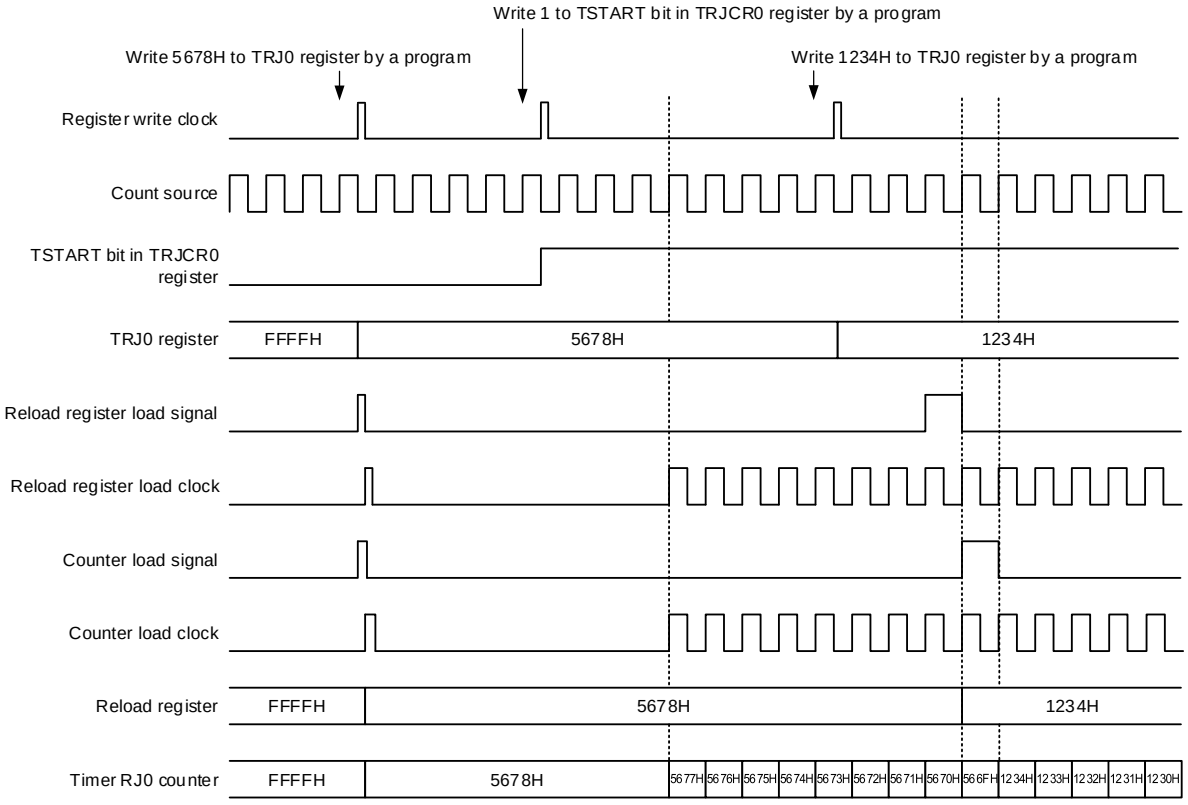
7.4 Operation

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



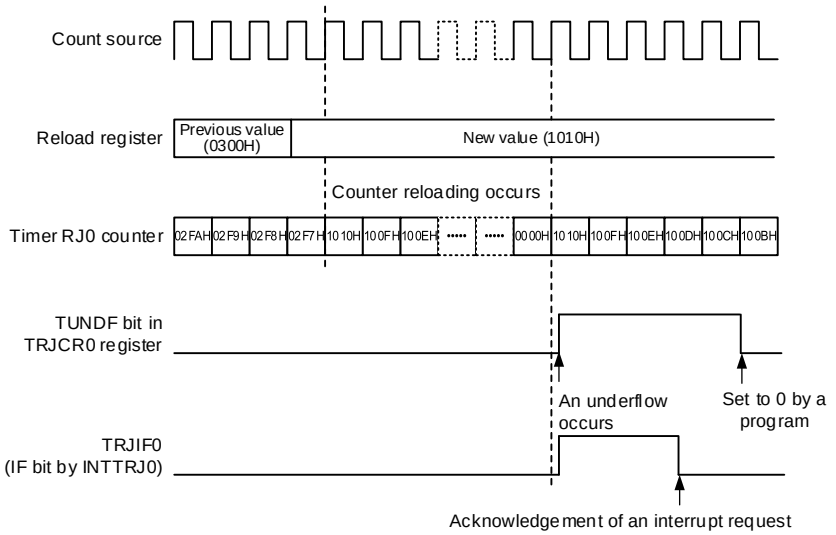
7.4.2 Timer Mode

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.

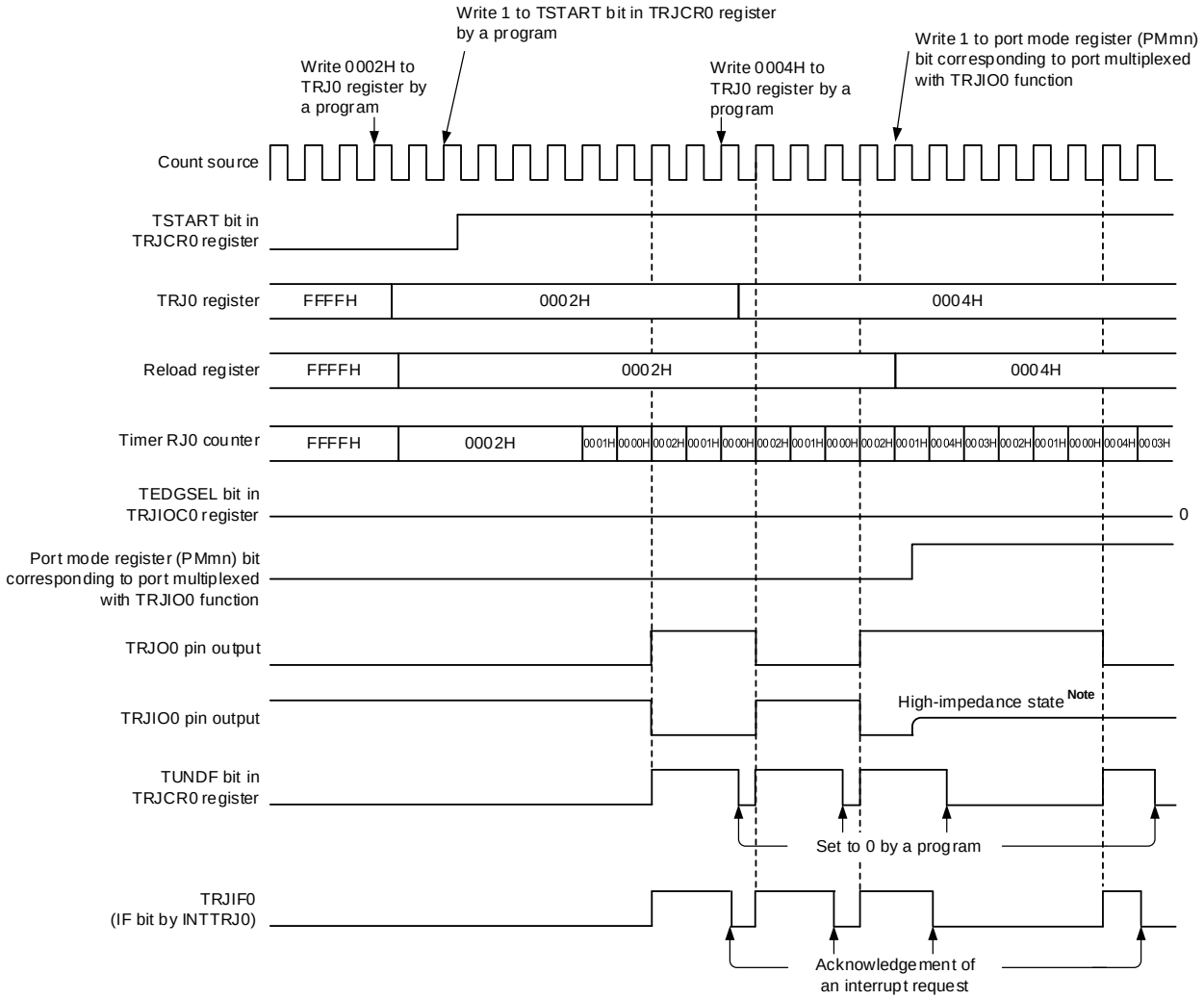
In addition, a pulse can be output from pins TRJIO0 and TRJO0. The output level is inverted each time an underflow occurs. The pulse output from the TRJO0 pin can be stopped by the TOENA bit in the TRJIOC0 register.

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 RL78/F22 24-pin products do not have TRJIO0 and TRJO0 pins. Therefore, the pulse output mode cannot be used.

Figure 7-13. Operation Example in Pulse Output Mode



Note 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 RL78/F22 24-pin products do not have TRJIO0 and TRJO0 pins. Therefore, the event counter mode cannot be used.

Figure 7-14. Operation Example 1 in Event Counter Mode

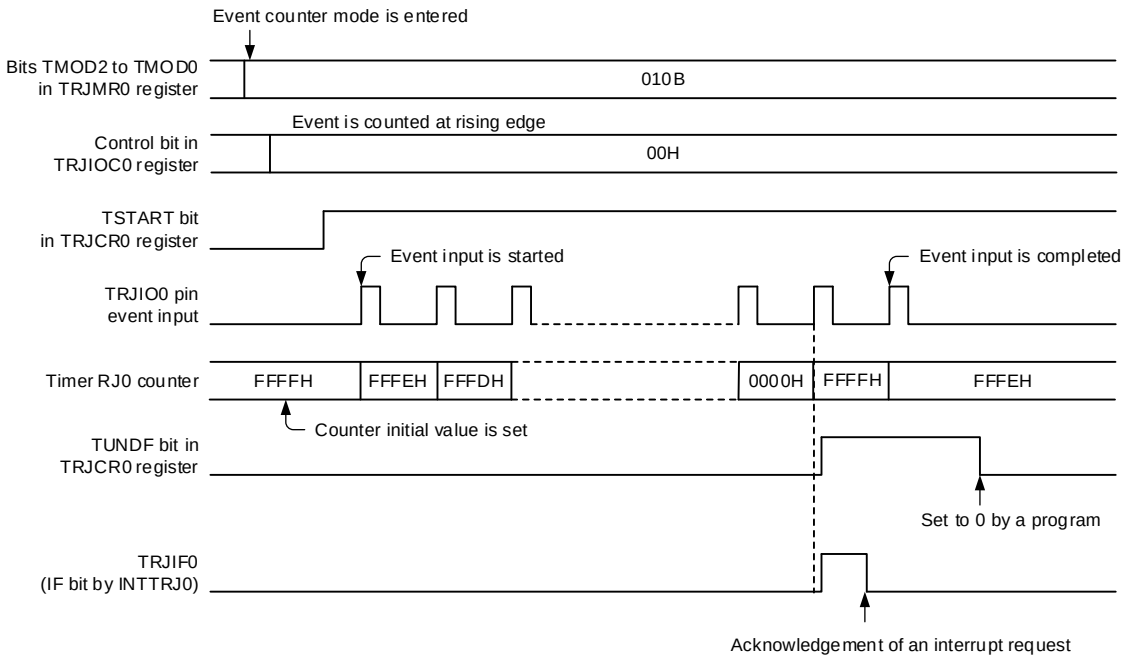
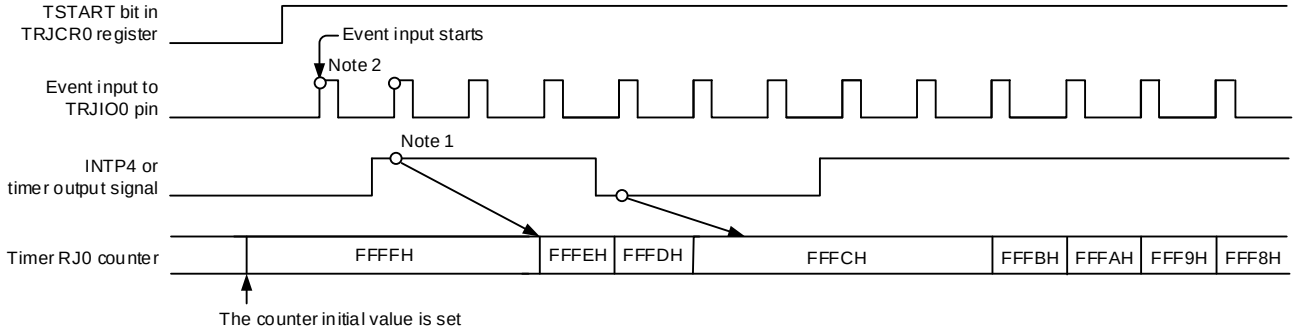


Figure 7-15 shows an operation example for counting during the specified period in event counter mode (bits TIOGT1 and TIOGT0 in the TRJIOC0 register are set to 01B or 10B).

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.

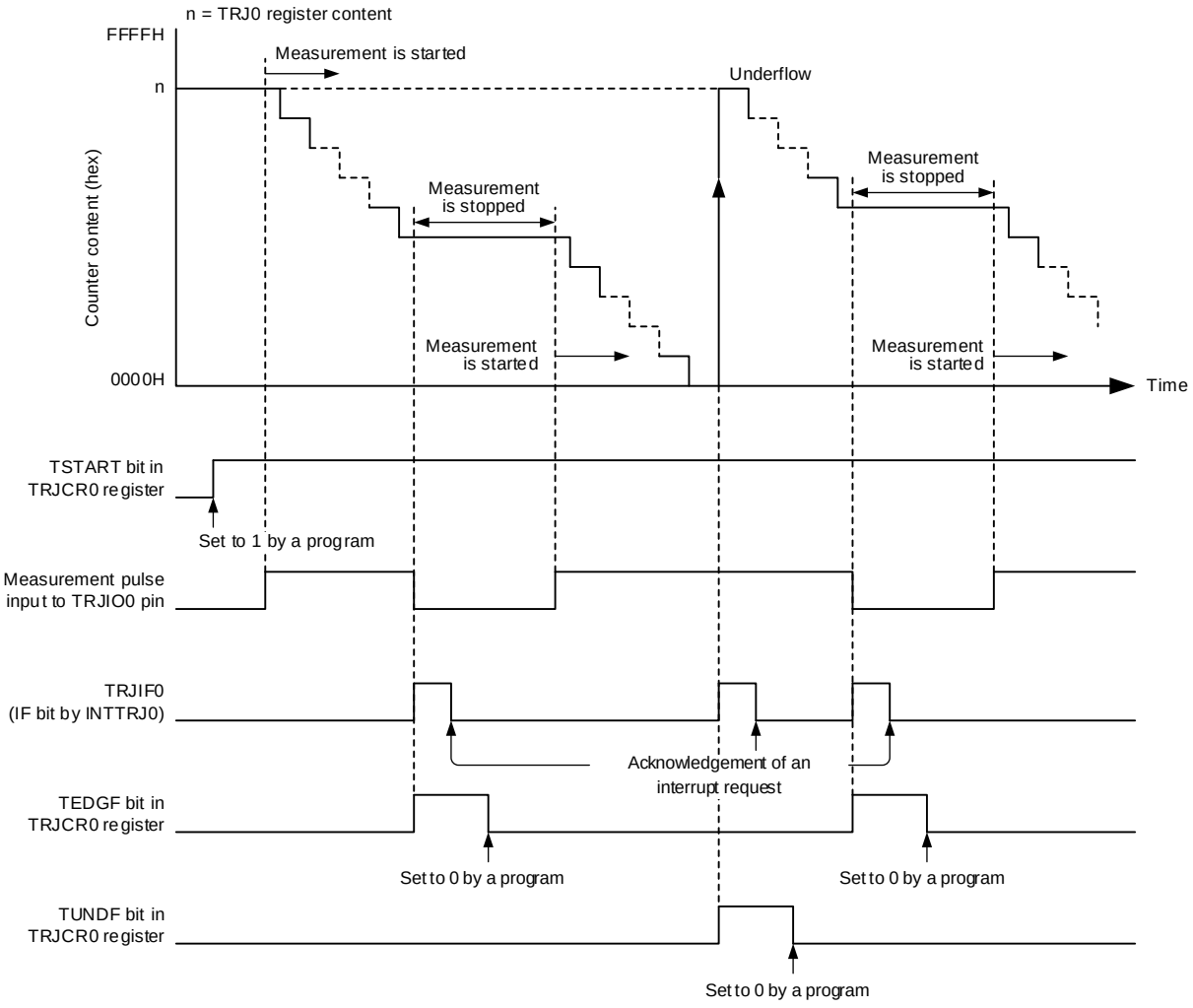
Figure 7-16 shows the Operation Example in Pulse Width Measurement Mode.

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 RL78/F22 24-pin products do not have TRJIO0 and TRJO0 pins. Therefore, 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.

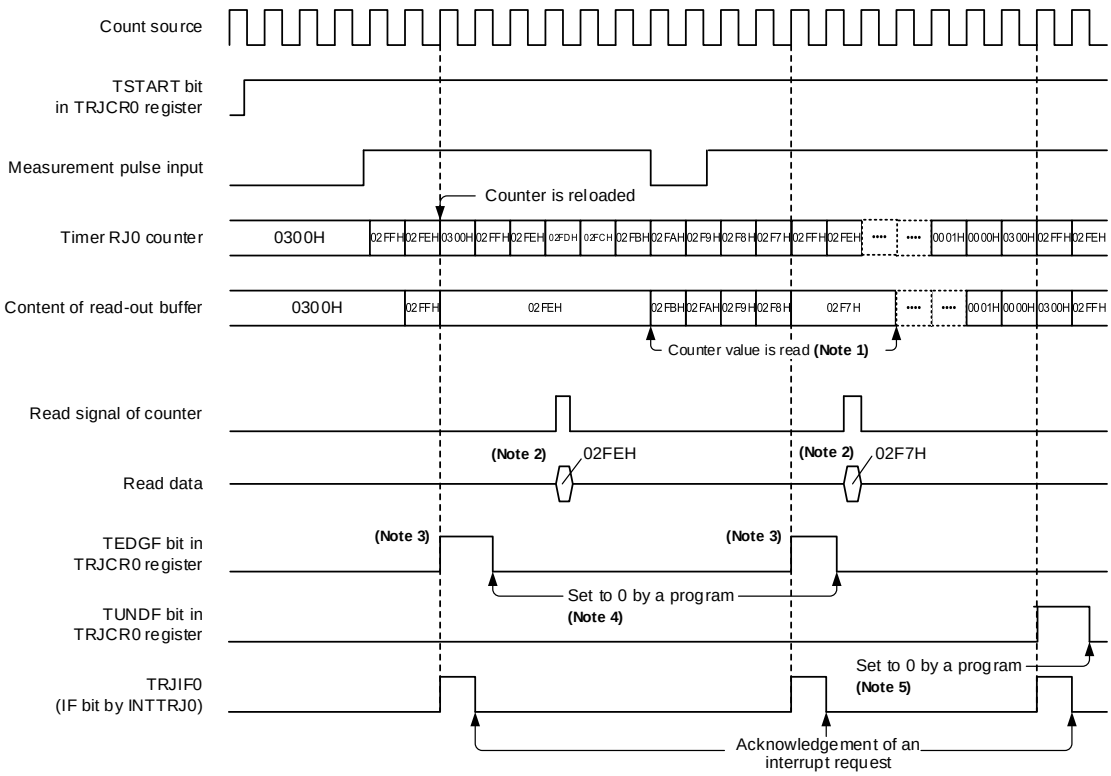
The counter is decremented by the count source selected by bits TCK0 to TCK2 in the TRJMR0 register. When a pulse with the period specified by the TEDGSEL bit in the TRJIOC0 register is input to the TRJIO0 pin, the count value is transferred to the read-out buffer at the rising edge of the count source. 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 RL78/F22 24-pin products do not have TRJIO0 and TRJO0 pins. Therefore, 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/F25.

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	Port Mode Register 4	TRJIOC0 Register	TRJIO0 Pin I/O
	PM41	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			

7.5 Notes on Timer RJ

7.5.1 Count Operation Start and Stop Control

- **When in a mode other than event counter mode, 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 22 INTERRUPT FUNCTIONS** for details.

  Note Registers associated with timer RJ: TRJ0, TRJCR0, TRJIOC0, TRJMR0, and TRJISR0

- **When in event counter mode, 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 22 INTERRUPT FUNCTIONS** for details.
  The ELC is only available in the RL78/F25.

  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 RL78/F22 24-pin products do not have TRJIO0 and TRJO0 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 RDe

8.1 Overview

Timer RDe contains two 16-bit timer units (timer RD0 and timer RD1).

Each of timer RD0 and timer RD1 has four I/O pins.

The timer RDe operating clock (f_{TRD}) is selectable from f_{CLK}, f_{MP}, or f_{SL}.

Figure 8-1 shows the Timer RDe Block Diagram.

Refer to **8.6 PWM Option Unit A (PWMOPA)** for the PWMOPA function and **8.7 Interrupt Decimation Module (TRDMBK)** for the decimation function.

Table 8-1 lists the Timer RDe Pin Configuration.

Timer RDe has six 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 five 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
- Extended PWM mode: Output PWM waveforms (4 (2 in counter)) with a fixed period
  And it has reload function of compare registers, counter restart function, dithering and gate function as extensions.
- Extended complementary PWM mode:
  Output symmetry/asymmetry three-phase waveforms (6) with triangular wave modulation and dead time

The timer mode (input capture function, output compare function, and PWM function) and extended PWM mode 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.

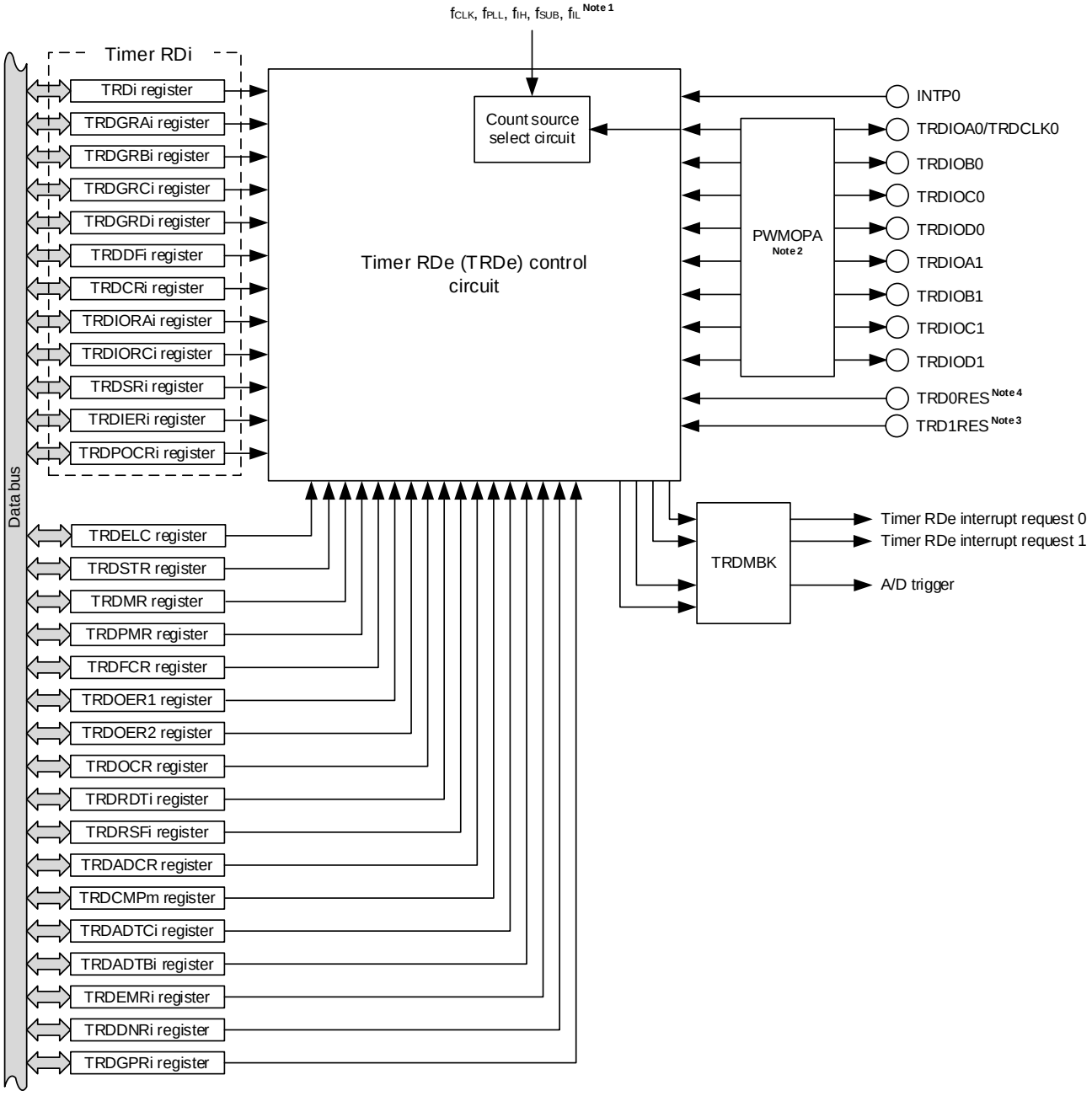
In reset synchronous PWM mode, complementary PWM mode, PWM3 mode, and extended complementary PWM mode, a waveform is output with a combination of counters and registers in timer RD0 and timer RD1. Pin functions depend on the mode.

The PWM option unit A (PWMOPA) is used to cutoff and release the output from timer RDe and ports with the comparator 0 output, external interrupt 0 (INTP0), and event link controller (ELC) as trigger signals.

This function is available to do software release or hardware release which is selected for output forced cutoff release. See **8.6 PWM Option Unit A (PWMOPA)**.

The interrupt decimation module (TRDMBK) can decimate and output the timer RDe request signals for interrupt and A/D triggers. See **8.7 Interrupt Decimation Module (TRDMBK)**.

Figure 8-1. Timer RDe Block Diagram



Notes 1. f_{IH} can be selected when it is 80MHz or 64 MHz. f_{PLL} can be selected when it is over 40 MHz.

2. Output signals can be cut off, while input signals cannot.

3. TRD1RES pin is not available in the RL78/F22 24-, 32-pin products, so be sure to set 00B for CCLV1, CCLV0 bits of the TRDEMR1 register in the RL78/F22 24-, 32-pin products.

4. TRD0RES pin is not available in the RL78/F22 24-pin products, so be sure to set 00B for CCLV1, CCLV0 bits of the TRDEMR0 register in the RL78/F22 24-pin products.

Remark i = 0 or 1, m = B0, D0, A1, B1, C1, D1

Table 8-1. Timer RDe Pin Configuration

Pin Name	Assigned Pin	I/O	Function
TRDIOA0/TRDCLK0	P13 (P15)	Input/Output	Function varies depending on the mode. 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	
TRD0RES	P41	Input	TRD0 counter clear input
TRD1RES	P140	Input	TRD1 counter clear input

8.2 Registers

Table 8-2 lists the Timer RDe Register Configuration.

Table 8-2. Timer RDe Register Configuraion (1/2)

Address	Register Name	Symbol	After Reset	Access Size
F02C1H	Peripheral enable register 1	PER1	00H	1, 8
F02CAH	Clock select register	CKSEL	00H	1, 8
F0250H	Timer RDe ELC register	TRDELC	00H Note 1	1, 8
F0253H	Timer RDe start register	TRDSTR	0CH Note 1	8
F0254H	Timer RDe mode register	TRDMR	00H Note 1	1, 8
F0255H	Timer RDe PWM function select register	TRDPMR	00H Note 1	1, 8
F0256H	Timer RDe function control register	TRDFCR	80H Note 1	1, 8
F0257H	Timer RDe output master enable register 1	TRDOER1	FFH Note 1	1, 8
F0258H	Timer RDe output master enable register 2	TRDOER2	00H Note 1	1, 8
F0259H	Timer RDe output control register	TRDOCR	00H Note 1	1, 8
F025AH	Timer RDe digital filter function select register 0	TRDDF0	00H Note 1	1, 8
F025BH	Timer RDe digital filter function select register 1	TRDDF1	00H Note 1	1, 8
F0260H	Timer RDe control register 0	TRDCR0	00H Note 1	1, 8
F0261H	Timer RDe I/O control register A0	TRDIORA0	00H Note 1	1, 8
F0262H	Timer RDe I/O control register C0	TRDIORC0	88H Note 1	1, 8
F0263H	Timer RDe status register 0	TRDSR0	00H Note 1	1, 8
F0264H	Timer RDe interrupt enable register 0	TRDIER0	00H Note 1	1, 8
F0265H	Timer RDe PWM output level control register 0	TRDPOCR0	00H Note 1	1, 8
F0266H	Timer RDe counter 0	TRD0	0000H Note 1	16
F0268H	Timer RDe general register A0	TRDGRA0	FFFFH Note 1	16
F026AH	Timer RDe general register B0	TRDGRB0	FFFFH Note 1	16
F0270H	Timer RDe control register 1	TRDCR1	00H Note 1	1, 8
F0271H	Timer RDe I/O control register A1	TRDIORA1	00H Note 1	1, 8
F0272H	Timer RDe I/O control register C1	TRDIORC1	88H Note 1	1, 8
F0273H	Timer RDe status register 1	TRDSR1	40H Note 1	1, 8
F0274H	Timer RDe interrupt enable register 1	TRDIER1	00H Note 1	1, 8
F0275H	Timer RDe PWM output level control register 1	TRDPOCR1	00H Note 1	1, 8
F0276H	Timer RDe counter 1	TRD1	0000H Note 1	16
F0278H	Timer RDe general register A1	TRDGRA1	FFFFH Note 1	16
F027AH	Timer RDe general register B1	TRDGRB1	FFFFH Note 1	16
F0280H	Timer RDe extended compare register B0	TRDCMPB0	FFFFH Note 1	16
F0284H	Timer RDe extended compare register A1	TRDCMPA1	FFFFH Note 1	16
F0288H	Timer RDe extended compare register B1	TRDCMPB1	FFFFH Note 1	16
F028CH	Timer RDe A/D trigger compare register 0	TRDADTC0	FFFFH Note 1	16
F0290H	Timer RDe A/D trigger compare register 1	TRDADTC1	FFFFH Note 1	16
F0296H	Timer RDe reload status flag register	TRDRSF01	0000H Note 1	16
	Timer RDe reload status flag register 0	TRDRSF0	00H Note 1	1, 8
F0297H	Timer RDe reload status flag register 1	TRDRSF1	00H Note 1	1, 8
F0298H	Timer RDe A/D trigger control register	TRDADCR	00H Note 1	1, 8
F029AH	Timer RDe extended PWM mode register 0	TRDEMR0	00H Note 1	1, 8
F029BH	Timer RDe extended PWM mode register 1	TRDEMR1	00H Note 1	1, 8
FFF58H	Timer RDe general register C0	TRDGRC0	FFFFH Note 1	16
FFF5AH	Timer RDe general register D0	TRDGRD0	FFFFH Note 1	16
FFF5CH	Timer RDe general register C1	TRDGRC1	FFFFH Note 1	16
FFF5EH	Timer RDe general register D1	TRDGRD1	FFFFH Note 1	16

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

Table 8-2. Timer RDe Register Configuration (2/2)

Address	Register Name	Symbol	After Reset	Access Size
FFF60H	Timer RDe extended compare register D0	TRDCMPD0	FFFFH Note 1	16
FFF62H	Timer RDe extended compare register C1	TRDCMPC1	FFFFH Note 1	16
FFF64H	Timer RDe extended compare register D1	TRDCMPD1	FFFFH Note 1	16
FFF66H	Timer RDe A/D trigger buffer register 0	TRDADTB0	FFFFH Note 1	16
FFF68H	Timer RDe A/D trigger buffer register 1	TRDADTB1	FFFFH Note 1	16
FFF6AH	Timer RDe reload trigger register	TRDRDT01	0000H Note 1	16
	Timer RDe reload trigger register 0	TRDRDT0	00H Note 1	1, 8
FFF6BH	Timer RDe reload trigger register 1	TRDRDT1	00H Note 1	1, 8
FFF6CH	Timer RDe dithering/gate control register 0	TRDDGCR0	0000H Note 1	16
	Timer RDe dithering number register 0	TRDDNR0	00H Note 1	1, 8
FFF6DH	Timer RDe gate pattern register 0	TRDGPR0	00H Note 1	1, 8
FFF6EH	Timer RDe dithering/gate control register 1	TRDDGCR1	0000H Note 1	16
	Timer RDe dithering number register 1	TRDDNR1	00H Note 1	1, 8
FFF6FH	Timer RDe gate pattern register 1	TRDGPR1	00H Note 1	1, 8
F0228H	PWM output delay control register 0	PWMDLY0	0000H	16
F02C8H	PLL control register	PLLCTL	00H	1, 8
F02C9H	PLL status register	PLLSTS	00H	1, 8
F02CBH	f_{MP} clock division register	MDIV	00H/01H Note 2	8
FFFA4H	System clock control register	CKC	00H	1, 8

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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. The value of the FRQSEL4 bit in the user option byte (000C2H/040C2H) becomes the initial value of the MDIV0 bit in the MDIV register.

Remark See **8.2.33 Port Mode Registers (PM1, PM3, PM4, PM12, PM14)** for the port mode registers (PMxx) and port registers (Pxx).

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 RDe, 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: F02C1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER1	DACEN Note	CTSUEN	CMPEN Note	TRD0EN	DTCEN	PWMOPEN	0	TRJ0EN

TRD0EN	Control of timer RDe input clock supply
0	Stops input clock supply. • SFR used by timer RDe cannot be written. • Timer RDe is in the reset status.
1	Enables input clock supply. • SFR used by timer RDe can be read and written.

Note Only for RL78/F25.

Cautions 1. When setting timer RDe, be sure to set the TRD0EN bit to 1 first. If TRD0EN = 0, writing to a control register of timer RDe is ignored, and all read values are default values (except for port mode registers 1, 3, 4, 12, and 14 (PM1, PM3, PM4, PM12, and PM14) and port registers 1, 3, 4, 12, and 14 (P1, P3, P4, P12, and P14).

2. Be sure to clear the following bits to 0.
RL78/F22: bits 1, 5, and 7
RL78/F25: bit 1

3. When selecting f_{IH} (80MHz or 64 MHz) as the count source, set f_{TRD} to f_{IH}. When selecting f_{PLL} (over 40 MHz) as the count source, set f_{TRD} to f_{PLL}. When selecting f_{SUB} or f_{IL} as the count source to access the timer RDe-related registers, set f_{TRD} to f_{SUB} or f_{IL}, respectively.

4. When setting PWMOPA, be sure to set the PWMOPEN bit to 1 first. If PWMOPEN = 0, writing to a control register of PWMOPA is ignored, and all read values are default values. For PWMOPA, see 8.6 PWM Option Unit A (PWMOPA).

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 RDe, 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: F02CAH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	<2>	1	<0>
CKSEL	FPLLDIV	FMAINDIV1	FMAINDIV0	0	0	TRD_CKSEL	0	SELLOSC Notes 5, 6, 7

FPLLDIV	Control of PLL frequency division selection
0	PLLCTL.PLLDIV0 enabled
1	No division

FMAINDIV1	FMAINDIV0	Control of PLL input clock (f_{PLLI}) division selection
0	0	No division
1	0	Divided by 2 (f_{MAIN} = 16 MHz input only)
1	1	Divided by 4 (f_{MAIN} = 20 MHz input only)
0	1	Setting prohibited

TRD_CKSEL	Control of timer RDe clock selection
0	Selects f_{CLK} or f_{MP} Note 1
1	Selects f_{SL} Note 2

SELLOSC Notes 5, 6, 7	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/040C2H) or PLLDIV1 = 1 (f_{PLL} > 40 MHz) in the PLLCTL register, set the TRD_CKSEL bit to 0.
   When FRQSEL4 = 1 in the user option byte (000C2H/040C2H) or PLLDIV1 = 1 (f_{PLL} > 40 MHz) in the PLLCTL register, the timer RDe clock becomes f_{MP}.
2. When f_{SL} is selected as the timer RDe 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 RL78/F22 24-, 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.

8.2.3 Timer RDe ELC Register (TRDELC)

This register is used to select the ELC event input.

This register is only available in the RL78/F25 product.

Set the TRDELC register by a 1-bit or 8-bit memory manipulation instruction.

Figure 8-4. Format of Timer RDe ELC Register (TRDELC)

Address: F0250H After Reset: 00H [Note]

Symbol	7	6	5	4	3	2	1	0
TRDELC	0	0	ELCOBE1	ELCICE1	0	0	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 RDe 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 RDe 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 RDe 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 RDe 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	

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

Caution The TRDELC register is only available in the RL78/F25 products. Do not access with RL78/F22 products.

8.2.4 Timer RDe Start Register (TRDSTR)

This register is used to select the TRDi count start and TRDi count operation.

Set the TRDSTR register by an 8-bit memory manipulation instruction. See **8.5.1 (1) TRDSTR Register**.

Figure 8-5. Format of Timer RDe Start Register (TRDSTR)

Address: F0253H After Reset: 0CH [Note 1]

Symbol	7	6	5	4	3	2	1	0
TRDSTR	0	0	0	0	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/040C2H) 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 (compare match between TRD1 and TRDGRA1) 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 (compare match between TRD0 and TRDGRA0) is generated, this flag is set to 0 (count stops).

8.2.5 Timer RDe Mode Register (TRDMR)

This register is used to select the function of TRDGRCi and TRDGRDi (i = 0, 1) registers, and the synchronous operation of TRD0 and TRD1.

Set the TRDMR register by a 1-bit or 8-bit memory manipulation instruction.

Figure 8-6. Format of Timer RDe Mode Register (TRDMR)

Address: F0254H 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, 3]	R/W
0	General register	R/W
1	Buffer register for TRDGRB1 register	

TRDBFC1	TRDGRC1 register function select [Notes 2, 3]	R/W
0	General register	R/W
1	Buffer register for TRDGRA1 register	

TRDBFD0	TRDGRD0 register function select [Notes 2, 3]	R/W
0	General register	R/W
1	Buffer register for TRDGRB0 register	

TRDBFC0	TRDGRC0 register function select [Notes 2, 3, 4]	R/W
0	General register	R/W
1	Buffer register for TRDGRA0 register	

Bits 3 to 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	R

TRDSYNC	Timer RDe synchronous [Note 5]	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/040C2H) 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. This bit is not used in extended PWM mode and extended complementary PWM mode.
4. Set to 0 (general register) in complementary PWM mode.
5. Set to 0 (TRD0 and TRD1 operate independently) in reset synchronous PWM mode, complementary PWM mode, PWM3 mode, and extended complementary PWM mode.

8.2.6 Timer RDe PWM Function Select Register (TRDPMR)

This register is used to select the PWM function.

This register is not used in reset synchronous PWM mode, complementary PWM mode, PWM3 mode, and extended complementary PWM mode.

Settings to the TRDPOCRi register is enabled only in the PWM function of the timer mode or the extended PWM mode.

Set the TRDPMR register by a 1-bit or 8-bit memory manipulation instruction.

Figure 8-7. Format of Timer RDe PWM Function Select Register (TRDPMR)

[Timer mode / Extended PWM mode]

Address: F0255H After Reset:00H [Note 1]

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 [Note 2]	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 [Note 2]	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	

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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 0 (TRDIOC0 and TRDIOC1 are not used as PWM function) in extended PWM mode.

8.2.7 Timer RDe Function Control Register (TRDFCR)

This register is used to select the PWM function and the external clock input.

Set the TRDFCR register by a 1-bit or 8-bit memory manipulation instruction.

Figure 8-8. Format of Timer RDe Function Control Register (TRDFCR)

Address: F0256H After Reset: 80H Note 1

Symbol	7	6	5	4	3	2	1	0
TRDFCR	PWM3	STCLK	EPWM	CPSS	OLS1	OLS0	CMD1	CMD0

PWM3	PWM3 mode select Note 2	R/W
	• In timer mode and extended PWM mode, set to 1 (other than PWM3 mode). • In PWM3 mode, set to 0 (PWM3 mode). • Disabled in reset synchronous PWM, complementary PWM, and extended complementary PWM modes.	R/W

STCLK	External clock input select	R/W
	• In timer mode, reset synchronous PWM mode, complementary PWM mode, extended PWM mode, and extended complementary PWM mode, 0: External clock input disabled 1: External clock input enabled • In PWM3 mode, set to 0 (external clock input disabled).	R/W

EPWM	Extended complementary PWM mode or extended PWM mode select	R/W
	• In extended complementary PWM mode or extended PWM mode, set to 1. • In other than extended complementary PWM mode or extended PWM mode, set to 0.	R/W

CPSS	Extended complementary PWM mode action select	R/W
	• In extended complementary PWM mode, 0: Symmetry PWM output 1: Asymmetry PWM output • In other than extended complementary PWM mode, set to 0.	R/W

OLS1	Counter-phase output level select	R/W
	• In reset synchronous PWM mode and complementary PWM mode, 0: High initial output and low active level 1: Low initial output and high active level • In extended complementary PWM modes Note 3, 0: Low active level 1: High active level • Disabled in timer, PWM3. and extended PWM modes.	R/W

OLS0	Normal-phase output level select	R/W
	• In reset synchronous PWM mode and complementary PWM mode, 0: High initial output and low active level 1: Low initial output and high active level • In extended complementary PWM modes Note 3, 0: Low active level 1: High active level • Disabled in timer, PWM3. and extended PWM modes.	R/W

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

CMD1	CMD0	Combination mode select Notes 4, 5	R/W
• In timer, PWM3, and extended PWM modes, set to 00B. • In reset synchronous PWM mode, set to 01B. • In complementary PWM mode, 10B: Transfer from the buffer register to the general register when TRD1 underflows 11B: Transfer from the buffer register to the general register at compare match between registers TRD0 and TRDGRA0 • In extended complementary PWM mode, set to 10B. 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/040C2H) 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, PWM3 mode, or extended PWM mode), the setting of the PWM3 bit is enabled.

3. Set the TRDOCR register with initial output level.

4. Set bits CMD0 and CMD1 when both the TSTART0 and TSTART1 bits in the TRDSTR register are set to 0 (count stops).

5. When bits CMD1 and CMD0 are set to 01B, 10B, or 11B, the MCU enters reset synchronous PWM mode, complementary PWM mode, or extended complementary PWM mode regardless of the settings of the TRDPMR register.

8.2.8 Timer RDe Output Master Enable Register 1 (TRDOER1)

This register is used to select the outputs of TRDIOji (j = A, B, C, D, i = 0, 1) pins.

Set the TRDOER1 register by a 1-bit or 8-bit memory manipulation instruction.

Figure 8-9. Format of Timer RDe Output Master Enable Register 1 (TRDOER1)

[Output compare function, PWM function, Reset synchronous PWM Mode, Complementary PWM mode, PWM3 mode, Extended PWM mode, and Extended complementary PWM mode]

Address: F0257H 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 Notes 2, 3	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, 4	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 Notes 2, 3	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, 5	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/040C2H) 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 extended PWM mode.

4. Set to 1 in PWM function.

5. Set to 1 in reset synchronous PWM mode, complementary PWM mode, and extended complementary PWM mode.

8.2.9 Timer RDe Output Master Enable Register 2 (TRDOER2)

This register is used to select the pulse output forced cutoff function.

Settings to the TRDOER2 register is enabled only in the PWM function of the timer mode, reset synchronouse PWM mode, complementary PWM mode, PWM3 mode, extended PWM mode, or the extended complementary PWM mode.

Set the TRDOER2 register by a 1-bit or 8-bit memory manipulation instruction.

Figure 8-10. Format of Timer RDe Output Master Enable Register 2 (TRDOER2)

[PWM function, Reset synchronous PWM mode, Complementary PWM mode, PWM3 mode, Extended PWM mode, and Extended complementary PWM Mode]

Address: F0258H 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 (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/040C2H) 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. ELC function is only available in the RL78/F25 products.

8.2.10 Timer RDe Output Control Register (TRDOCR)

This register is used to select the output level of TRDIOji (j = A, B, C, D, i = 0, 1) pins.

Write to the TRDOCR register when bits TSTART0 and TSTART1 in the TRDSTR register are both 0 (count stops).

Set the TRDOCR register by a 1-bit or 8-bit memory manipulation instruction.

The specifications of this register differ depending on the operating mode.

Figure 8-11. Format of Timer RDe Output Control Register (TRDOCR)

[Output compare function, Extended Complementary PWM mode]

Address: F0259H 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/040C2H) 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.

[PWM function, Extended PWM mode]

Address: F0259H 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	TRDIOC0 initial output level select [Note 2]	R/W
0	Initial output is not active level	R/W
1	Initial output is active level	
In extended PWM mode, set to 0.		

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	
In extended PWM mode, set to 0.		

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/040C2H) 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.

[Reset synchronous PWM mode, Complementary PWM mode]

Address: F0259H 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
Set to 0. The initial output value is selected by OLS0 and OLS1 bits in the TRDFCR register.		R/W

TOC1	TRDIOC1 initial output level select [Note 2]	R/W
Set to 0. The initial output value is selected by OLS0 and OLS1 bits in the TRDFCR register.		R/W

TOB1	TRDIOB1 initial output level select [Note 2]	R/W
Set to 0. The initial output value is selected by OLS0 and OLS1 bits in the TRDFCR register.		R/W

TOA1	TRDIOA1 initial output level select [Note 2]	R/W
Set to 0. The initial output value is selected by OLS0 and OLS1 bits in the TRDFCR register.		R/W

TOD0	TRDIOD0 initial output level select [Note 2]	R/W
Set to 0. The initial output value is selected by OLS0 and OLS1 bits in the TRDFCR register.		R/W

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	

TOB0	TRDIOB0 initial output level select [Note 2]	R/W
Set to 0. The initial output value is selected by OLS0 and OLS1 bits in the TRDFCR register.		R/W

TOA0	TRDIOA0 initial output level select [Note 2]	R/W
Set to 0. The initial output value is selected by OLS0 and OLS1 bits in the TRDFCR register.		R/W

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

[PWM3 Mode]

Address: F0259H 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/040C2H) 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 RDe Digital Filter Function Select Register i (TRDDFi) (i = 0 or 1)

This register is used to select the digital filter function in input capture function, and to select the forced cutoff control of TRDIOji (j = A, B, C, D) pins in each PWM output modes.

Set the TRDDFi register by a 1-bit or 8-bit memory manipulation instruction.

The specifications of this register differ depending on the operating mode.

Figure 8-12. Format of Timer RDe Digital Filter Function Select Register i (TRDDFi) (i = 0 or 1)

[Input capture function]

Address: F025AH (TRDDF0), F025BH (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/040C2H) 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.

[PWM function, Reset synchronous PWM mode, Complementary PWM mode, PWM3 mode, Extended PWM mode, and Extended complementary PWM mode]

Address: F025AH (TRDDF0), F025BH (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 RDe 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 RDe 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 RDe 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 RDe 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/040C2H) 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 RDe Control Register i (TRDCRi) (i = 0 or 1)

This register is used to select the TRDi count source and the TRDi count clear source.

Set the TRDCRi register by a 1-bit or 8-bit memory manipulation instruction.

The specifications of this register differ depending on the operating mode.

The TRDCR1 register is not used in reset synchronous PWM mode or PWM3 mode.

Figure 8-13. Format of Timer RDe Control Register i (TRDCRi) (i = 0 or 1)

[Input capture function and Output compare function]

Address: F0260H (TRDCR0), F0270H (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 to 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	TRDCLK0 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/040C2H) 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 (TRDCLK0 input) and the STCLK bit is set to 1 (external clock input enabled).

4. As the timer RDe operating clock (f_{TRD}), f_{CLK} is selected when FRQSEL4 = 0 in the user option byte (000C2H/040C2H), (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 8-37**. When selecting the count source for the timer RDe, 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 RDe operating clock (f_{TRD}).

[PWM Function and Extended PWM mode]

Address: F0260H (TRDCR0), F0270H (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	TRDCLK0 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/040C2H) 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 (TRDCLK0 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/040C2H), (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 8-37**. When selecting the count source for the timer RDe, 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).

[Reset Synchronous PWM Mode]

Address: F0260H (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	TRDCLK0 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/040C2H) 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 (TRDCLK0 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/040C2H), (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 8-37**. When selecting the count source for the timer RDe, 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).

Remark The TRDCR1 register is not used in reset synchronous PWM mode.

[Complementary PWM mode and Extended complementary PWM mode]

Address: F0260H (TRDCR0), F0270H (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	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 [Note 3]	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	TRDCLK0 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/040C2H) 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 (TRDCLK0 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/040C2H), (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 8-37**. When selecting the count source for the timer RDe, 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).

[PWM3 Mode]

Address: F0260H (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/040C2H) 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/040C2H), (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 8-37**. When selecting the count source for the timer RDe, 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 The TRDCR1 register is not used in PWM3 mode.

8.2.13 Timer RDe I/O Control Register Ai (TRDIORAi) (i = 0 or 1)

This register is used to select the TRDGRAi and TRDGRBi register function.

Use this register only in input capture function and output compare function.

Set the TRDIORAi register by a 1-bit or 8-bit memory manipulation instruction.

The specifications of this register differ depending on the operating mode.

Figure 8-14. Format of Timer RDe I/O Control Register Ai (TRDIORAi) (i = 0 or 1)

[Input Capture Function]

Address: F0261H (TRDIORA0), F0271H (TRDIORA1) After Reset: 00H [Note 1]

Symbol	7	6	5	4	3	2	1	0
TRDIORAi	0	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
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/040C2H) 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.

[Output Compare Function]

Address: F0261H (TRDIORA0), F0271H (TRDIORA1) After Reset: 00H [Note 1]

Symbol	7	6	5	4	3	2	1	0
TRDIORAi	0	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
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/040C2H) 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 RDe I/O Control Register Ci (TRDIORCi) (i = 0 or 1)

This register is used to select the TRDGRCi and TRDGRDi register function.
Use this register only in input capture function and output compare function.
Set the TRDIORCi register by a 1-bit or 8-bit memory manipulation instruction.
The specifications of this register differ depending on the operating mode.

Figure 8-15. Format of Timer RDe I/O Control Register Ci (TRDIORCi)

[Input Capture Function]

Address: F0262H (TRDIORC0), F0272H (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/040C2H) 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.

[Output Compare Function]

Address: F0262H (TRDIORC0), F0272H (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 (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 (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/040C2H) 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 RDe Status Register i (TRDSRi) (i = 0 or 1)

This register stores each status flag of timer RDe.

This register has different specifications in input capture function of timer mode and other than input cpature function.

The TRDSR0 register is not used in extended complementary PWM mode.

Access the TRDSRi register by a 1-bit or 8-bit memory manipulation instruction.

Nothing is assigned to bits 5 and 6 in the TRDSR0 register.

Figure 8-16. Format of Timer RDe Status Register i (TRDSRi) (i = 0 or 1)

[Input Capture Function]

Address: F0263H (TRDSR0), F0273H (TRDSR1) After Reset: 00H (TRDSR0), 40H (TRDSR1) Note 1

Symbol	7	6	5	4	3	2	1	0
TRDSRi	0	UDS	UDF	OVF	IMFD	IMFC	IMFB	IMFA

Bit 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	R

UDS	Up / down counter status flag Note 2	R/W
Disabled in input capture function.		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] Write 0 after reading. Note 4 [Source for setting to 1] When the TRDi register overflows		R/W

IMFD	Input capture/compare match flag D Note 7	R/W
[Source for setting to 0] Write 0 after reading. Note 4 [Source for setting to 1] 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] Write 0 after reading. Note 4 [Source for setting to 1] 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] Write 0 after reading. Note 4 [Source for setting to 1] 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] Write 0 after reading. Note 4 [Source for setting to 1] 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/040C2H) 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 bits 5 and 6 in the TRDSR0 register. The write value must be 00B for bits 6 to 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 at the same time.

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.

[Functions other than Input capture function]

Address: F0263H (TRDSR0), F0273H (TRDSR1) After Reset: 00H (TRDSR0), 40H (TRDSR1) Note 1

Symbol	7	6	5	4	3	2	1	0
TRDSRi	0	UDS	UDF	OVF	IMFD	IMFC	IMFB	IMFA

Bit 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	R

UDS	Up / down counter status flag Note 2	R/W
	• In extended complementary PWM mode, [Source for setting to 0] TRDi is in decrement counting. [Source for setting to 1] TRDi is in increment counting. • Disabled in other than extended complementary PWM mode. Do not use in complementary PWM mode.	R

UDF	Underflow flag Note 2	R/W
	• In complementary PWM and extended complementary PWM modes, [Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] When TRDi underflows. • Disabled in other than complementary PWM and extended complementary PWM modes.	R/W

OVF	Overflow flag Note 4	R/W
	• In extended complementary PWM modes, [Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] When the value of TRD0 and TRDGRA0 match. Note 7 • In other than extended complementary PWM modes. [Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] When the TRDi register overflows	R/W

IMFD	Input capture/compare match flag D Note 6	R/W
	• Do not use in extended complementary PWM mode. • In other than extended complementary PWM modes. [Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] When the values of TRDi and TRDGRDi match. Note 5	R/W

IMFC	Input capture/compare match flag C Note 6	R/W
	• Do not use in extended complementary PWM mode. • In other than extended complementary PWM modes. [Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] When the values of TRDi and TRDGRCi match. Note 5	R/W

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

IMFB	Input capture/compare match flag B Note 6	R/W
• Do not use in Extended Complementary PWM mode. • In other than extended complementary PWM modes. [Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] When the values of TRDi and TRDGRBi match. Note 5		R/W

IMFA	Input capture/compare match flag A Note 6	R/W
• Do not use in Extended Complementary PWM mode mode. • In other than extended complementary PWM modes. [Source for setting to 0] Write 0 after reading. Note 3 [Source for setting to 1] When the values of TRDi and TRDGRAi match. Note 5		R/W

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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 bits 5 and 6 in the TRDSR0 register. The write value must be 00B for bits 6 to 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 at the same time.

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.

7. Only TRDSR1 register. In extended complementary PWM mode, TRDSR0 register is not used.

8.2.16 Timer RDe Interrupt Enable Register i (TRDIERi) (i = 0 or 1)

This register is used to select the interrupt enable of timer RDe.

Set the TRDIERi register by a 1-bit or 8-bit memory manipulation instruction.

Figure 8-17. Format of Timer RDe Interrupt Enable Register i (TRDIERi)

Address: F0264H (TRDIER0), F0274H (TRDIER1) After Reset: 00H Note 1

Symbol	7	6	5	4	3	2	1	0
TRDIERi	0	0	0	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 Note 2	Overflow/underflow interrupt enable	R/W
0	Interrupt by bits OVF and UDF disabled	R/W
1	Interrupt by bits OVF and UDF enabled	

IMIED Note 2	Input capture/compare match interrupt enable D	R/W
0	Interrupt by the IMFD bit is disabled	R/W
1	Interrupt by the IMFD bit is enabled	

IMIEC Notes 2, 3	Input capture/compare match interrupt enable C	R/W
0	Interrupt by the IMFC bit is disabled	R/W
1	Interrupt by the IMFC bit is enabled	

IMIEB Note 2	Input capture/compare match interrupt enable B	R/W
0	Interrupt by the IMFB bit is disabled	R/W
1	Interrupt by the IMFB bit is enabled	

IMIEA Note 2	Input capture/compare match interrupt enable A	R/W
0	Interrupt by the IMFA bit is disabled	R/W
1	Interrupt by the IMFA bit is enabled	

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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 extended complementary PWM mode does not use this register. See **8.4 Timer RDe Interrupt** and **8.7 Interrupt Decimation Module (TRDMBK)** for details on extended complementary PWM mode interrupts.

3. In extended PWM mode, set this bit to 0.

8.2.17 Timer RDe PWM Function Output Level Control Register i (TRDPOCRi) (i = 0 or 1)

This register is used to select the output level of TRDIOji (j = B, C, D) pins.

Settings to the TRDPOCRi register are enabled only in PWM function. When not in PWM function, they are disabled.

Set the TRDPOCRi register by a 1-bit or 8-bit memory manipulation instruction.

Figure 8-18. Format of Timer RDe PWM Function Output Level Control Register i (TRDPOCRi)

[PWM function, Extended PWM mode]

Address: F0265H (TRDPOCR0), F0275H (TRDPOCR1) After Reset: 00H Note 1

Symbol	7	6	5	4	3	2	1	0
TRDPOCRi	0	0	0	0	0	POLD	POLC	POLB

Bits 7 to 4	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	R

Bit 3	Nothing is assigned	R/W
–	The write value must be 0.	R/W

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 Note 2	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.	

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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 extended PWM mode, set this bit to 0.

8.2.18 Timer RDe Counter i (TRDi) (i = 0 or 1)

This register sets the timer count value or the dead time count value.

Set the TRDi register by a 16-bitt memory manipulation instruction.

The specifications of this register differ depending on the operating mode.

[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]

Access the TRDi register in 16-bit units. Do not access it in 8-bit units.

[Extended PWM Mode]

Access the TRDi register in 16-bit units. Do not access it in 8-bit units.

[Extended Complementary PWM Mode]

Access the TRDi register in 16-bit units. Do not access it in 8-bit units.

Figure 8-19. Format of Timer RDe Counter i (TRDi) (i = 0 or 1)

[Timer mode]

Address: F0266H (TRD0), F0276H (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. 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/040C2H) 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.

[Reset synchronous PWM mode and PWM3 mode]

Address: F0266H (TRD0) After Reset: 0000H [Note]

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRD0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

–	Function	Setting Range	R/W
Bits 15 to 0	Count the count source. Count operation is incremented. 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/040C2H) 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.

Remark TRD1 register is not used in reset synchronous PWM mode and PWM3 mode.

[Complementary PWM mode]

Address: F0266H (TRD0) After Reset: 0000H [Note]

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRD0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

–	Function	Setting Range	R/W
Bits 15 to 0	Dead time must be set. Count the count source. Count operation is incremented or decremented. When an overflow occurs, the OVF bit in the TRDSR0 register is set to 1.	0001H to FFFFH	R/W

Address: F0276H (TRD1) After Reset: 0000H [Note]

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRD1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

–	Function	Setting Range	R/W
Bits 15 to 0	Set to 0000H. Count the count source. Count operation is incremented or decremented. 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/040C2H) 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.

[Extended PWM mode]

Address: F0266H (TRD0), F0276H (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. Count the count source. Count operation is incremented. When the values of TRDi and TRDGRAi match, the IMFA 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/040C2H) 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.

[Extended complementary PWM mode]

Address: F0266H (TRD0) After Reset: 0000H Note 1

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRD0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

–	Function	Setting Range	R/W
Bits 15 to 0	Dead time must be set. Count the count source. Count operation is incremented or decremented. When the values of TRD0 and TRDGRA0 match, the OVF bit in the TRDSR0 register is set to 1.	0002H to 1/4 of the PWM period Note 2	R/W

Address: F0276H (TRD1) After Reset: 0000H Note 1

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRD1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

–	Function	Setting Range	R/W
Bits 15 to 0	Set to 0000H. Count the count source. Count operation is incremented or decremented. When an underflow occurs, the UDF bit in the TRDSR1 register is set to 1.	0000H to FFFFH	R/W

Notes
1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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. Be sure to set the dead time to less than 1/4 of the complementary PWM period. For details of the setting values, refer to **8.3.9 Extended Complementary PWM Mode Table 8-26 Extended Complementary PWM Mode Specifications.**

8.2.19 Timer RDe General Registers ji (TRDGRji) (i = 0 or 1, j = A, B, C, or D)

This register stores the timer value captured in input capture mode and sets the timer comparison value in other modes.
Access the TRDGRji register by a 16-bit memory manipulation instruction.
The specifications of this register differ depending on the operating mode.

[Input Capture Function]

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.
Set the pulse width of the input capture signal applied to the TRDIOji pin to three or more cycles of the timer RDe operating clock (f_{TRD}) when no digital filter is used (j = A, B, C, or D).

[Output Compare Function]

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

[PWM Function]

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

[Reset Synchronous PWM Mode]

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.

[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.
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.
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).

[Extended PWM Mode]

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.
When transferring from buffer registers to general registers, use the reload function. After setting the buffer register, set the RDT bit in the TRDRDTi register to 1. If the RSF bit in the TRDRSFi register is requested to reload at 1, the next value is transferred from buffer registers to general registers. If the RSF bit is requested to reload at 0, the value of buffer register is not transferred.

[Extended Complementary PWM Mode]

Access registers TRDGRAi to TRDGRDi in 16-bit units. Do not access them in 8-bit units.
The value of PWM period cannot be rewritten during the operation.
When transferring from buffer registers to general registers, use the reload function. After setting the buffer register, set the RDT bit in the TRDRDT1 register to 1. If the RSF bit in the TRDRSF1 register is requested to reload at 1, the next value is transferred from buffer registers to general registers. If the RSF bit is requested to reload at 0, the value of buffer register is not transferred.

Figure 8-20. Format of Timer RDe General Registers ji (TRDGRji)

[Input capture function]

Address: F0268H (TRDGRA0), F026AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0278H (TRDGRA1), F027AH (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	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
TRDGRBi																
TRDGRCi																
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/040C2H) 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

[Output compare function]

Address: F0268H (TRDGRA0), F026AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0278H (TRDGRA1), F027AH (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 TRDGRBi TRDGRCi 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/040C2H) 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		TRDIOAi
TRDGRDi			(see **8.3.1 (2) Buffer Operation**.)		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 (set in the general 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. 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

[PWM function]

Address: F0268H (TRDGRA0), F026AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0278H (TRDGRA1), F027AH (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 TRDGRBi TRDGRCi 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/040C2H) 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 (set in the general register (TRDGRAi)) 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

[Reset synchronous PWM mode]

Address: F0268H (TRDGRA0), F026AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0278H (TRDGRA1), F027AH (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 TRDGRBi TRDGRCi 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/040C2H) 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 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 TRDIOC1
TRDGRB1	–	General register. Set the changing point of PWM3 output.	TRDIOB1 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 TRDIOD0
TRDGRC1	TRDBFC1 = 1	Buffer register. Set the changing point of the next PWM2. (see **8.3.1 (2) Buffer Operation**.)	TRDIOA1 TRDIOC1
TRDGRD1	TRDBFD1 = 1	Buffer register. Set the changing point of the next PWM3. (see **8.3.1 (2) Buffer Operation**.)	TRDIOB1 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 (set in the general register (TRDGRA0)) 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

[Complementary PWM Mode]

Address: F0268H (TRDGRA0), F026AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0278H (TRDGRA1), F027AH (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	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
TRDGRBi																
TRDGRCi																
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/040C2H) 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. Setting range: ≥ Value set in TRD0 register ≤ FFFFh - value set in TRD0 register 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. Setting range: ≥ Value set in TRD0 register ≤ Value set in TRDGRA0 register - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	TRDIOB0 TRDIOD0
TRDGRA1	–	General register. Set the changing point of PWM2 output at initialization. Setting range: ≥ Value set in TRD0 register ≤ Value set in TRDGRA0 register - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	TRDIOA1 TRDIOC1
TRDGRB1	–	General register. Set the changing point of PWM3 output at initialization. Setting range: ≥ Value set in TRD0 register ≤ Value set in TRDGRA0 register - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	TRDIOB1 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**.) Setting range: ≥ Value set in TRD0 register ≤ Value set in TRDGRA0 register - value set in TRD0 register Set this register to the same value as the TRDGRB0 register for initialization.	TRDIOB0 TRDIOD0
TRDGRC1	TRDBFC1 = 1	Buffer register. Set the changing point of next PWM2 output. (see **8.3.1 (2) Buffer Operation**.) Setting range: ≥ Value set in TRD0 register ≤ Value set in TRDGRA0 register - value set in TRD0 register Set this register to the same value as the TRDGRA1 register for initialization.	TRDIOA1 TRDIOC1
TRDGRD1	TRDBFD1 = 1	Buffer register. Set the changing point of next PWM3 output. (see **8.3.1 (2) Buffer Operation**.) Setting range: ≥ Value set in TRD0 register ≤ Value set in TRDGRA0 register - value set in TRD0 register Set this register to the same value as the TRDGRB1 register for initialization.	TRDIOB1 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 (set in the general register (TRDGRA0)) 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

[PWM3 Mode]

Address: F0268H (TRDGRA0), F026AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0278H (TRDGRA1), F027AH (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 TRDGRBi TRDGRCi 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/040C2H) 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. Setting range: ≥ Value set in TRDGRA1 register	TRDIOA0
TRDGRA1	–	General register. Set the changing point (active level timing) of PWM output 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. 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 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**.) Setting range: ≥ Value set in TRDGRC1 register	TRDIOA0
TRDGRC1	TRDBFC1 = 1	Buffer register. Set the changing point of next PWM output. (see **8.3.1 (2) Buffer Operation**.) Setting range: ≤ Value set in TRDGRC0 register	–
TRDGRD0	TRDBFD0 = 1	Buffer register. Set the changing point of next PWM output. (see **8.3.1 (2) Buffer Operation**.) 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. (see **8.3.1 (2) Buffer Operation**.) 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 (set in the general register (TRDGRA0)) 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

[Extended PWM Mode]

Address: F0268H (TRDGRA0), F026AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0278H (TRDGRA1), F027AH (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	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
TRDGRBi																
TRDGRCi																
TRDGRDi																

–	Function	R/W
Bits 15 to 0	See **Table 8-9. TRDGRji Register Functions in Extended PWM Mode**.	R/W

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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 8-9. TRDGRji Register Functions in Extended PWM Mode

Register	Register Function	PWM Output Pin
TRDGRAi	General register. Set the PWM period.	–
TRDGRBi	General register. Set the changing point of PWM output.	TRDIOBi
TRDGRCi	Buffer register. Set the next PWM period.	–
TRDGRDi	Buffer register. Set the next changing point of PWM output.	TRDIOBi

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 (set in the general register (TRDGRAi)) 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

[Extended Complementary PWM Mode]

Address: F0268H (TRDGRA0), F026AH (TRDGRB0), FFF58H (TRDGRC0), FFF5AH (TRDGRD0), F0278H (TRDGRA1), F027AH (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	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
TRDGRBi																
TRDGRCi																
TRDGRDi																

–	Function	R/W
Bits 15 to 0	See **Table 8-10. TRDGRji Register Functions in Extended Complementary PWM Mode**.	R/W

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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 8-10. TRDGRji Register Functions in Extended Complementary PWM Mode

Register [Note]	Setting	Register Function	PWM Output Pin
TRDGRA0	–	General register. Set the PWM period at initialization. Setting range: ≥ Value set in TRD0 register ≤ FFFFh - value set in TRD0 register 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	CPSS = 0	General register. Set the changing point of PWM1 output.	TRDIOB0 TRDIOD0
	CPSS = 1	General register. Set the changing point of PWM1 output during counter increment.	
	–	Setting range: > Value set in TRD0 register < Value set in TRDGRA0 register - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	
TRDGRA1	CPSS = 0	General register. Set the changing point of PWM2 output.	TRDIOA1 TRDIOC1
	CPSS = 1	General register. Set the changing point of PWM2 output during counter increment.	
	–	Setting range: > Value set in TRD0 register < Value set in TRDGRA0 register - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	
TRDGRB1	CPSS = 0	General register. Set the changing point of PWM3 output.	TRDIOB1 TRDIOD1
	CPSS = 1	General register. Set the changing point of PWM3 output during counter increment.	
	–	Setting range: > Value set in TRD0 register < Value set in TRDGRA0 register - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	
TRDGRC0	–	Buffer register. Set the PWM period at initialization. Setting range: > Twice the value set in TRD0 register ≤ FFFFh - value set in TRD0 register	(TRDIOC0, output inverted every half period)
TRDGRD0	CPSS = 0	Buffer register. Set the changing point of next PWM1 output. (see **8.3.1 (2) Buffer Operation**.)	TRDIOB0 TRDIOD0
	CPSS = 1	Buffer register. Set the changing point of next PWM1 output during counter increment. (see **8.3.1 (2) Buffer Operation**.)	
	–	Setting range: > Value set in TRD0 register < Value set in TRDGRA0 register - value set in TRD0 register	
TRDGRC1	CPSS = 0	Buffer register. Set the changing point of next PWM2 output. (see **8.3.1 (2) Buffer Operation**.)	TRDIOA1 TRDIOC1
	CPSS = 1	Buffer register. Set the changing point of next PWM2 output during counter increment. (see **8.3.1 (2) Buffer Operation**.)	
	–	Setting range: > Value set in TRD0 register < Value set in TRDGRA0 register - value set in TRD0 register	
TRDGRD1	CPSS = 0	Buffer register. Set the changing point of next PWM3 output. (see **8.3.1 (2) Buffer Operation**.)	TRDIOB1 TRDIOD1
	CPSS = 1	Buffer register. Set the changing point of next PWM3 output during counter increment. (see **8.3.1 (2) Buffer Operation**.)	
	–	Setting range: > Value set in TRD0 register < Value set in TRDGRA0 register - value set in TRD0 register	

Note For asymmetry waveform, aloso set the TRDCMPm register. (see **8.2.20 Timer RDe Extended Compare Register m (TRDCMPm) (m = B0, D0, A1, B1, C1, or D1)**).
Use the TRDGRD0, TRDGRC1, and TRDGRD1 registers to update the changing point of the PWM output. The PWM period cannot be rewritten.

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 (set in the general register (TRDGRA0)) 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.
Be sure to set the general register (except for TRDGRA0) to a value that is larger than Dead time and smaller than TRDGRA0 (PWM cycle) - Dead time, except when setting to output duty0% and duty100%.

Remark i = 0 or 1, j = A, B, C, or D
CPSS: Bit in TRDFCR register

8.2.20 Timer RDe Extended Compare Register m (TRDCMPm) (m = B0, D0, A1, B1, C1, or D1)

This register sets the timer comparison value.

Settings to the TRDCMPm register is enabled only in the extended PWM mode, or the extended complementary PWM mode.

Set the TRDCMPm register by a 16-bit memory manipulation instruction.

[Extended PWM Mode]

Access registers TRDCMPm in 16-bit units. Do not access them in 8-bit units.

When transferring from buffer registers to general registers, use the reload function. After setting the buffer register, set the RDT bit in the TRDRDTi register to 1. If the RSF bit in the TRDRSFi register is requested to reload at 1, the next value is transferred from buffer registers to compare registers. If the RSF bit is requested to reload at 0, the value of buffer register is not transferred.

[Extended Complementary PWM Mode]

Access registers TRDCMPm in 16-bit units. Do not access them in 8-bit units.

When transferring from buffer registers to general registers, use the reload function. After setting the buffer register, set the RDT bit in the TRDRDT1 register to 1. If the RSF bit in the TRDRSF1 register is requested to reload at 1, the next value is transferred from buffer registers to compare registers. If the RSF bit is requested to reload at 0, the value of buffer register is not transferred.

Figure 8-21. Format of Timer RDe Extended Compare Register m (TRDCMPm)

[Extended PWM mode]

Address: F0280H (TRDCMPB0), FFF60H (TRDCMPD0), F0284H (TRDCMPA1), FFF62H (TRDCMPC1), F0288H (TRDCMPB1), FFF64H (TRDCMPD1)

After Reset: FFFFH [Note]

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDCMPm	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

–	Function	R/W
Bits 15 to 0	See **Table 8-11. TRDCMPm Register Functions in Extended PWM Mode**.	R/W

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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 8-11. TRDCMPm Register Functions in Extended PWM Mode

Register	Register Function	PWM Output Pin
TRDCMPB0	General register. Set the changing point of PWM output.	TRDIOD0
TRDCMPD0	Buffer register. Set the next changing point of PWM output.	
TRDCMPA1	Not used in extended PWM mode.	–
TRDCMPC1		
TRDCMPB1	General register. Set the changing point of PWM output.	TRDIOD1
TRDCMPD1	Buffer register. Set the next changing point of PWM output.	

[Extended Complementary PWM Mode]

Address: F0280H (TRDCMPB0), FFF60H (TRDCMPD0), F0284H (TRDCMPA1), FFF62H (TRDCMPC1), F0288H (TRDCMPB1), FFF64H (TRDCMPD1)

After Reset: FFFFH Note

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDCMPm	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

–	Function	R/W
Bits 15 to 0	See **Table 8-12. TRDCMPm Register Functions in Extended Complementary PWM Mode**.	R/W

Table 8-12. TRDCMPm Register Functions in Extended Complementary PWM Mode

Register	Setting	Register Function	PWM Output Pin
TRDCMPB0	CPSS = 1	General register. Set the changing point of PWM1 output during counter decrement. Setting range: > Value set in TRD0 register < Value set in TRDGRA0 register - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	TRDIOB0 TRDIOD0
TRDCMPD0	CPSS = 1	Buffer register. Set the changing point of next PWM1 output during counter decrement. Setting range: > Value set in TRD0 register < Value set in TRDGRA0 register - value set in TRD0 register	
TRDCMPA1	CPSS = 1	General register. Set the changing point of PWM2 output during counter decrement. Setting range: > Value set in TRD0 register < Value set in TRDGRA0 register - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	TRDIOA1 TRDIOC1
TRDCMPC1	CPSS = 1	Buffer register. Set the changing point of next PWM2 output during counter decrement. Setting range: > Value set in TRD0 register < Value set in TRDGRA0 register - value set in TRD0 register	
TRDCMPB1	CPSS = 1	General register. Set the changing point of PWM3 output during counter decrement. Setting range: > Value set in TRD0 register < Value set in TRDGRA0 register - value set in TRD0 register Do not write to this register when bits TSTART0 and TSTART1 in the TRDSTR register are set to 1 (count starts).	TRDIOB1 TRDIOD1
TRDCMPD1	CPSS = 1	Buffer register. Set the changing point of next PWM3 output during counter decrement. Setting range: > Value set in TRD0 register < Value set in TRDGRA0 register - value set in TRD0 register	

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

Caution **In case of CPSS = 0, it is not possible to write to TRDCMPm registers. To output the symmetry waveform with CPSS = 1, set TRDCMPm register to the same value as TRDGRji registers.**
Be sure to set the compare register to a value that is larger than Dead time and smaller than TRDGRA0 (PWM cycle) - Dead time, except when setting to output duty0% and duty100%.

Remark i = 0 or 1, j = A, B, C, or D
CPSS: Bit in TRDFCR register

8.2.21 Timer RDe A/D Trigger Compare Register 0 (TRDADTC0)

This register sets the value of A/D trigger 0 timing.

Settings to the TRDADTC0 register is enabled only in the extended complementary PWM mode.

Set the TRDADTC0 register by a 16-bit memory manipulation instruction.

TRDADTC0 register is enabled when the ADE0 bit in the TRDADCR register is 1. Use the TRDADTB0 register to rewrite this register after the count starts.

Figure 8-22. Format of Timer RDe A/D Trigger Compare Register 0 (TRDADTC0)

[Extended complementary PWM mode]

Address: F028CH (TRDADTC0) After Reset: FFFFH Note 1

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDADTC0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

–	Function	Setting Range	R/W
Bits 15 to 0	Set the value of A/D trigger 0 timing.	0000H ≤ (TRDGRA0 – TRD0 setting value) Note 2	R/W

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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. Only if the ADMD0 bit in the TRDADCR register is 0. If the ADMD0 bit is 1, it will be "0000H ≤ (TRDGRA0 − TRD0 setting value + 1)".

8.2.22 Timer RDe A/D Trigger Buffer Register 0 (TRDADTB0)

This register is a buffer register of the TRDADTC0 register. Sets the value of next A/D trigger 0 timing.

Settings to the TRDADTB0 register is enabled only in the extended complementary PWM mode.

Set the TRDADTB0 register by a 16-bit memory manipulation instruction.

When transferring from buffer registers to general registers, use the reload function. Set the RDT bit in the TRDRDT1 register to 1. If the RSF bit in the TRDRSF1 register is requested to reload at 1, the next value is transferred from buffer registers to compare registers. If the RSF bit is at 0, the value of buffer register is not transferred.

Figure 8-23. Format of Timer RDe A/D Trigger Buffer Register 0 (TRDADTB0)

[Extended complementary PWM mode]

Address: FFF66H (TRDADTB0) After Reset: FFFFH Note 1

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDADTB0	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

–	Function	Setting Range	R/W
Bits 15 to 0	Set the value of next A/D trigger 0 timing.	0000H ≤ (TRDGRA0 – TRD0 setting value) Note 2	R/W

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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. Only if the ADMD0 bit in the TRDADCR register is 0. If the ADMD0 bit is 1, it will be "0000H ≤ (TRDGRA0 − TRD0 setting value + 1)".
When generating an A/D trigger with the compare match between TRD0 and TRDGRA0 registers, set the value of "(TRDGRA0 − TRD0 setting value)" after setting the ADMD0 bit to 0. When generating an A/D trigger with the underflow of TRD1 register, set 0000H after setting the ADMD0 bit to 1.

8.2.23 Timer RDe A/D Trigger Compare Register 1 (TRDADTC1)

This register sets the value of A/D trigger 1 timing.

Settings to the TRDADTC1 register is enabled only in the extended complementary PWM mode.

Set the TRDADTC1 register by a 16-bit memory manipulation instruction.

TRDADTC1 register is enabled when the ADE1 bit in the TRDADCR register is 1. Use the TRDADTB1 register to rewrite this register after the count starts.

Figure 8-24. Format of Timer RDe A/D Trigger Compare Register 1 (TRDADTC1)

[Extended complementary PWM mode]

Address: F0290H (TRDADTC1) After Reset: FFFFH Note 1

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDADTC1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

–	Function	Setting Range	R/W
Bits 15 to 0	Set the value of A/D trigger 1 timing.	0000H ≤ (TRDGRA0 – TRD0 setting value) Note 2	R/W

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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. Only if the ADMD1 bit in the TRDADCR register is 0. If the ADMD1 bit is 1, it will be "0000H ≤ (TRDGRA0 − TRD0 setting value + 1)".

8.2.24 Timer RDe A/D Trigger Buffer Register 1 (TRDADTB1)

This register is a buffer register of the TRDADTC1 register. Sets the value of next A/D trigger 1 timing.

Settings to the TRDADTB1 register is enabled only in the extended complementary PWM mode.

Set the TRDADTB1 register by a 16-bit memory manipulation instruction.

When transferring from buffer registers to general registers, use the reload function. Set the RDT bit in the TRDRDT1 register to 1. If the RSF bit in the TRDRSF1 register is requested to reload at 1, the next value is transferred from buffer registers to compare registers. If the RSF bit is at 0, the value of buffer register is not transferred.

Figure 8-25. Timer RDe A/D Trigger Buffer Register 1 (TRDADTB1)

[Extended complementary PWM mode]

Address: FFF68H (TRDADTB1) After Reset: FFFFH Note 1

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRDADTB1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

–	Function	Setting Range	R/W
Bits 15 to 0	Set the value of next A/D trigger 1 timing.	0000H ≤ (TRDGRA0 − TRD0 setting value) Note 2	R/W

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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. Only if the ADMD1 bit in the TRDADCR register is 0. If the ADMD1 bit is 1, it will be "0000H ≤ (TRDGRA0 − TRD0 setting value + 1)".
When generating an A/D trigger with the compare match between TRD0 and TRDGRA0 registers, set the value of "(TRDGRA0 − TRD0 setting value)" after setting the ADMD1 bit to 0. When generating an A/D trigger with the underflow of TRD1 register, set 0000H after setting the ADMD1 bit to 1.

8.2.25 Timer RDe Reload Trigger Register (TRDRDT01)

This register is used when requesting a register reload from buffer registers to general registers.

Settings to the TRDRDT0 register (Lower 8-bit) is enabled only in the extended PWM mode.

Settings to the TRDRDT1 register (Higher 8-bit) is enabled only in the extended PWM mode, or the extended complementary PWM mode.

Set the TRDRDT01 register by a 16-bit memory manipulation instruction.

Set the TRDRDT0 (lower 8-bit) and TRDRDT1 (higher 8-bit) registers by a 1-bit, or 8-bit memory manipulation instruction.

Figure 8-26. Format of Timer RDe Reload Trigger Register (TRDRDT01)

Address: FFF6BH After Reset: 00H [Note]

Symbol	15	14	13	12	11	10	9	8
TRDRDT1	0	0	0	0	0	0	0	RDT

Address: FFF6AH After Reset: 00H [Note]

Symbol	7	6	5	4	3	2	1	0
TRDRDT0	0	0	0	0	0	0	0	RDT

●Timer RDe Reload Trigger Register 0 (TRDRDT0)

[Extended PWM mode]

Address: FFF6AH After Reset: 00H [Note]

Symbol	7	6	5	4	3	2	1	0
TRDRDT0	0	0	0	0	0	0	0	RDT

Bits 7 to 1	Nothing is assigned	R/W
–	The write value must be 0.	–

RDT	Reload Function Control	R/W
0	Writing 0 is invalid.	R/W
1	TRDGRA0, TRDGRB0, TRDCMPB0 are reloaded.	

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

- **Timer RDe Reload Trigger Register 1 (TRDRDT1)**

Figure 8-27. Format of Timer RDe Reload Trigger Register 1 (TRDRDT1)

[Extended PWM mode]

Address: FFF6BH After Reset: 00H [Note]

Symbol	7	6	5	4	3	2	1	0
TRDRDT1	0	0	0	0	0	0	0	RDT

Bits 7 to 1	Nothing is assigned	R/W
–	The write value must be 0.	–

RDT	Reload Function Control	R/W
0	Writing 0 is invalid.	R/W
1	TRDGRA1, TRDGRB1, TRDCMPB1 are reloaded.	

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

[Extended complementary PWM mode]

Address: FFF6BH After Reset: 00H [Note]

Symbol	7	6	5	4	3	2	1	0
TRDRDT1	0	0	0	0	0	0	0	RDT

Bits 7 to 1	Nothing is assigned	R/W
–	The write value must be 0.	–

RDT	Reload Function Control	R/W
0	Writing 0 is invalid.	R/W
1	All general registers are reloaded.	

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

8.2.26 Timer RDe Reload Status Flag Register (TRDRSF01)

This register shows the status of the register reload function.

The TRDSRF0 register (Lower 8-bit) is enabled only in the extended PWM mode.

The TRDRSF1 register (Higher 8-bit) is enabled only in the extended PWM mode, or the extended complementary PWM mode.

Access the TRDRSF01 register by a 16-bit memory manipulation instruction.

Access the TRDRSF0 (lower 8-bit) and TRDRSF1 (higher 8-bit) registers by a 1-bit, or 8-bit memory manipulation instruction.

Figure 8-28. Format of Timer RDe Reload Status Flag Register (TRDRSF01)

Address: F0297H After Reset: 00H Note

Symbol	7	6	5	4	3	2	1	0
TRDRSF1	0	0	0	0	0	0	0	RSF

Address: F0296H After Reset: 00H Note

Symbol	7	6	5	4	3	2	1	0
TRDRSF0	0	0	0	0	0	0	0	RSF

- **Timer RDe Reload Status Flag Register (TRDRSF0)**

[Extended PWM mode]

Address: F0296H After Reset: 00H Note

Symbol	7	6	5	4	3	2	1	0
TRDRSF0	0	0	0	0	0	0	0	RSF

Bits 7 to 1	Nothing is assigned	R/W
–	These bits are setting prohibited.	–

RSF	Reload Trigger Status Flag	R/W
0	Reload trigger enable status.	R
1	Waiting for reload trigger status.	
[Source for setting to 0] During reload trigger operation or when counting operation starts. [Source for setting to 1] When RDT bit of TRDRDT0 register is set to 1.		

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

- **Timer RDe Reload Status Flag Register 1 (TRDRSF1)**

Figure 8-29. Format of Timer RDe Reload Status Flag Register 1 (TRDRSF1)

[Extended PWM mode, Extended complementary PWM mode]

Address: F0297H After Reset: 00H [Note]

Symbol	7	6	5	4	3	2	1	0
TRDRSF1	0	0	0	0	0	0	0	RSF

Bits 7 to 1	Nothing is assigned	R/W
–	These bits are setting prohibited.	–

RSF	Reload Trigger Status Flag	R/W
0	Reload trigger enable status.	R
1	Waiting for reload trigger status.	
[Source for setting to 0] During reload trigger operation or when counting operation starts. [Source for setting to 1] When RDT bit of TRDRDT1 register is set to 1.		

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

8.2.27 Timer RDe A/D Trigger Control Register (TRDADCR)

This register is used to select control for A/D trigger signals.

Settings to the TRDADCR register is enabled only in the extended complementary PWM mode.

Set the TRDADCR register by a 1-bit or 8-bit memory manipulation instruction.

Figure 8-30. Format of Timer RDe A/D Trigger Control Register (TRDADCR)

[Extended complementary PWM mode]

Address: F0298H After Reset: 00H [Note]

Symbol	7	6	5	4	3	2	1	0
TRDADCR	0	0	ADMD1	ADE1	0	0	ADMD0	ADE0

Bits 7, 6	Nothing is assigned	R/W
–	The write value must be 0.	–

ADMD1	A/D Trigger Mode Selection 1	R/W
0	A/D trigger 1 compares during up counting.	R/W
1	A/D trigger 1 compares during down counting.	

ADE1	A/D Trigger Enable 1	R/W
0	A/D trigger 1 is disabled.	R/W
1	A/D trigger 1 is enabled.	

Bits 3, 2	Nothing is assigned	R/W
–	These bits are setting prohibited.	–

ADMD0	A/D Trigger Mode Selection 0	R/W
0	A/D trigger 0 compares during up counting.	R/W
1	A/D trigger 0 compares during down counting.	

ADE0	A/D Trigger Enable 0	R/W
0	A/D trigger 0 is disabled.	R/W
1	A/D trigger 0 is enabled.	

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

8.2.28 Timer RDe Extended PWM Mode Register i (TRDEMRi) (i = 0 or 1)

This register is used to select control for extended PWM mode.

Settings to the TRDEMRi register is enabled only in the extended PWM mode.

Set this register when bits TSTART0 and TSTART1 in the TRDSTR register are 0.

Set the TRDEMRi register by a 1-bit or 8-bit memory manipulation instruction.

Figure 8-31. Format of Timer RDe Extended PWM Mode Register i (TRDEMRi)

[Extended PWM mode]

Address: F029AH (TRDEMR0), F029BH (TRDEMR1) After Reset: 00H [Note 1]

Symbol	7	6	5	4	3	2	1	0
TRDEMRi	DTEND	GTEND	DTENB	GTENB	0	0	CCLV1	CCLV0

DTEND	TRDIODi Dithering Function Enable	R/W
0	TRDIODi dithering function is disabled.	R/W
1	TRDIODi dithering function is enabled.	

GTEND	TRDIODi Gate Function Enable	R/W
0	TRDIODi gate function is disabled.	R/W
1	TRDIODi gate function is enabled.	

DTENB	TRDIOBi Dithering Function Enable	R/W
0	TRDIOBi dithering function is disabled.	R/W
1	TRDIOBi dithering function is enabled.	

GTENB	TRDIOBi Gate Function Enable	R/W
0	TRDIOBi gate function is disabled.	R/W
1	TRDIOBi gate function is enabled.	

Bits 3, 2	Nothing is assigned	R/W
–	The write value must be 0.	–

CCLV1	CCLV0	TRDi Counter Restart Control [Note 2]	R/W
0	0	Counter restart function is disabled.	R/W
0	1	Counter restart at the falling edge of TRDiRES input.	
1	0	Counter restart at the rising edge of TRDiRES input.	
1	1	Counter restart at both edge of TRDiRES input.	

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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 case of synchronous operation (TRDSYNC bit in the TRDMR register is 1), since only TRD0RES input signal is valid, the CCLV1 and CCLV0 bit settings in TRDEMR0 register are valid, the CCLV1 and CCLV0 bit settings in TRDEMR1 register are invalid.

8.2.29 Timer RDe Dithering Number Register i (TRDDNRi) (i = 0 or 1)

This register is used to select control for dithering function in the extended PWM mode.

Settings to the TRDDNRi register is enabled only in the extended PWM mode.

Set the TRDDNRi register by a 1-bit or 8-bit memory manipulation instruction.

TRDDNRi and TRDGPRi registers can be combined and accessed as 16-bit length register (TRDDGCRi).

Figure 8-32. Format of Timer RDe Dithering Number Register i (TRDDNRi)

[Extended PWM mode]

Address: FFF6CH (TRDDNR0), FFF6EH (TRDDNR1) After Reset: 00H Note 1

Symbol	7	6	5	4	3	2	1	0
TRDDNRi	0	0	0	0	DNR[3:0]			

Bits 7 to 4	Nothing is assigned	R/W
–	The write value must be 0.	–

DNR[3:0]	Dithering Number Setting Note 2	R/W
0000B	0/16 time	R/W
0001B	1/16 time	
0010B	2/16 times	
:	:	
1110B	14/16 times	
1111B	15/16 times	

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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. To rewrite these bits during counting, set the RDT bit in the TRDRDTi register to 1 after setting the value to DNR[3:0] bits.

Caution Set the DNR[3:0] bit to 0000B, and then set the TSTARTi bit in the TRDSTR register from 0 to 1 (count starts). To rewrite this register during counting, set this register and then set RDT bit in the TRDRDTi register to 1.

8.2.30 Timer RDe Gate Pattern Register i (TRDGPRi) (i = 0 or 1)

This register is used to select control for gate function in the extended PWM mode.

Settings to the TRDGPRi register is enabled only in the extended PWM mode.

Set the TRDGPRi register by a 1-bit or 8-bit memory manipulation instruction.

TRDDNRi and TRDGPRi registers can be combined and accessed as 16-bit length register (TRDDGCRi).

Figure 8-33. Format of Timer RDe Gate Pattern Register i (TRDGPRi)

[Extended PWM mode]

Address: FFF6DH (TRDGPR0), FFF6FH (TRDGPR1) After Reset: 00H [Note 1]

Symbol	7	6	5	4	3	2	1	0
TRDGPRi	GPAT	0	0	0	GPR[3:0]			

GPAT	Gate Pattern Selection	R/W
0	Pattern.1 is selected.	R/W
1	Pattern.2 is selected.	

Bits 6 to 4	Nothing is assigned	R/W
–	The write value must be 0.	–

GPR[3:0]	Gate Pattern Setting [Note 2]	R/W
0000B	[Pattern.1] 0/16 time, [Pattern.2] 15/16 times	R/W
0001B	[Pattern.1] 1/16 time, [Pattern.2] 14/16 times	
0010B	[Pattern.1] 2/16 times, [Pattern.2] 13/16 times	
:	:	
1110B	[Pattern.1] 14/16 times, [Pattern.2] 1/16 time	
1111B	[Pattern.1] 15/16 times, [Pattern.2] 0/16 time	

Notes 1. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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. To rewrite these bits during counting, set the RDT bit in the TRDRDTi register to 1 after setting the value to GPR[3:0] bits.

Caution To rewrite this register during counting, set this register and then set RDT bit in the TRDRDTi register to 1.

8.2.31 Timer RDe Dithering / Gate Control Register i (TRDDGCRi) (i = 0 or 1)

TRDDNRi and TRDGPRi registers can be combined and accessed as 16-bit length register (TRDDGCRi).

This register is used to select control of dithering and gate function in the extended PWM mode.

Settings to the TRDDGCRi register is enabled only in the extended PWM mode.

To rewrite GPR[3:0] bits or DNR[3:0] bits during counting, set this register and then set the RDT bit in the TRDRDTi register to 1.

Set the TRDDGCRi register by a 16-bit memory manipulation instruction.

Figure 8-34. Format of Timer RDe Dithering / Gate Control Register i (TRDDGCRi)

[Extended PWM mode]

Address: FFF6CH (TRDDGCR0), FFF6EH (TRDDGCR1) After Reset: 0000H Note

Symbol	15	14	13	12	11	10	9	8
TRDGPRi	GPAT	0	0	0	GPR[3:0]			

Symbol	7	6	5	4	3	2	1	0
TRDDNRi	0	0	0	0	DNR[3:0]			

GPAT	Gate Pattern Selection	R/W
0	Pattern.1 is selected.	R/W
1	Pattern.2 is selected.	

Bits 14 to 12	Nothing is assigned	R/W
–	The write value must be 0.	–

GPR[3:0]	Gate Pattern Setting	R/W
0000B	[Pattern.1] 0/16 time, [Pattern.2] 15/16 times	R/W
0001B	[Pattern.1] 1/16 time, [Pattern.2] 14/16 times	
0010B	[Pattern.1] 2/16 times, [Pattern.2] 13/16 times	
:	:	
1110B	[Pattern.1] 14/16 times, [Pattern.2] 1/16 time	
1111B	[Pattern.1] 15/16 times, [Pattern.2] 0/16 time	

Bits 7 to 4	Nothing is assigned	R/W
–	The write value must be 0.	–

DNR[3:0]	Dithering Number Setting	R/W
0000B	0/16 time	R/W
0001B	1/16 time	
0010B	2/16 times	
:	:	
1110B	14/16 times	
1111B	15/16 times	

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

Caution Set the DNR[3:0] bit to 0000B, and then set the TSTARTi bit in the TRDSTR register from 0 to 1 (count starts). To rewrite this register during counting, set this register and then set RDT bit in the TRDRDTi register to 1.

8.2.32 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.

Figure 8-35. Format of PWM Output Delay Control Register 0 (PWMDLY0)

Address: F0229H After Reset: 00H R/W

	15	14	13	12	11	10	9	8
PWMDLY0	TRDD11 Note	TRDD10 Note	TRDC11 Note	TRDC10 Note	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 8-13. 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 RDe operating clock (f_{TRD}).
1	0	TRDIOj1 signal delayed by two cycles of the timer RDe operating clock (f_{TRD}).
1	1	TRDIOj1 signal delayed by three cycles of the timer RDe operating clock (f_{TRD}).

j = A, B, C, or D

Table 8-14. 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 RDe operating clock (f_{TRD}).
1	0	TRDIOj0 signal delayed by two cycles of the timer RDe operating clock (f_{TRD}).
1	1	TRDIOj0 signal delayed by three cycles of the timer RDe 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-bit. 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 RDe 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 (i = 0, 1, j = A, B, C, D).**
7. **When using reset synchronous PWM mode, complementary PWM mode, PWM3 mode, or extended complementary PWM mode, and when using PWM option unit A (PWMOPA), the TRDji[1,0] bits for the target pin should be set to 00B (no delay) (i = 0, 1, j = A, B C, D).**

8.2.33 Port Mode Registers (PM1, PM3, PM4, PM12, PM14)

These registers set input/output of port in 1-bit units. PM1, PM3, PM4, PM12 and PM14 are used in timer RDe.

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.

The PM1, PM3, PM4, PM12 and PM14 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to FFH.

Figure 8-36. Format of Port Mode Registers (PM1, PM3, PM4, PM12, PM14)

Address: FFF21H After Reset: FFH R/W

	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After Reset: FFH R/W

	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	PM32	PM31	PM30

Address: FFF24H After Reset: FFH R/W

	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

Address: FFF2CH After Reset: FFH R/W

	7	6	5	4	3	2	1	0
PM12	PM127	PM126	PM125	1	1	1	1	PM120

Address: FFF2EH After Reset: FFH R/W

	7	6	5	4	3	2	1	0
PM14	1	1	1	1	1	1	1	PM140

PMmn	Pmn pin I/O mode selection (m = 1, 3, 4, 12, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

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 8-15. 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/040C2H) 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. 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/040C2H) is set to 1, and the TRD_CKSEL bit in the clock select register (CKSEL) is cleared to 0. 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. A 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. 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. 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). Bits TCK2 to TCK0 in the TRDCRi register are set to 101B. The active edge is selected by bits CKEG1 and CKEG0 in the TRDCRi register. The PMmn bit in the PMm register of the port used as a TRDCLK0 pin is set to 1 (input mode).

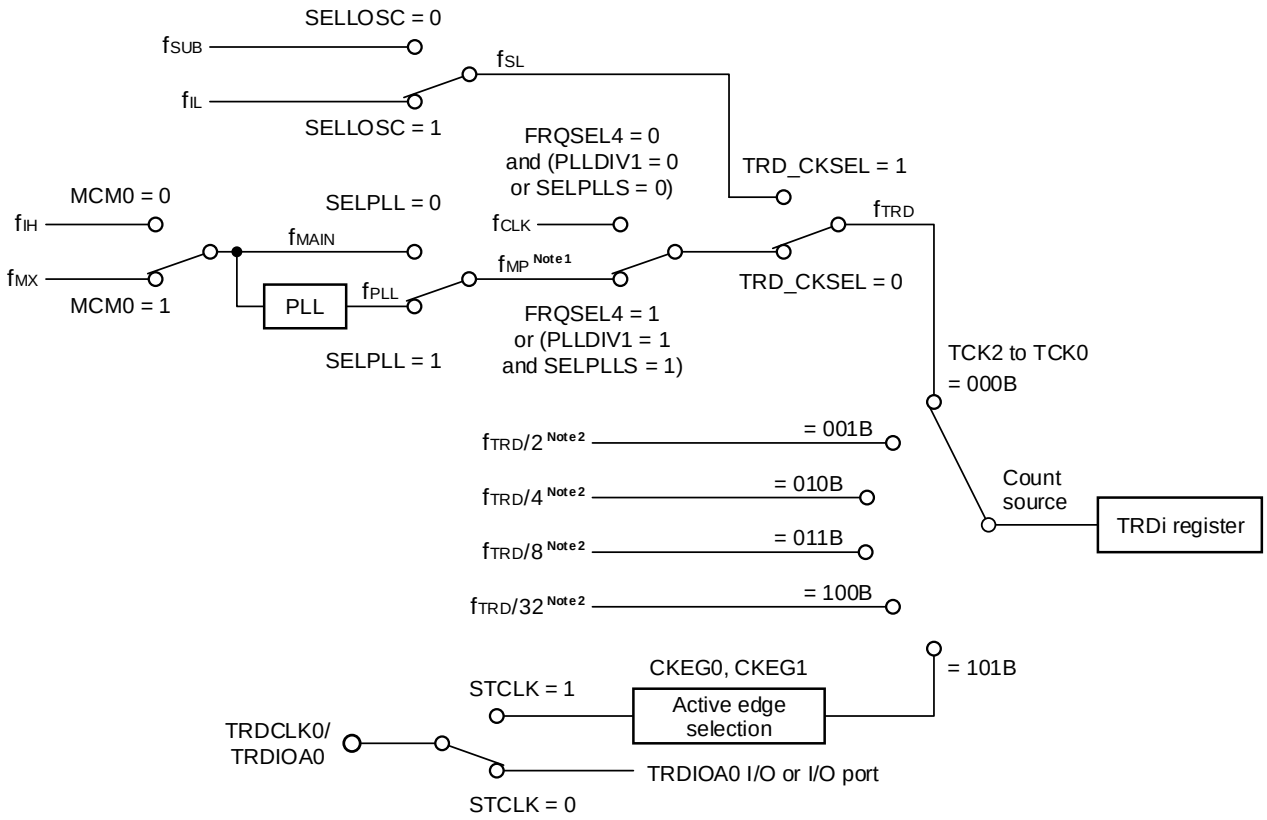
Remark i = 0 or 1

Notes on count source setting is described below.

Notes on Count Source Setting of Timer RDe

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/040C2H) to 0 ($f_{IH} \leq 40$ MHz), and the PLLDIV1 bit in the PLLCTL register to 0 ($f_{PLL} \leq 40$ MHz). Do not set $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, or $f_{CLK}/32$ when FRQSEL4 = 1. The count sources cannot be used when SNOOZE status is output.
f_{IH}, f_{PLL}	When f_{IH}/f_{PLL} (over 40 MHz) is used, set the CSS bit in the CKC register to 0 ($f_{CLK} = f_{MP}$ is selected). When f_{IH}/f_{PLL} (over 40 MHz) is used, set bits MDIV2 to MDIV0 in the MDIV register to 001B ($f_{MP}/2$ is selected). When f_{IH}/f_{PLL} (over 40 MHz) is used, set the TRD_CKSEL bit in the CKSEL register to 0 (f_{CLK}/f_{MP} is selected). When f_{IH}/f_{PLL} (over 40 MHz) is used, set the CSS bit, MDIV2 to MDIV0 bits, and TRD_CKSEL bit before setting the TRD0EN bit in the PER1 register. 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). Setting the FRQSEL4 bit in the user option byte (000C2H/040C2H) to 1 (f_{IH} = 80 MHz/64 MHz) and the PLLDIV bit in the PLLCTL register to 1 (f_{PLL} > 40 MHz) is prohibited. The count sources cannot be used when SNOOZE status is output.
f_{SUB}, f_{IL}	When the CPU accesses the timer RDe register, set the CSS bit in the CKC register to 1 ($f_{CLK} = f_{SL}$ is selected). 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). The count sources should be set when SNOOZE status is output.

Figure 8-37. Count Source Block Diagram



Notes **1.** f_{IH} can be selected even when it is 80 MHz or 64 MHz. f_{PLL} can be selected when it is over 40 MHz.

2. With this setting, select f_{CLK} as the timer RDe operating clock (f_{TRD}).

Remarks i = 0 or 1

TCK0 to TCK2, CKEG0, CKEG1:	Bits in TRDCRi register
STCLK:	Bit in TRDFCR register
FRQSEL4:	Bit in user option byte (000C2H/040C2H)
MCM0:	Bit in CKC register
SELPLL, and PLLDIV1:	Bits in PLLCTL register
SELPLLS:	Bit in PLLSTS register
SELLOSC, and TRD_CKSEL:	Bits in CKSEL register

Set the pulse width of the external clock applied to the TRDCLK0 pin to three or more cycles of the timer RDe 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. See **Table 8-16. Buffer Operation in Each Mode**.

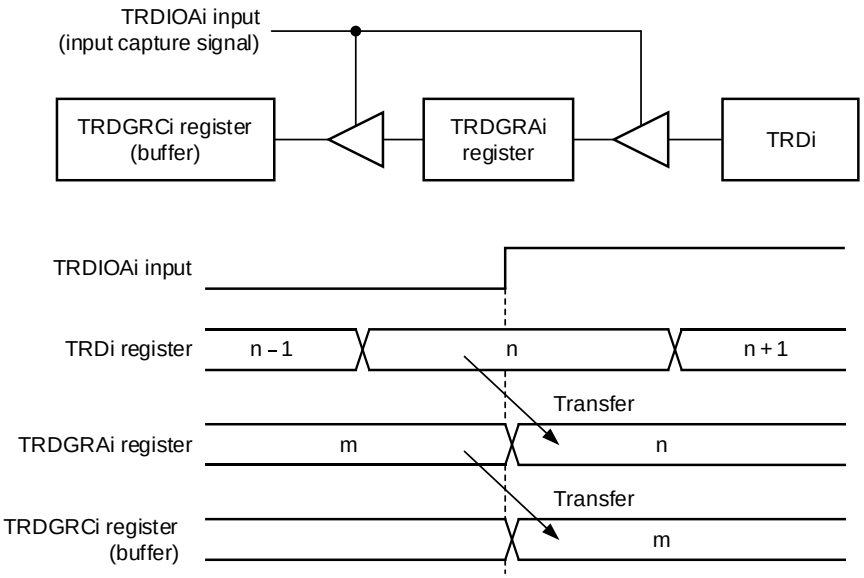
Table 8-16. Buffer Operation in Each Mode

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 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
Extended PWM mode		Compare match with TRDi register and TRDGRAi register, after setting the RDT bit in the TRDRDTi register to 1	Transfer content of buffer register to TRDGRAi, TRDGRBi, and TRDCMPBi registers
Extended complementary PWM mode		TRD1 register underflow, after setting the RDT bit in the TRDRDT1 register to 1 (Dead-time ahead of schedule)	Transfer content of buffer register to TRDGRB0, TRDGRA1, TRDGRB1 registers
		TRD1 register underflow, after setting the RDT bit in the TRDRDT1 register to 1	Transfer content of buffer register to TRDCMPB0, TRDCMPA1, TRDCMPB1 registers Note
		TRD1 register underflow, after setting the RDT bit in the TRDRDT1 register to 1	Transfer content of buffer register to TRDADTC0 and TRDADTC1 registers

Note Only when the CPSS bit in the TRDFCR register is 1.

Remark i = 0 or 1

Figure 8-38. Buffer Operation in Input Capture Function

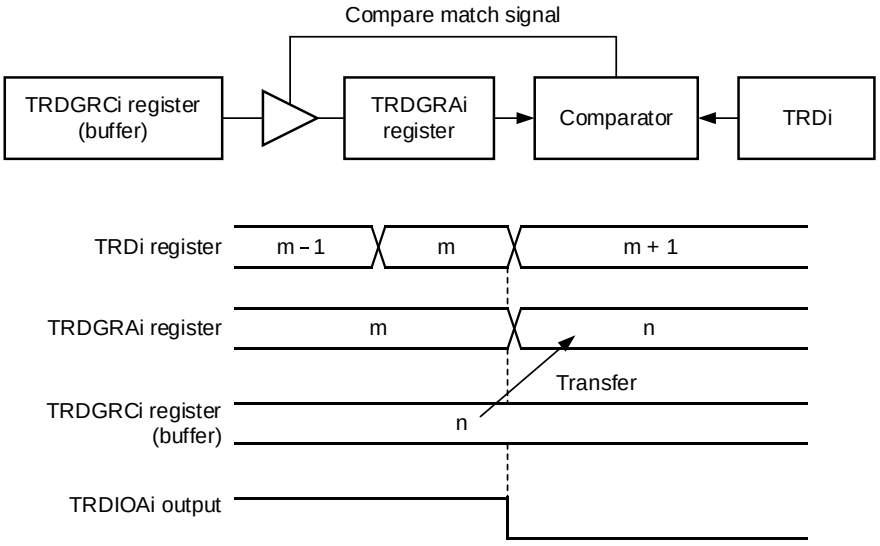


Remark i = 0 or 1

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 8-39. 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.

Extended PWM mode and extended complementary PWM mode support buffer operation with the register reload function. (see **8.3.1 (7) Register Reload Function**)

(3) Synchronous Operation

The TRD1 register is synchronized with the TRD0 register.

- Synchronous preset

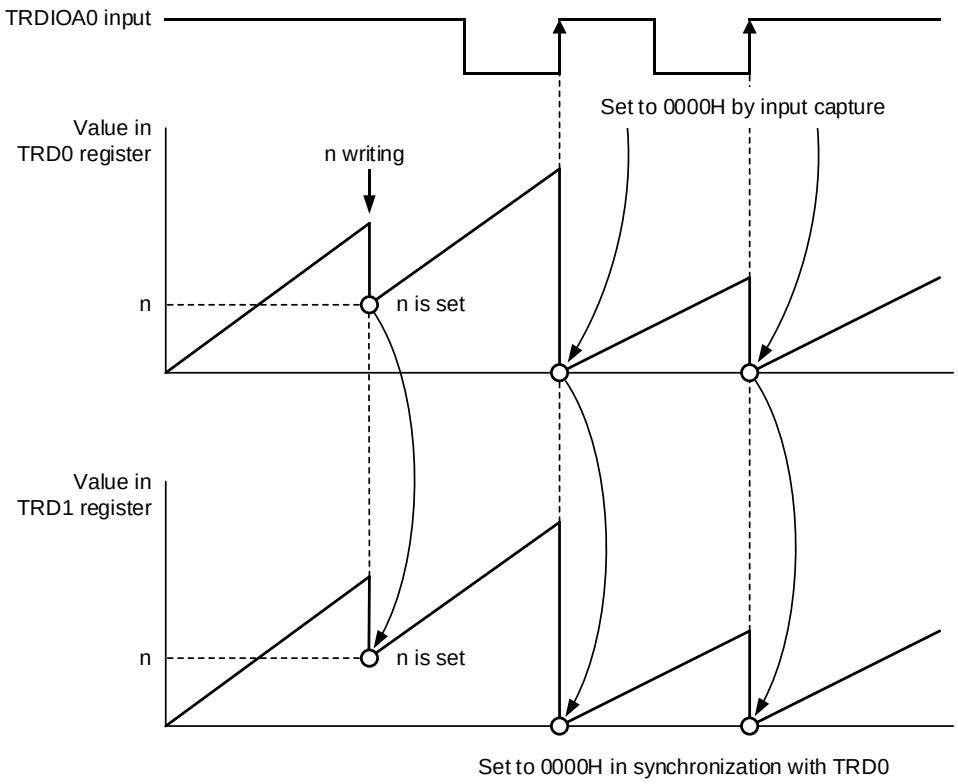
  When the TRDSYNC bit in the TRDMR register is set to 1 (synchronous operation), the data is written to both the TRD0 and TRD1 registers after writing to the TRDi register.

- 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 8-40. 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

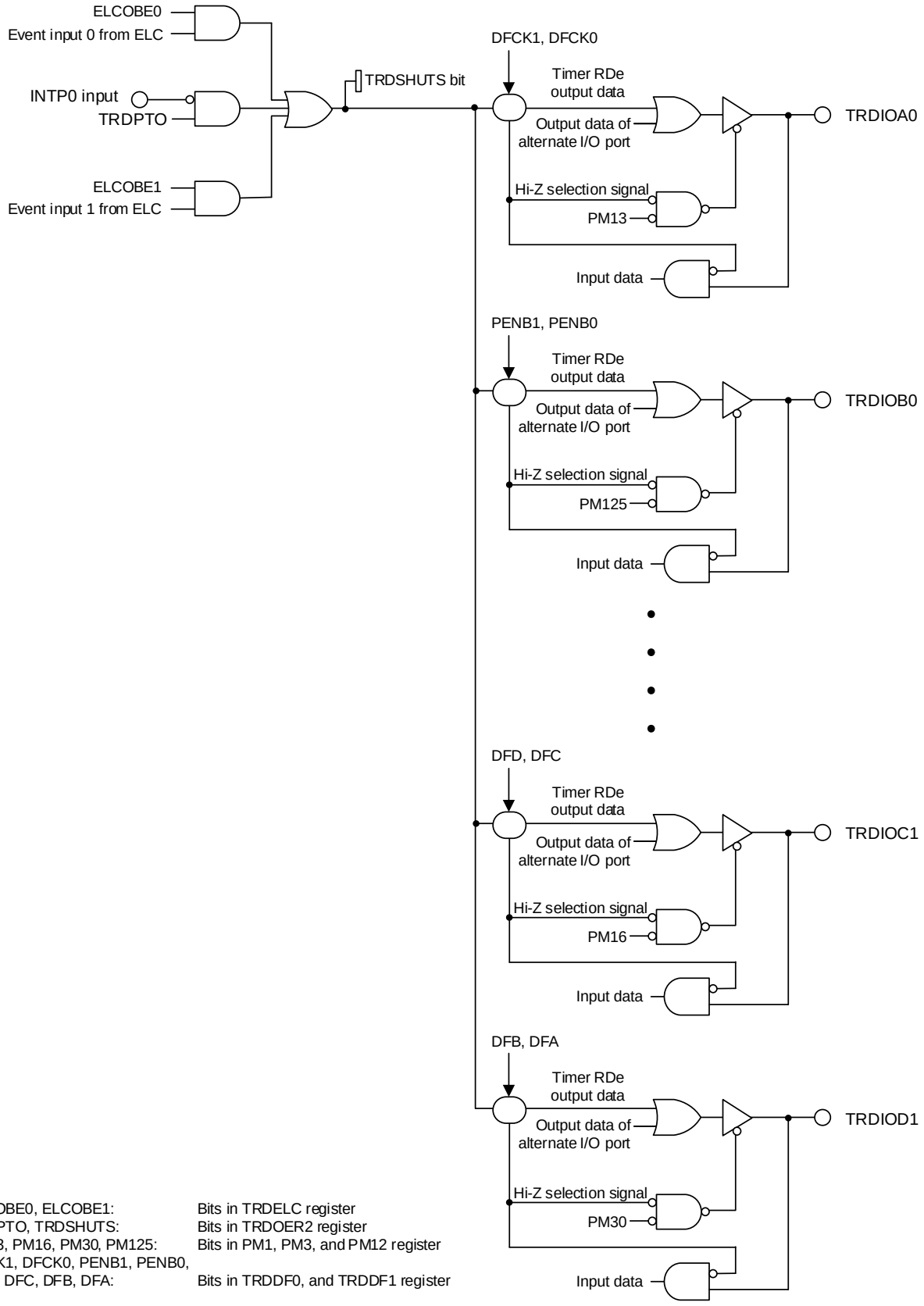
In the PWM function, reset synchronous PWM mode, complementary PWM mode, PWM3 mode, extended PWM mode, and extended complementary PWM mode, the TRDIOji output pin (i = 0 or 1, j = A, B, C, or D) can be forcibly set to an I/O port by the INTP0 pin input, and pulse output can be cut off.

The pins used for output in these functions or modes can function as the output pin of timer RDe when the corresponding bit in the TRDOER1 register is set to 0 (timer RDe 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 RDe 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 register.
- Refer to **8.3.1 (5) Event Input from Event Link Controller (ELC)** for details on pulse forced cutoff by ELC 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).

Figure 8-41. Pulse Output Forced Cutoff



Remark This figure is when bits PIOR70 and PIOR71 are 0. When bits PIOR70 and PIOR71 are 1, the port mode register (PMxx) changes to the corresponding port.

(5) Event Input from Event Link Controller (ELC)

Timer RDe performs two operations by event input from the ELC.

The ELC is only available in the RL78/F25.

See **8.6 PWM Option Unit A (PWMOPA)** for the forced cutoff operation of PWMOPA by event input from the ELC.

(a) Input capture operation D0/D1

Timer RDe 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, PWM3 mode, extended PWM mode, and extended complementary PWM 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, PWM3 mode, extended PWM mode, or extended complementary PWM 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 RDe 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) and DTC

Table 8-17 lists the Timer RDe Modes and Event Output to ELC/DTC.

The ELC is only available in the RL78/F25.

Extended complementary PWM mode does not support event output to ELC and DTC.

Table 8-17. Timer RDe Modes and Event Output to ELC/DTC

Used Mode	Output Source	ELC	DTC
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
	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
	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
Extended PWM mode	Compare match between registers TRD0 and TRDGRA0	Available	Available
	Compare match between registers TRD0 and TRDGRB0	Available	Available
	Compare match between registers TRD0 and TRDCMPB0	–	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 TRDCMPB1	–	Available
Complementary PWM mode	TRD1 register underflow	Available	–

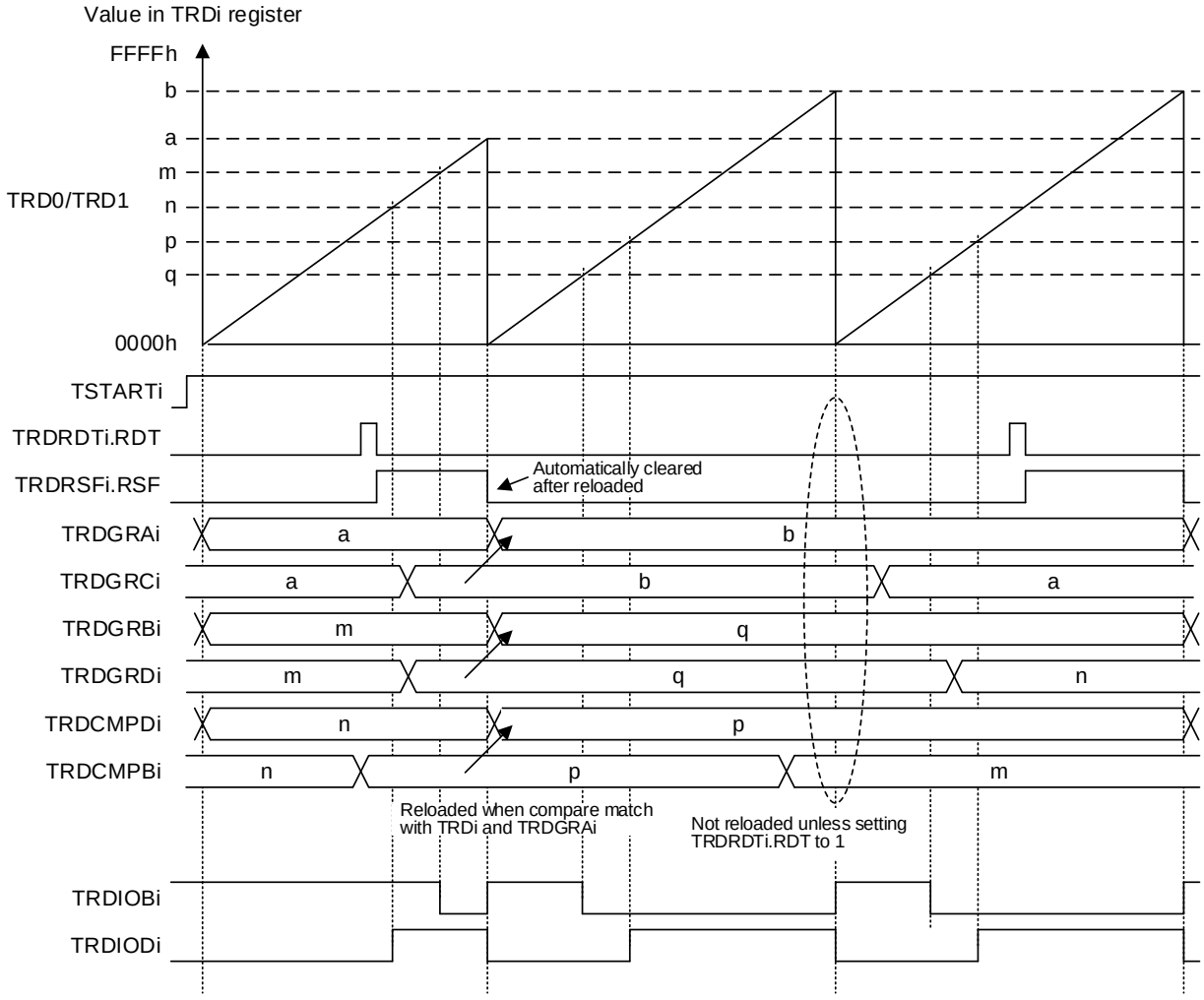
(7) Register Reload Function

Extended PWM mode and extended complementary PWM mode support the register reload function of compare registers. In extended complementary PWM mode, compare registers are reloaded when TRD1 register underflow (or the timming of TRD1 register underflow minus dead-time). In extended PWM mode, compare registers are reloaded when compare match with TRDi register and TRDGRAi register.

When changing the buffer register, make sure that TRDRSFi.RSF (i = 0, 1) is “0” before changing it. After setting the buffer register that needs to be changed, finally set the RDT bit of the TRDRDTi register to 1. By setting TRDRDTi.RDT to "1", TRDRSFi.RSF becomes "1". If the RSF bit in the TRDRSFi register is requested to reload at 1, it is transferred from the buffer register to the comparison register in the next period. If the RSF bit is requested to reload at 0, the value of buffer register is not transferred.

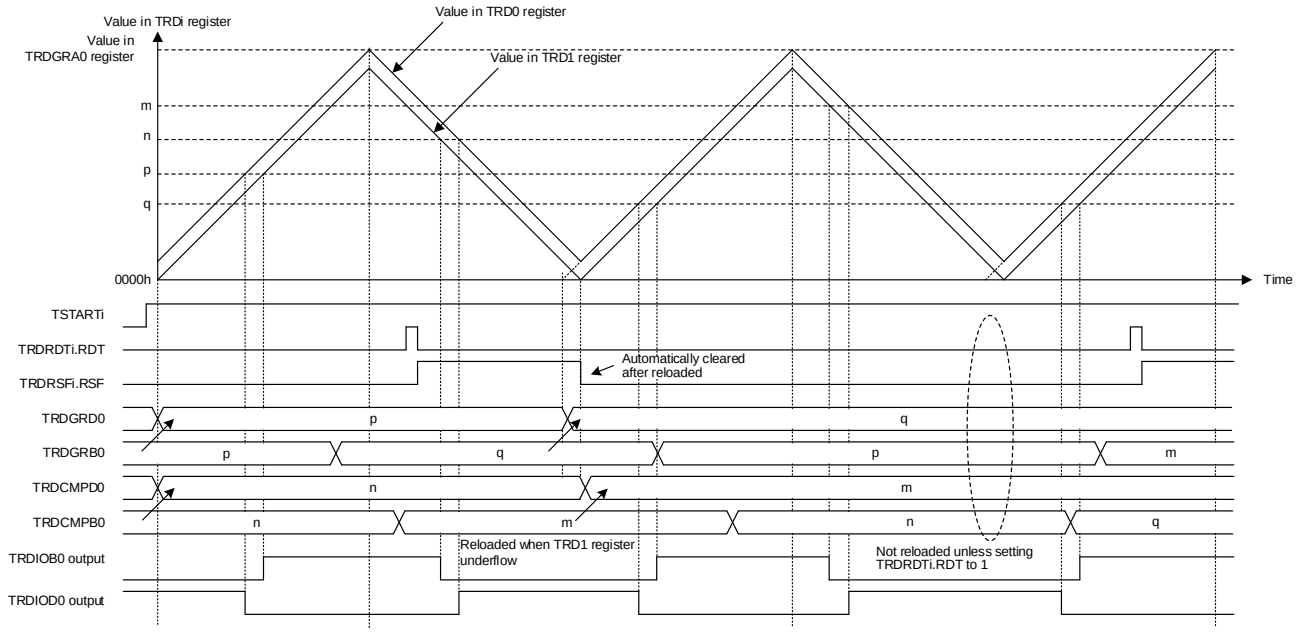
When the timer starts operating in the extended complementary PWM mode, it will be transferred from buffer registers to compare registers regardless of the setting of RSF bit.

Figure 8-42. Reload Function in Extended PWM Mode



Remarks i = 0 or 1
TRDIOBi: default "H", active level "L"
TRDIODi: default "L", active level "H"

Figure 8-43. Reload Function in Extended Complementary PWM Mode



Remark i = 0 or 1

The above figure is for the following conditions.
- TRDFCR.CPSS = 1

The target of the register reload function differs depending on the mode. **Table 8-18** shows the target registers for the register reload function.

Table 8-18. Reload Register in Each Modes

Operating Mode	Reload Target Registers	
	TRDRDT0.RDT = 1	TRDRDT1.RDT = 1
Extended PWM mode	TRDGRA0 TRDGRB0 TRDCMPB0	TRDGRA1 TRDGRB1 TRDCMPB1
Extended complementary PWM mode	—	TRDGRB0 TRDCMPB0 TRDGRA1 TRDCMPA1 TRDGRB1 TRDCMPB1 TRDADTC0 TRDADTC1

To rewrite the compare register by the register reload function, follow the procedure below.

(1) Confirm that TRDRSFi.RSF (i = 0, 1) is "0".

(2) Set value to reload all at once in the buffer registers.

(3) Write "1" to TRDRDTi.RDT.

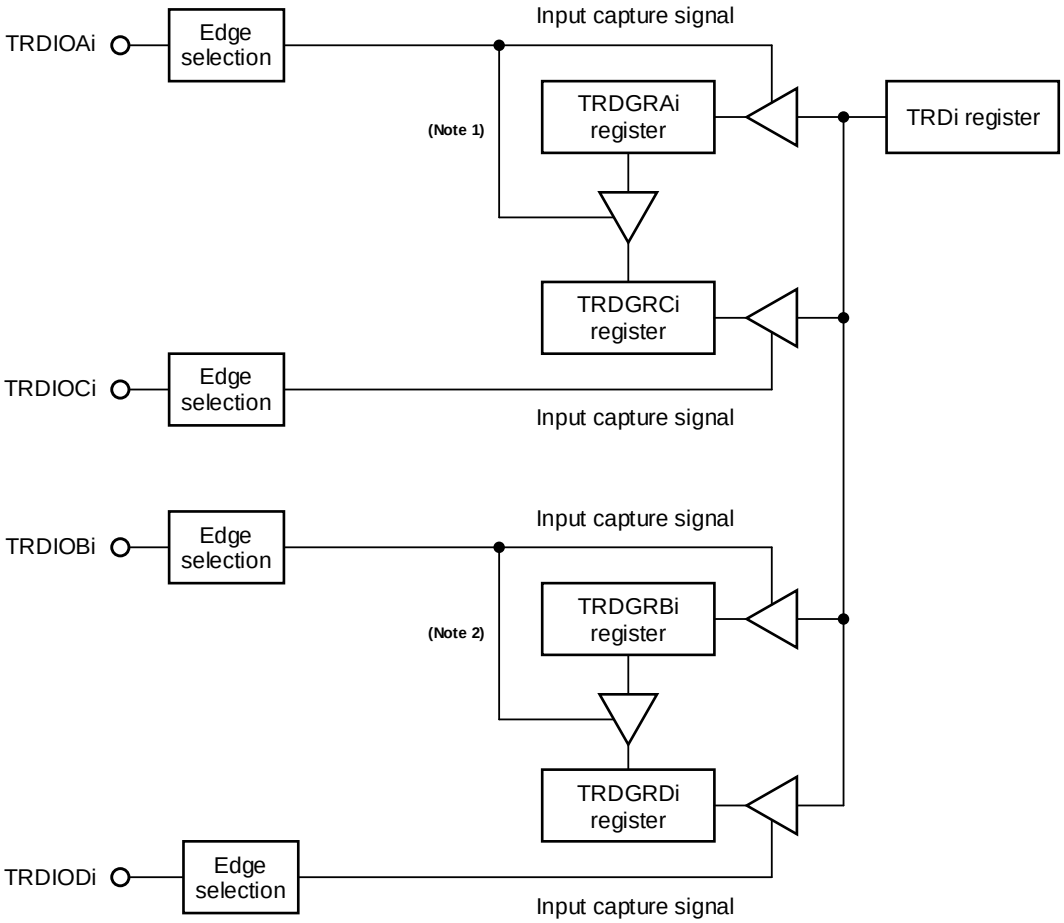
(4) TRDRSFi.RSF is set to "1", and TRDRSFi.RSF is automatically cleared to "0" after the simultaneous reloading is completed.

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 8-44 shows the Block Diagram of Input Capture Function, **Table 8-19** lists the Input Capture Function Specifications, and **Figure 8-45** shows an Operation Example of Input Capture Function.

Figure 8-44. Block Diagram of Input Capture Function



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).

Remark i = 0 or 1

Table 8-19. Input Capture Function Specifications

Item	Specification
Count sources Note	f_{CLK}, f_{PLL}, f_{IH}, f_{SUB}, f_{IL} 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). 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) • TRDi register overflow
TRDIOA0 pin function	I/O port, input-capture input, or TRDCLK0 (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). Data can be written to the TRDi register. • When the TRDSYNC bit in the TRDMR register is 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	• Input-capture input pin selection Either one pin or multiple pins of TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi. • Input-capture input active edge selection Rising edge, falling edge, or both rising and falling edges • Timing for setting the TRDi register to 0000H At overflow or input capture • Buffer operation (see **8.3.1 (2) Buffer Operation**) • Synchronous operation (see **8.3.1 (3) Synchronous Operation**) • Digital filter The TRDIOji input is sampled, and when the sampled input level match three times, that level is determined. • Input capture operation by event input from event link controller (ELC)

Note When selecting the count source for the timer RDe, set the same clock source as the count source for f_{CLK} before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

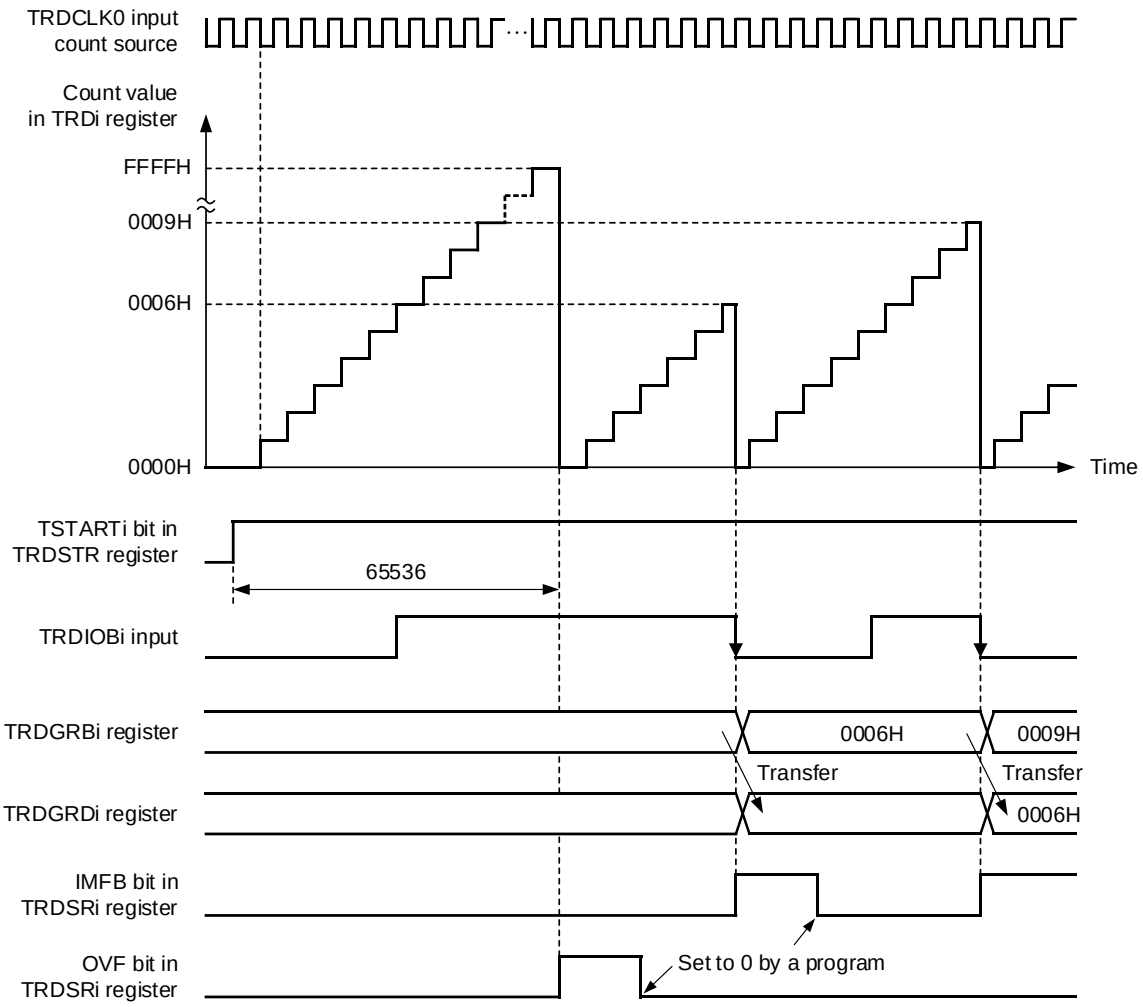
Remark i = 0 or 1, j = A, B, C, or D

(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. Figure 8-45 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 8-45. Operation Example of Input Capture Function



Remark i = 0 or 1

The above diagram applies under the following conditions:

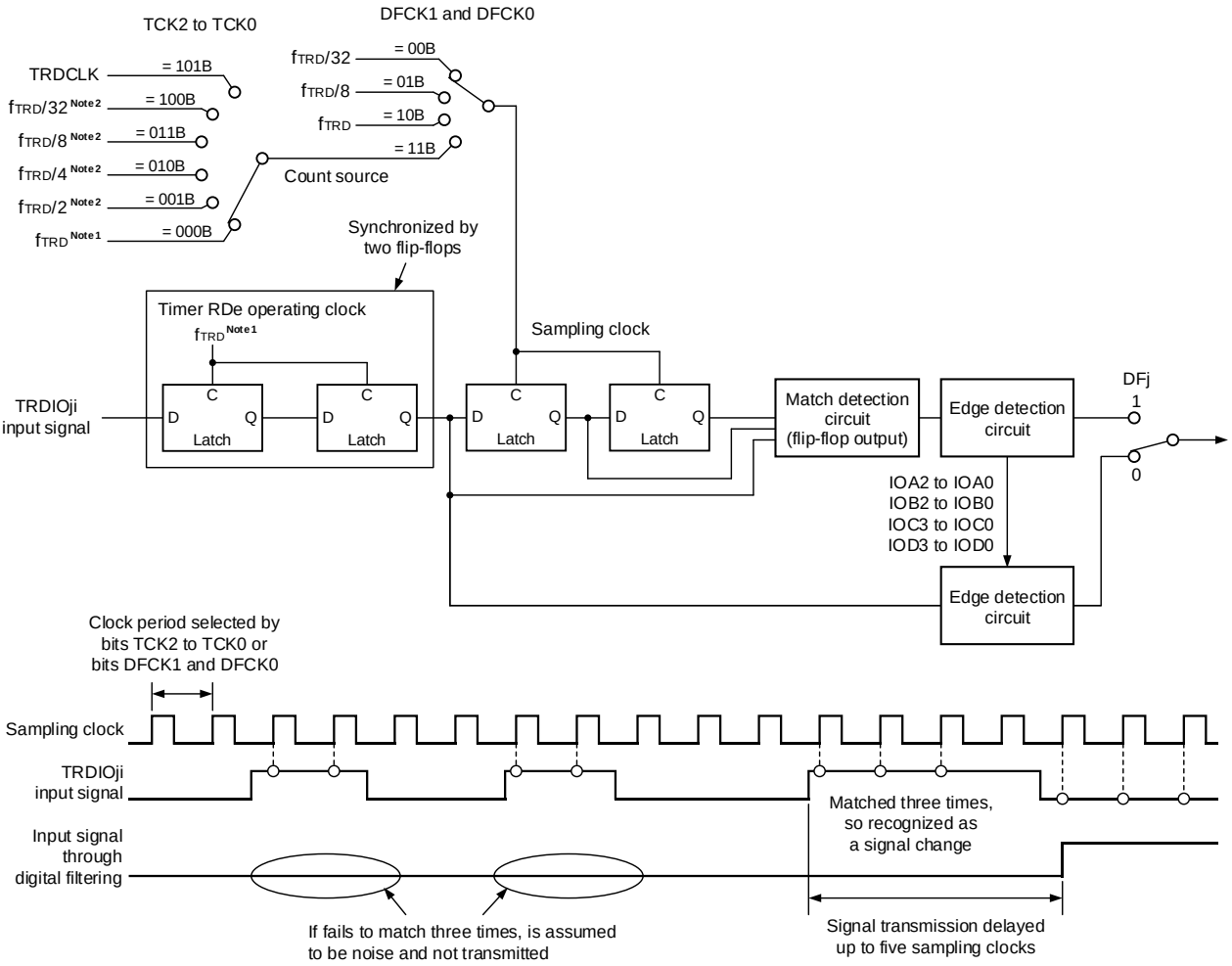
- Bits CCLR2 to CCLR0 in the TRDCRi register are set to 010B (TRDi register is set to 0000H by TRDGRBi register input capture).
- Bits TCK2 to TCK0 in the TRDCRi register are set to 101B (TRDCLK0 input for the count source).
- Bits CKEG1 and CKEG0 in the TRDCRi register are set to 01B (count at the falling edge for the count source).
- Bits IOB2 to IOB0 in the TRDIORAi register are set to 101B (input capture at the falling edge of TRDIOBi input).
- The TRDBFDi bit in the TRDMR register is set to 1 (TRDGRDi register is buffer register for TRDGRBi 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 8-46 shows the Block Diagram of Digital Filter.

Figure 8-46. Block Diagram of Digital Filter



Notes 1. As the timer RDe operating clock (f_{TRD}), f_{CLK} is selected when FRQSEL4 = 0 in the user option byte (000C2H/040C2H), (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 8-37**.

When selecting the count source for the timer RDe, 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 RDe operating clock (f_{TRD}).

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

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 8-47 shows the Block Diagram of Output Compare Function, **Table 8-20** lists the Output Compare Function Specifications, and **Figure 8-48** shows an Operation Example of Output Compare Function.

Figure 8-47. Block Diagram of Output Compare Function

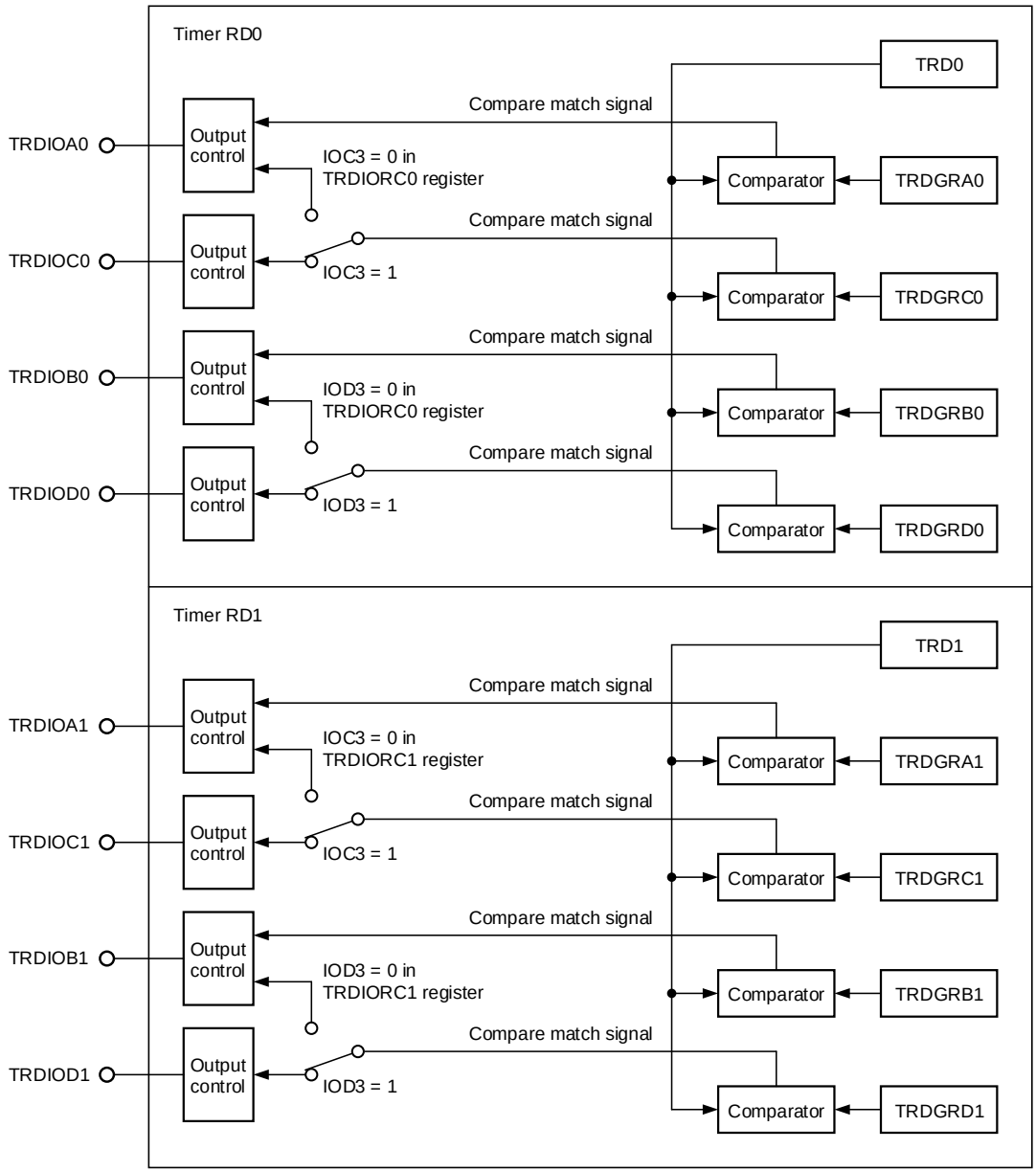


Table 8-20. Output Compare Function Specifications

Item	Specification
Count sources Note	f_{CLK}, f_{PLL}, f_{IH}, f_{SUB}, f_{IL} 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). 1/fk × 65536 fk: Frequency of count source • 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). 1/fk × (n + 1) 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. The output compare output pin holds the output level before the count stops. • When the CSELi bit in the TRDSTR register is set to 0, the count stops at the compare match with the TRDGRAi register. 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) • TRDi register overflow
TRDIOA0 pin function	I/O port, output-compare output, or TRDCLK0 (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. • 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 Either one pin or multiple pins of TRDIOAi, TRDIOBi, TRDIOCi, and TRDIODi. • Output level selection at compare match Low output, high output, or inverted output level • Initial output level selection The level can be set for the period from the count start to the compare match. • Timing for setting the TRDi register to 0000H Overflow or compare match in the TRDGRAi register • Buffer operation (see **8.3.1 (2) Buffer Operation**) • Synchronous operation (see **8.3.1 (3) Synchronous Operation**) • Changing output pins for registers TRDGRCi and TRDGRDi 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. • Pulse output forced cutoff signal input (see **8.3.1 (4) Pulse Output Forced Cutoff**) • Timer RDe can be used as the internal timer without output.

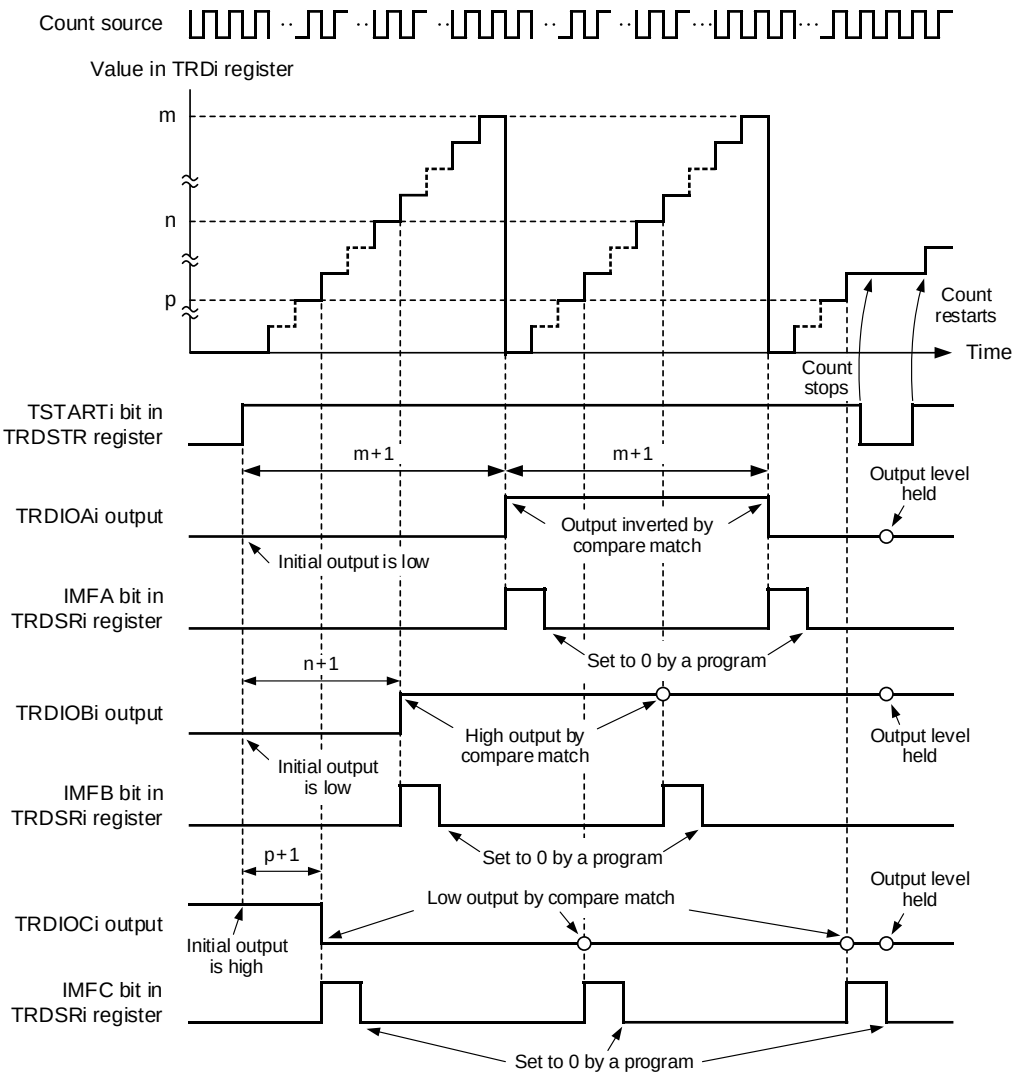
Note When selecting the count source for the timer RDe, set the same clock source as the count source for f_{CLK} before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

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 8-48. 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 :

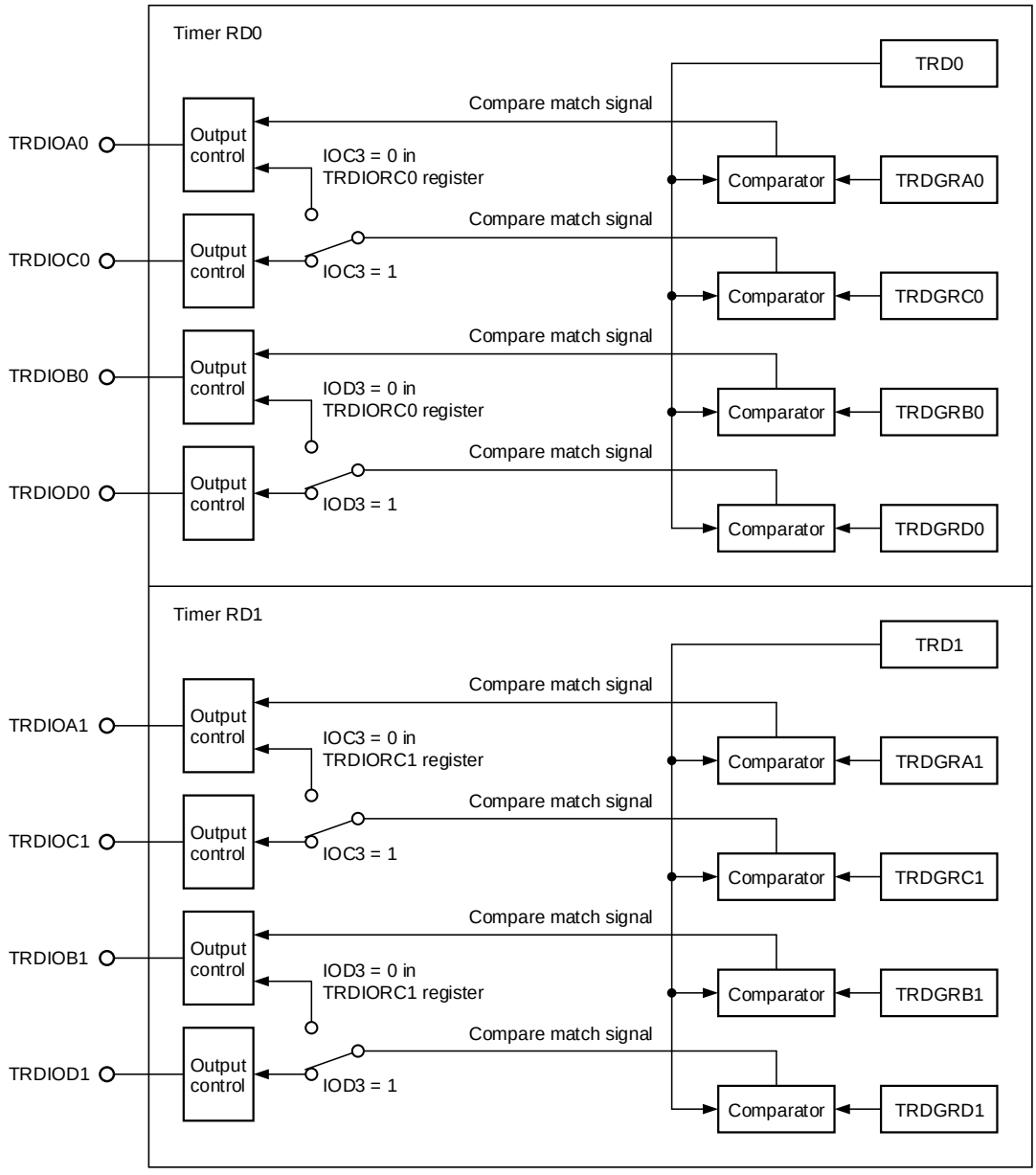
- The CSELi bit in the TRDSTR register is set to 1 (TRDi is not stopped by compare match).
- Bits TRDBFCi and TRDBFDi in the TRDMR register are set to 0 (TRDGRCi and TRDGRDi do not operate as buffers).
- Bits EAi, EBi, and ECi in the TRDOER1 register are set to 0 (TRDIOAi, TRDIOBi and TRDIOCi output 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 8-49. Changing Output Pins in Registers TRDGRCi and TRDGRDi

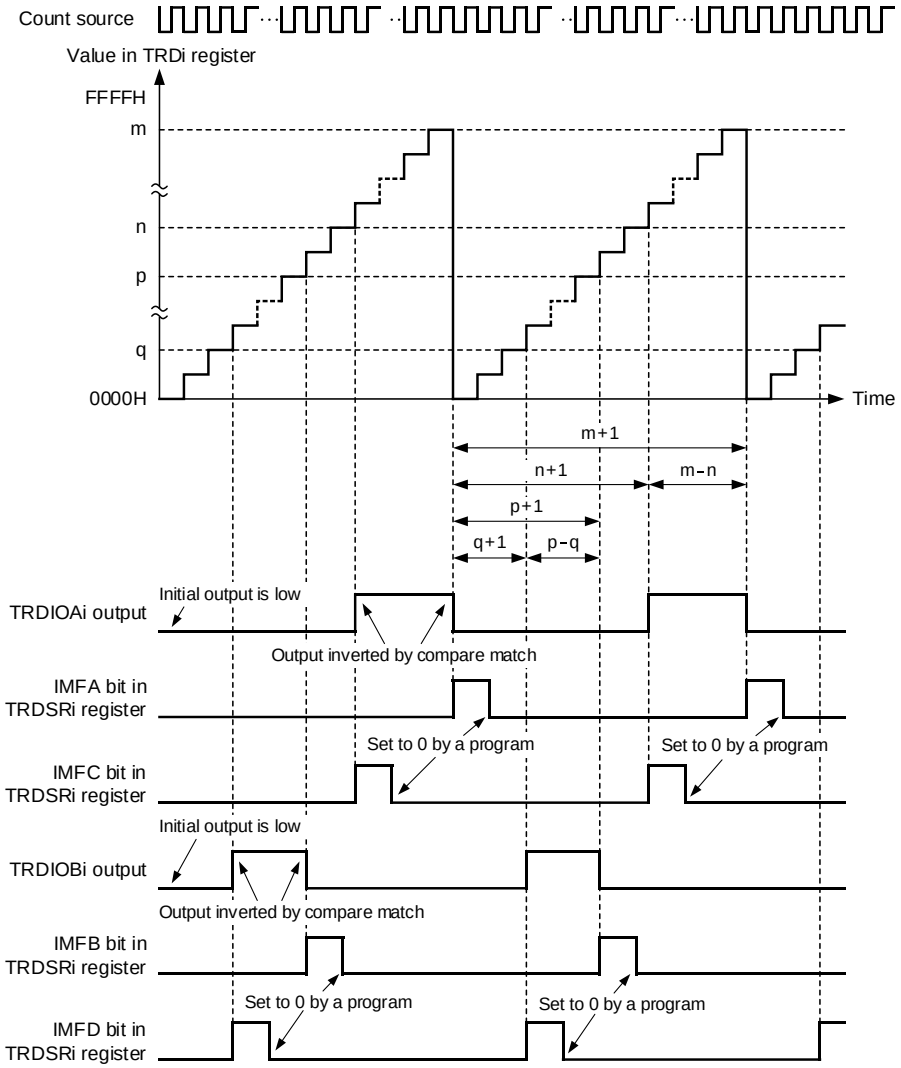


Change output pins in registers TRDGRCi and TRDGRDi as follows:

- 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 8-50 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 8-50. 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).

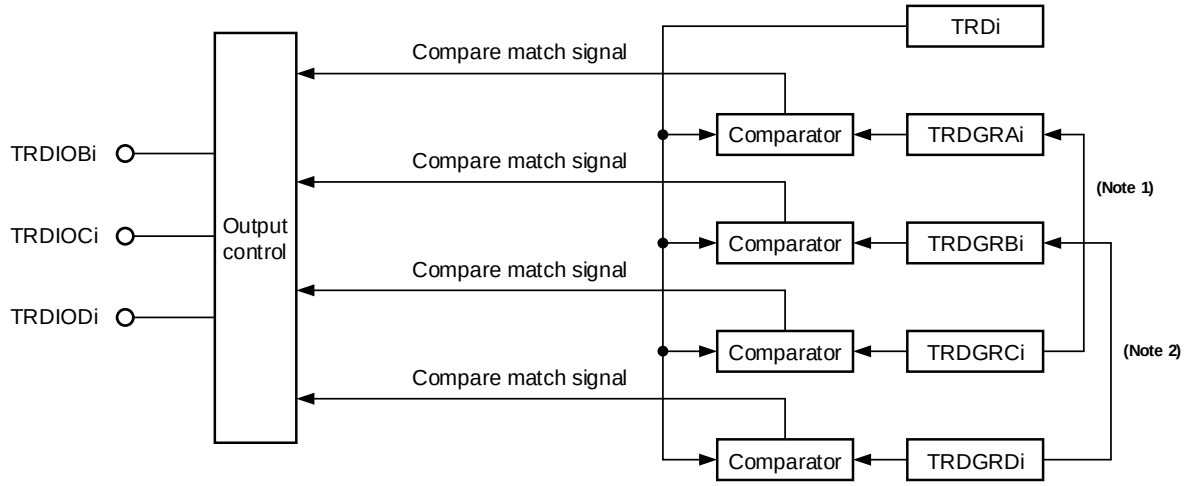
8.3.4 PWM Function

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 8-51 shows the Block Diagram of PWM function, **Table 8-21** lists the PWM Function Specifications, and **Figure 8-52** and **Figure 8-53** show Operation Examples in PWM Function.

Figure 8-51. Block Diagram of PWM Function

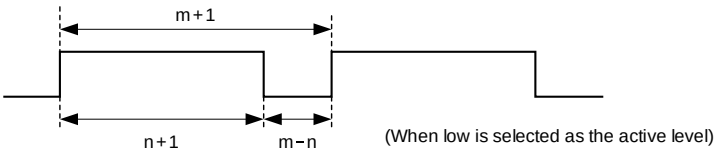


Notes 1. When the TRDBFCi bit in the TRDMR register is set to 1 (TRDGRCi register is buffer register for TRDGRAi register).

2. When the TRDBFDi bit in the TRDMR register is set to 1 (TRDGRDi register is buffer register for TRDGRBi register).

Remark i = 0 or 1

Table 8-21. PWM Function Specifications

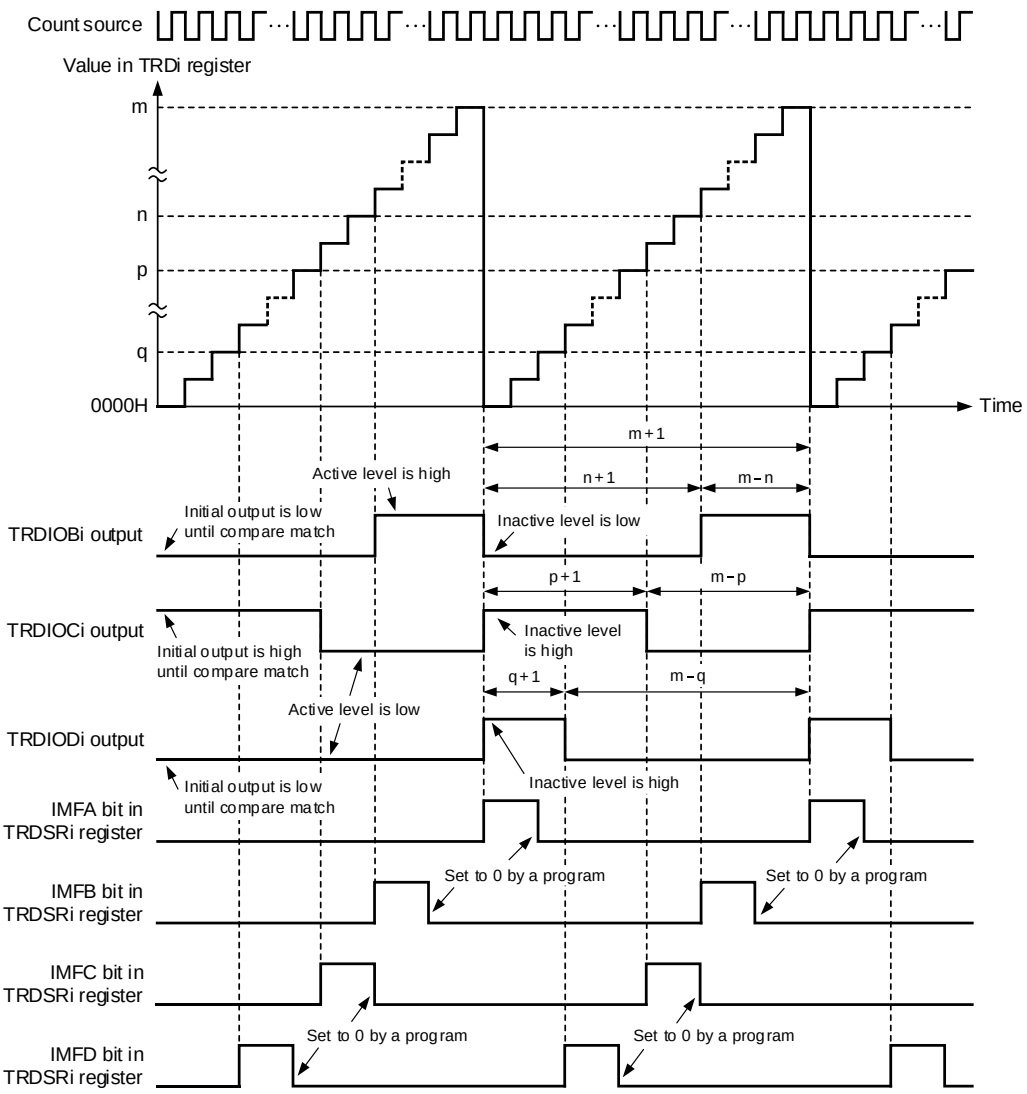
Item	Specification
Count sources Note	f_{CLK}, f_{PLL}, f_{IH}, f_{SUB}, f_{IL} External signal input to the TRDCLK0 pin (active edge selected by a program)
Count operations	Increment
PWM waveform	PWM period: 1/fk × (m + 1) Active level width: 1/fk × (m − n) Inactive level width: 1/fk × (n + 1) fk: Frequency of count source m: Value set in the TRDGRAi register n: Value set in the TRDGRji register
	 (When low is selected as the active level)
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. The PWM output pin holds the output level before the count stops. • When the CSELi bit in the TRDSTR register is set to 0, the count stops at the compare match with the TRDGRAi register. 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) • TRDi register overflow
TRDIOA0 pin function	I/O port or TRDCLK0 (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 Either one pin or multiple pins of TRDIOBi, TRDIOCi, and TRDIODi. • Active level selectable for each pin. • Initial output level selectable for each pin. • Synchronous operation (see **8.3.1 (3) Synchronous Operation**) • Buffer operation (see **8.3.1 (2) Buffer Operation**) • Pulse output forced cutoff signal input (see **8.3.1 (4) Pulse Output Forced Cutoff**)

Note When selecting the count source for the timer RDe, 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 8-52. Operation Example in PWM Function

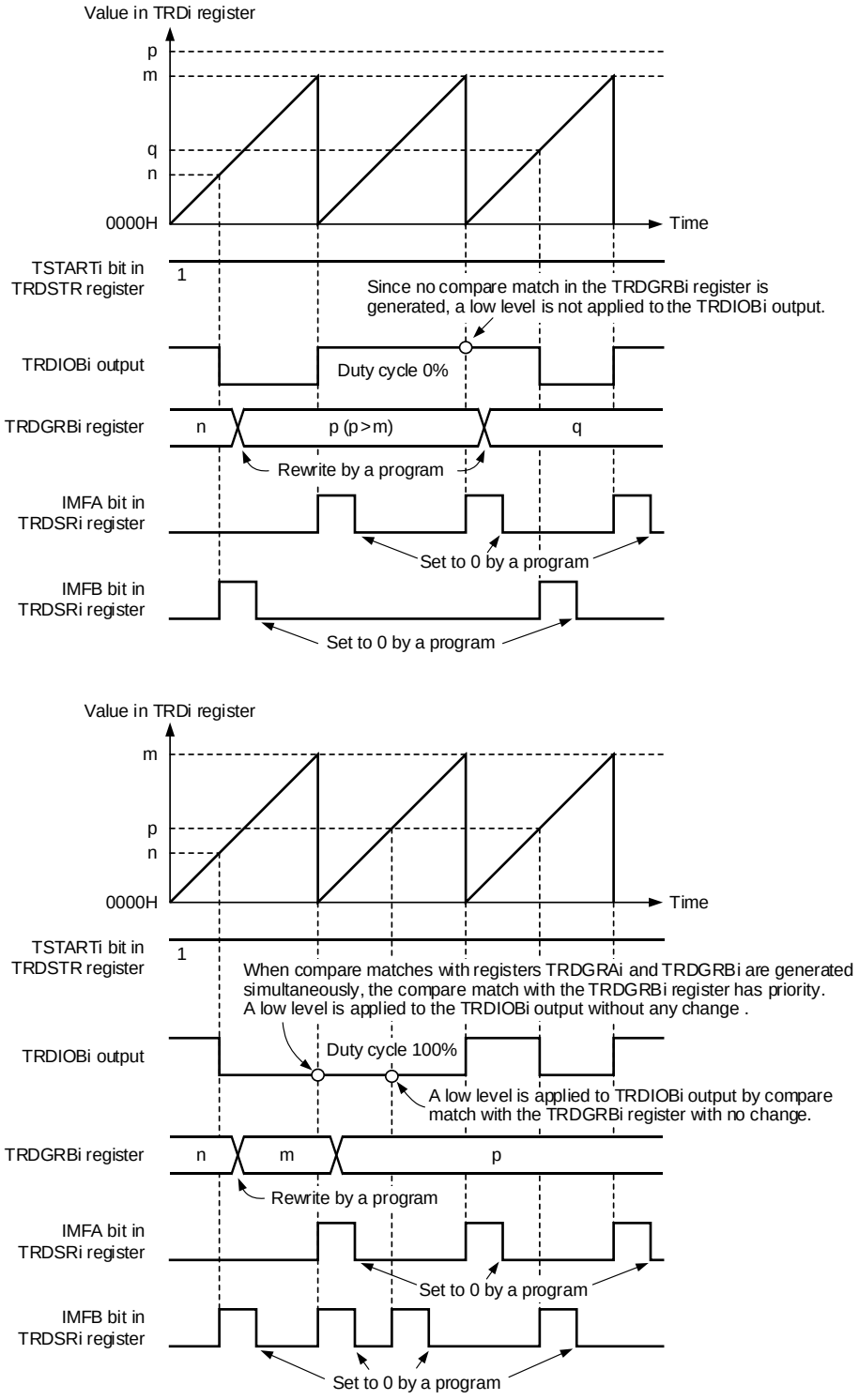


Remark i = 0 or 1
m: Value set in TRDGRAi register
n: Value set in TRDGRBi register
p: Value set in TRDGRCi register
q: Value set in TRDGRDi register

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 8-53. 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).

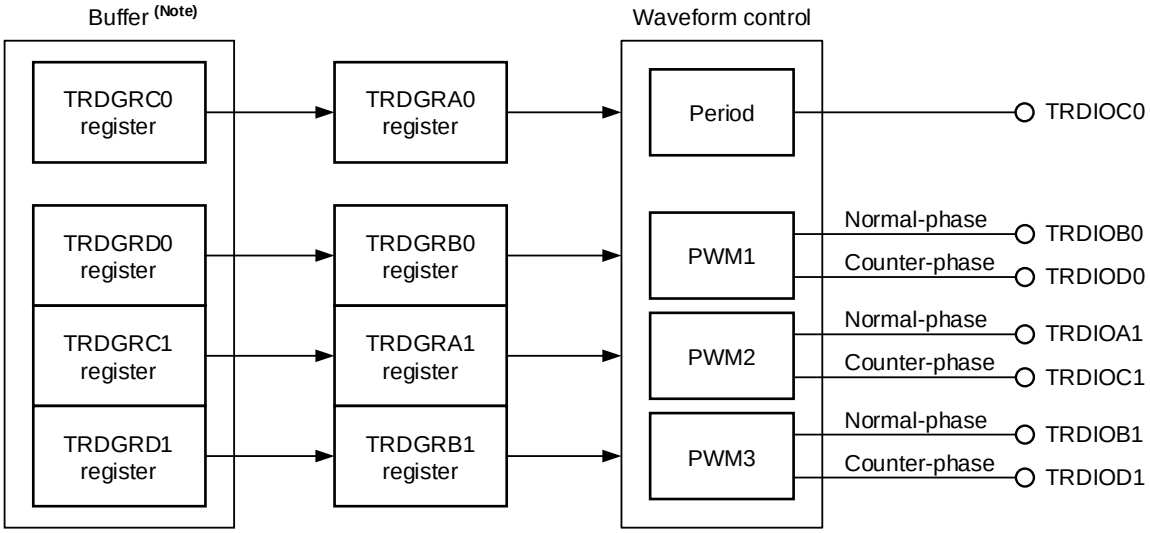
8.3.5 Reset Synchronous PWM Mode

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 8-54 shows the Block Diagram of Reset Synchronous PWM Mode, **Table 8-22** lists the Reset Synchronous PWM Mode Specifications, **Figure 8-55** shows an Operation Example in Reset Synchronous PWM Mode.

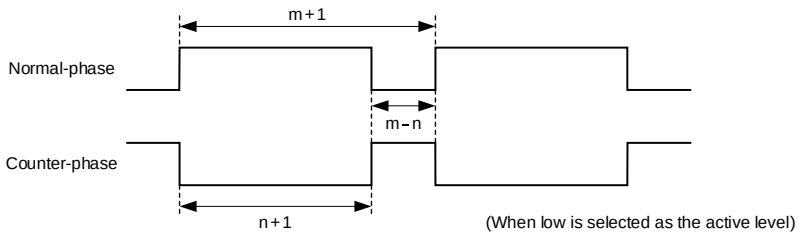
See **Figure 8-53. Operation Example in PWM Function (Duty Cycle 0%, Duty Cycle 100%)** for an operation example in PWM Mode with duty cycle 0% and duty cycle 100%.

Figure 8-54. Block Diagram of Reset Synchrnous PWM Mode



Note When bits TRDBFC0, TRDBFD0, TRDBFC1, and TRDBFD1 in the TRDMR register are set to 1 (buffer register).

Table 8-22. Reset Synchronous PWM Mode Specifications

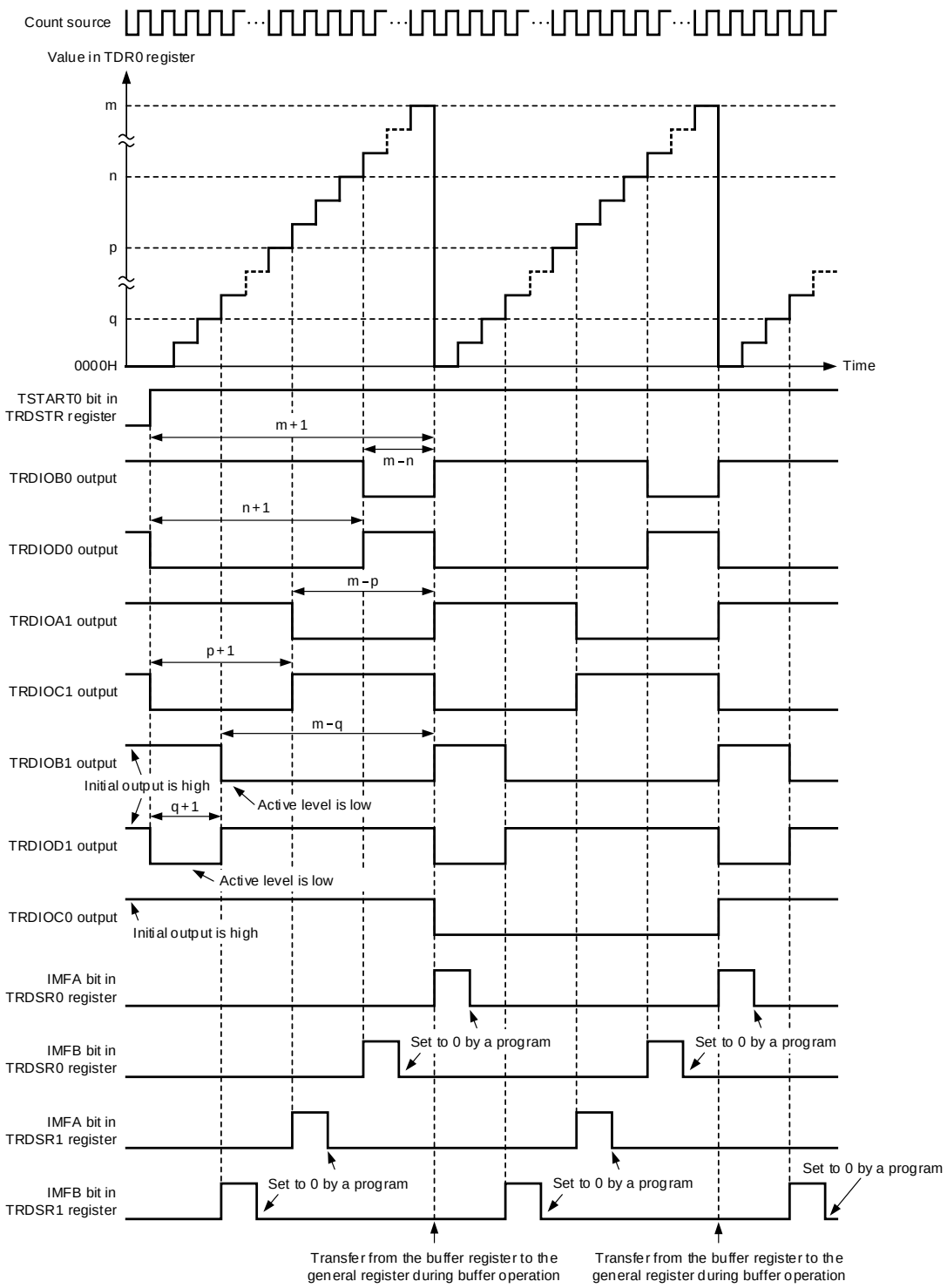
Item	Specification
Count sources [Note]	f_{CLK}, f_{PLL}, f_{IH}, f_{SUB}, f_{IL} External signal input to the TRDCLK0 pin (active edge selected by a program)
Count operations	The TRD0 register is incremented (the TRD1 register is not used).
PWM waveform	PWM period: 1/fk × (m + 1) Active level of normal-phase: 1/fk × (m − n) Inactive level of counter-phase: 1/fk × (n + 1) fk: Frequency of count source m: Value set in the TRDGRA0 register n: Value set in the TRDGRB0 register (PWM1 output) Value set in the TRDGRA1 register (PWM2 output) Value set in the TRDGRB1 register (PWM3 output)
	
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. The PWM output pin outputs the initial output level selected by bits OLS0 and OLS1 in the TRDFCR register. • When the CSEL0 bit in the TRDSTR register is set to 0, the count stops at the compare match with the TRDGRA0 register. 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) • TRD0 register overflow
TRDIOA0 pin function	I/O port or TRDCLK0 (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. • Buffer operation (see **8.3.1 (2) Buffer Operation**) • Pulse output forced cutoff signal input (see **8.3.1 (4) Pulse Output Forced Cutoff**)

Note When selecting the count source for the timer RDe, 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 8-55. Operation Example in Reset Synchronous PWM Mode



Remark 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:

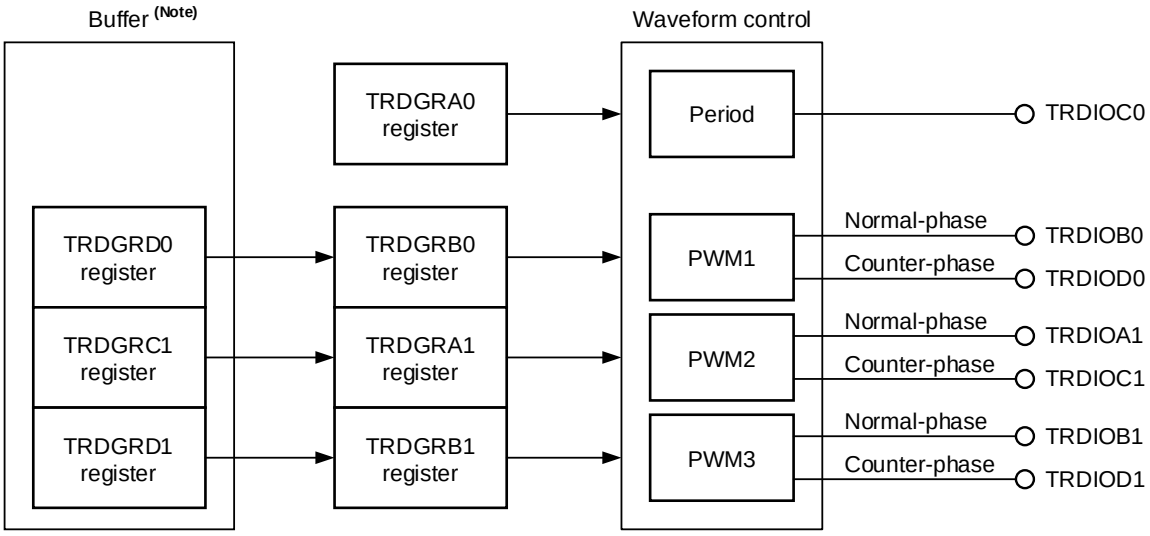
- Bits OLS1 and OLS0 in the TRDFCR register are set to 0 (initial output level is high, active level is low).

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).

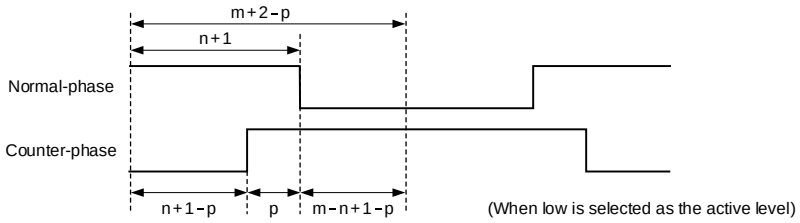
Figure 8-56 shows the Block Diagram of Complementary PWM Mode, **Table 8-23** lists the Complementary PWM Mode Specifications, and **Figure 8-57** shows the Output Model of Complementary PWM Mode, and **Figure 8-58** shows an Operation Example in Complementary PWM Mode.

Figure 8-56. Block Diagram of Complementary PWM Mode



Note When bits TRDBFD0, TRDBFC1, and TRDBFD1 in the TRDMR register are set to 1 (buffer register).

Table 8-23. Complementary PWM Mode Specifications

Item	Specification
Count sources Note 1	f_{CLK}, f_{PLL}, f_{IH}, f_{SUB}, f_{IL} External signal input to the TRDCLK0 pin (active edge selected by a program) 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. 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 Dead time: p Active level width of normal-phase: 1/fk × (m − n + 1 − p) × 2 Active level width of counter-phase: 1/fk × (n + 1 − p) × 2 fk: Frequency of count source m: Value set in the TRDGRA0 register n: Value set in the TRDGRB0 register (PWM1 output) Value set in the TRDGRA1 register (PWM2 output) Value set in the TRDGRB1 register (PWM3 output) p: Value set in the TRD0 register
	
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) • TRD1 register underflow
TRDIOA0 pin function	I/O port or TRDCLK0 (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)** • The normal-phase and counter-phase active level and initial output level are selected individually. • Transfer timing from the buffer register selection

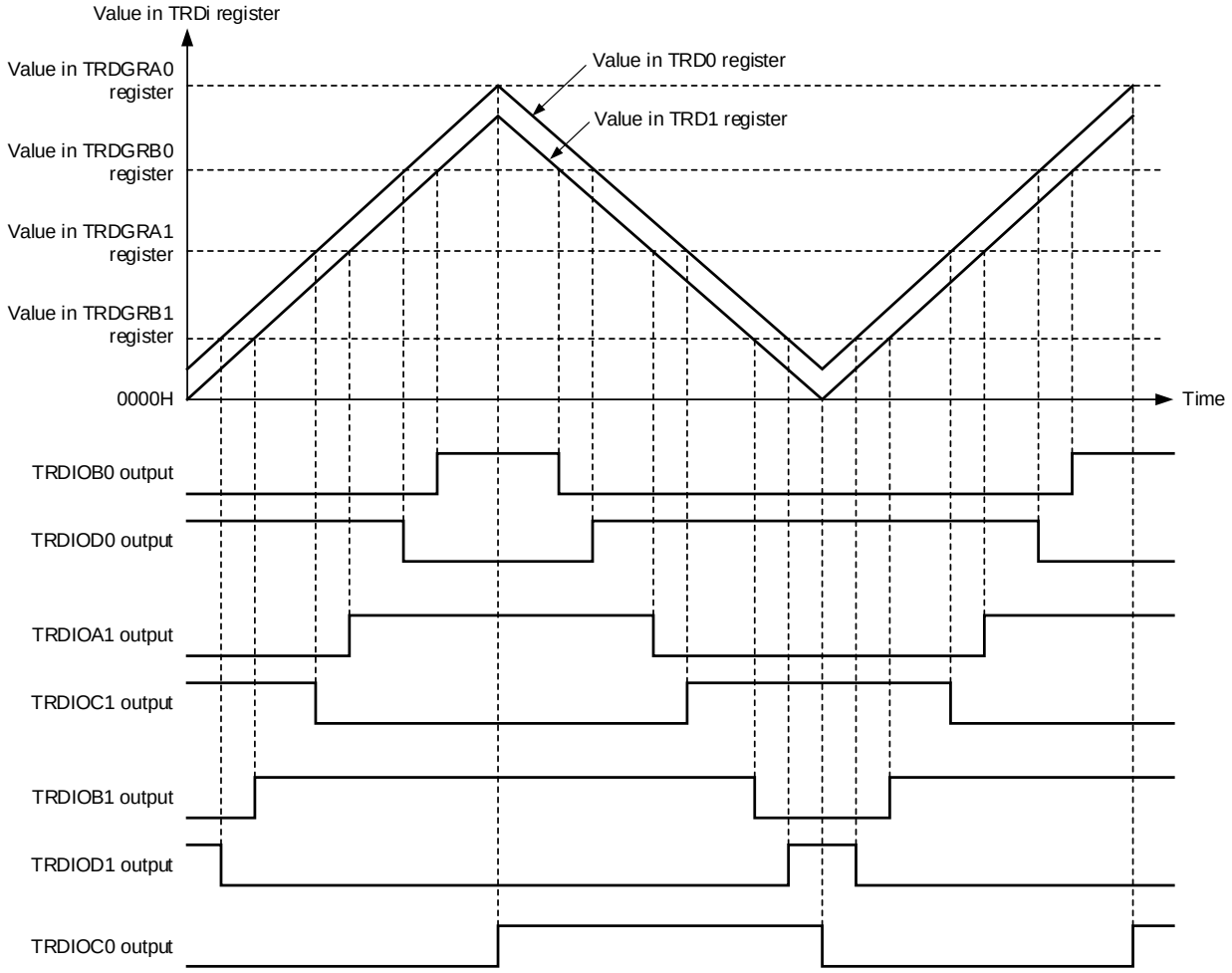
Notes 1. When selecting the count source for the timer RDe, set the same clock source as the count source for f_{CLK} before setting bit 4 (TRD0EN) in the peripheral enable register 1 (PER1).

2. After a count starts, the complementary PWM period is fixed.

Remark i = 0 or 1, j = A, B, C, or D

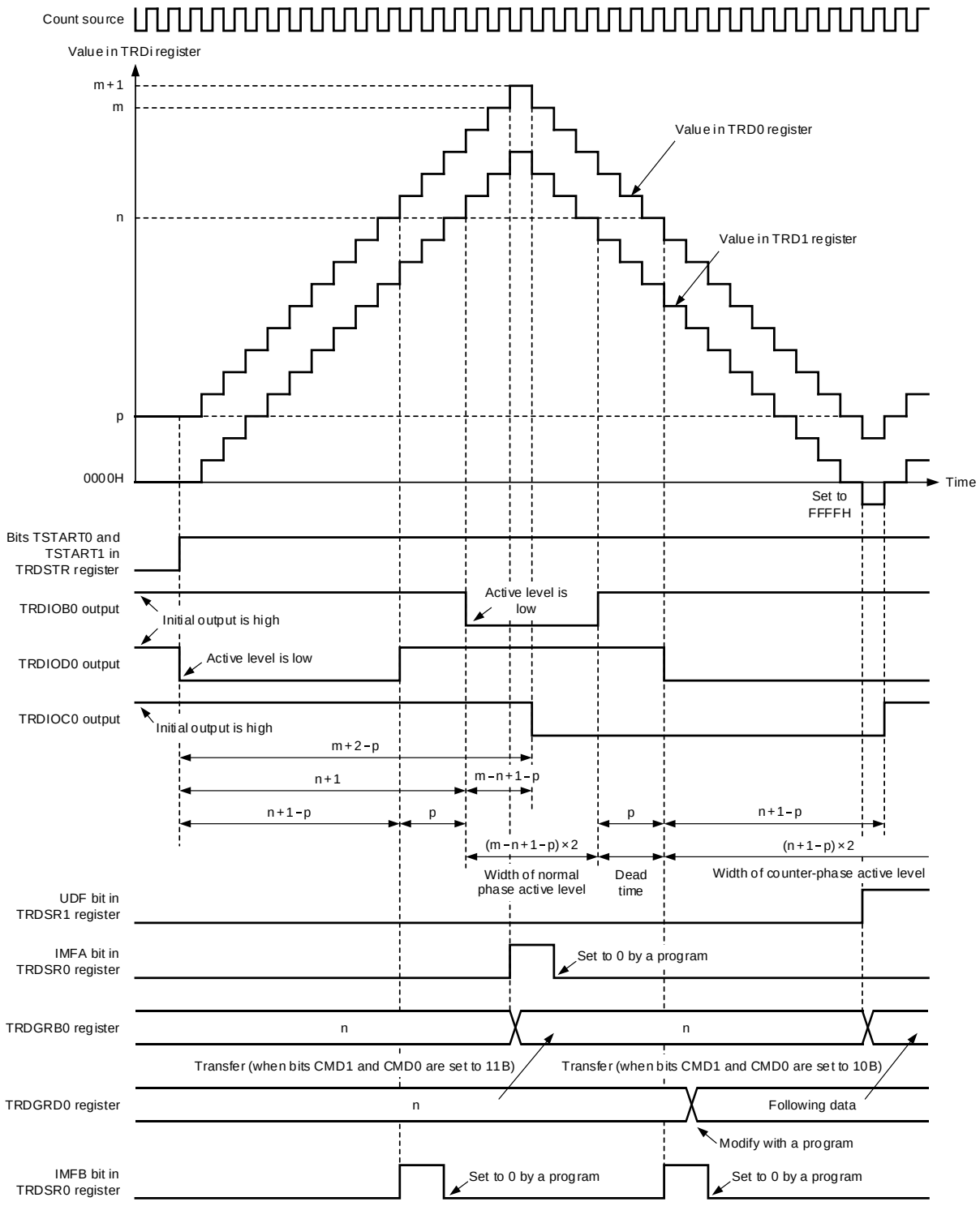
(1) Operation Example

Figure 8-57. Output Model of Complementary PWM Mode



Remark i = 0 or 1

Figure 8-58. 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 8-59 shows the Block Diagram of PWM3 Mode, **Table 8-24** lists the PWM3 Mode Specifications, and **Figure 8-60** shows an Operation Example in PWM3 Mode.

Figure 8-59. Block Diagram of PWM3 Mode

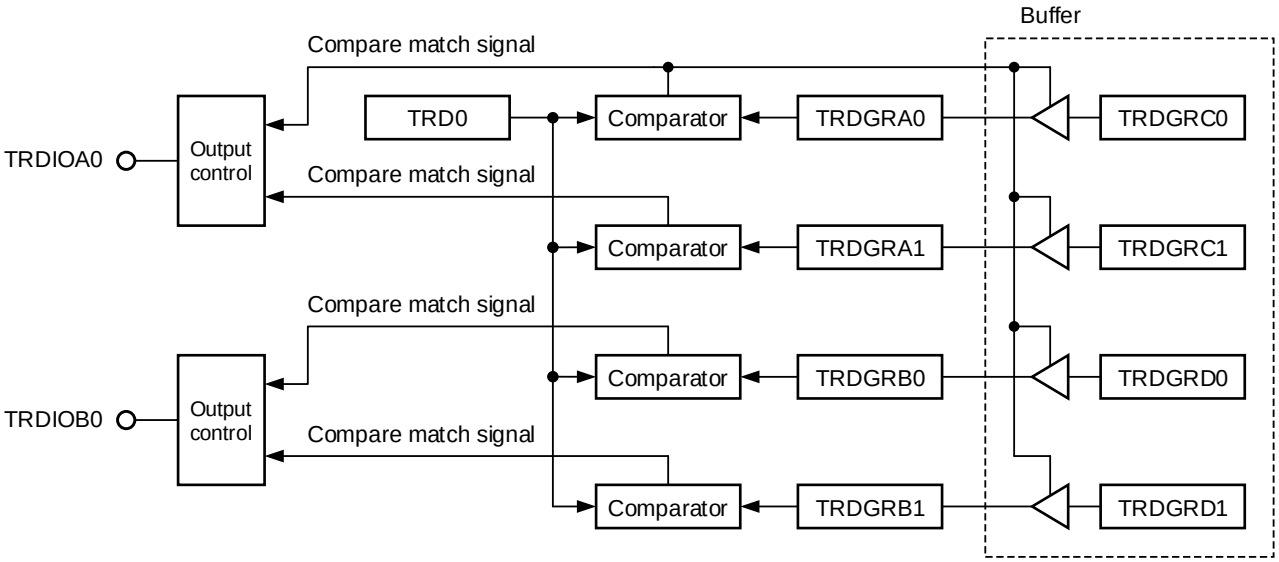
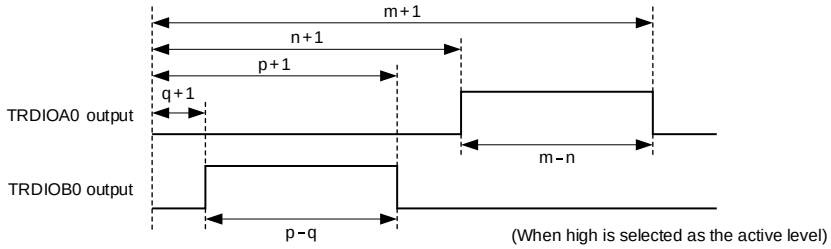


Table 8-24. PWM3 Mode Specifications

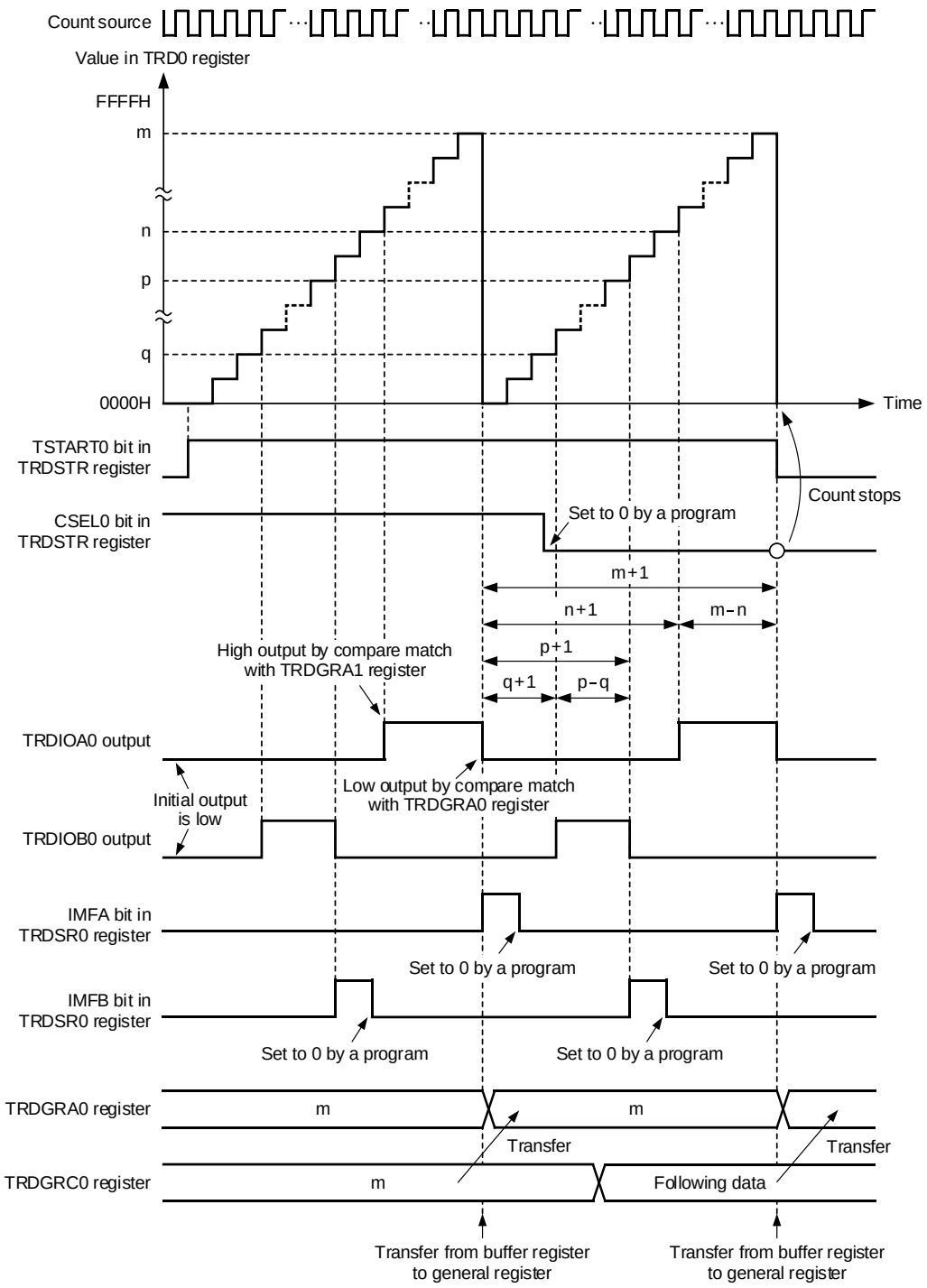
Item	Specification
Count sources Note	f_{CLK}, f_{PLL}, f_{IH}, f_{SUB}, f_{IL}
Count operations	The TRD0 register is incremented (the TRD1 register is not used).
PWM waveform	PWM period: 1/fk × (m + 1) Active level width of TRDIOA0 output: 1/fk × (m − n) Active level width of TRDIOB0 output: 1/fk × (p − q) fk: Frequency of count source m: Value set in the TRDGRA0 register n: Value set in the TRDGRA1 register p: Value set in the TRDGRB0 register q: Value set in the TRDGRB1 register
	
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. The PWM output pin holds the output level before the count stops. • When the CSEL0 bit in the TRDSTR register is set to 0, the count stops at compare match with the TRDGRA0 register. 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) • 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**) • Active level selectable for each pin. • Buffer operation (see **8.3.1 (2) Buffer Operation**)

Note When selecting the count source for the timer RDe, 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 8-60. 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 :

- 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).

8.3.8 Extended PWM Mode

In extended PWM mode, up to 2 PWM waveforms (TRDIOBi and TRDIODi) can be output for each timer RDi (i = 0 or 1). TRD0 and TRD1 can be synchronized to output 4 PWM waveforms with the same period. Also, in this mode, PWM period and PWM active level width can be rewritten by using register reload function. (see **8.3.1 (7) Register Reload Function**.)

As extended functions, it is equipped with a counter restart function by TRDiRES input, and dithering, gate functions.

Figure 8-61 shows the Block Diagram of Extended PWM Mode, **Table 8-25** lists the Extended PWM Mode Specifications, and **Figure 8-62** and **Figure 8-63** show Operation Examples in Extended PWM Mode.

Figure 8-61. Block Diagram of Extended PWM Mode

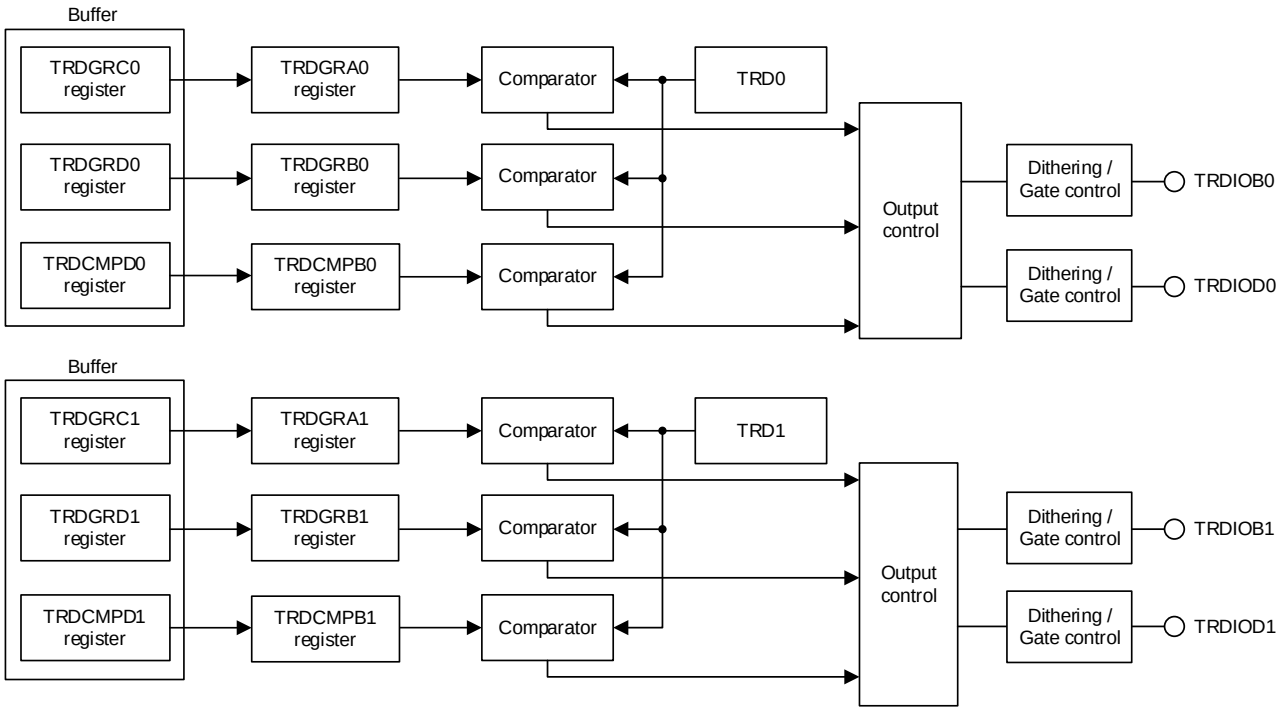
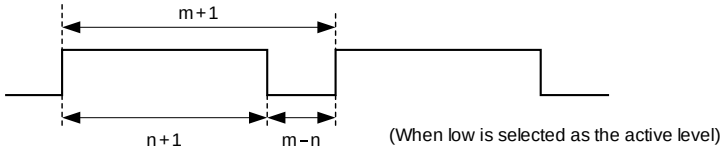


Table 8-25. Extended PWM Mode Specifications

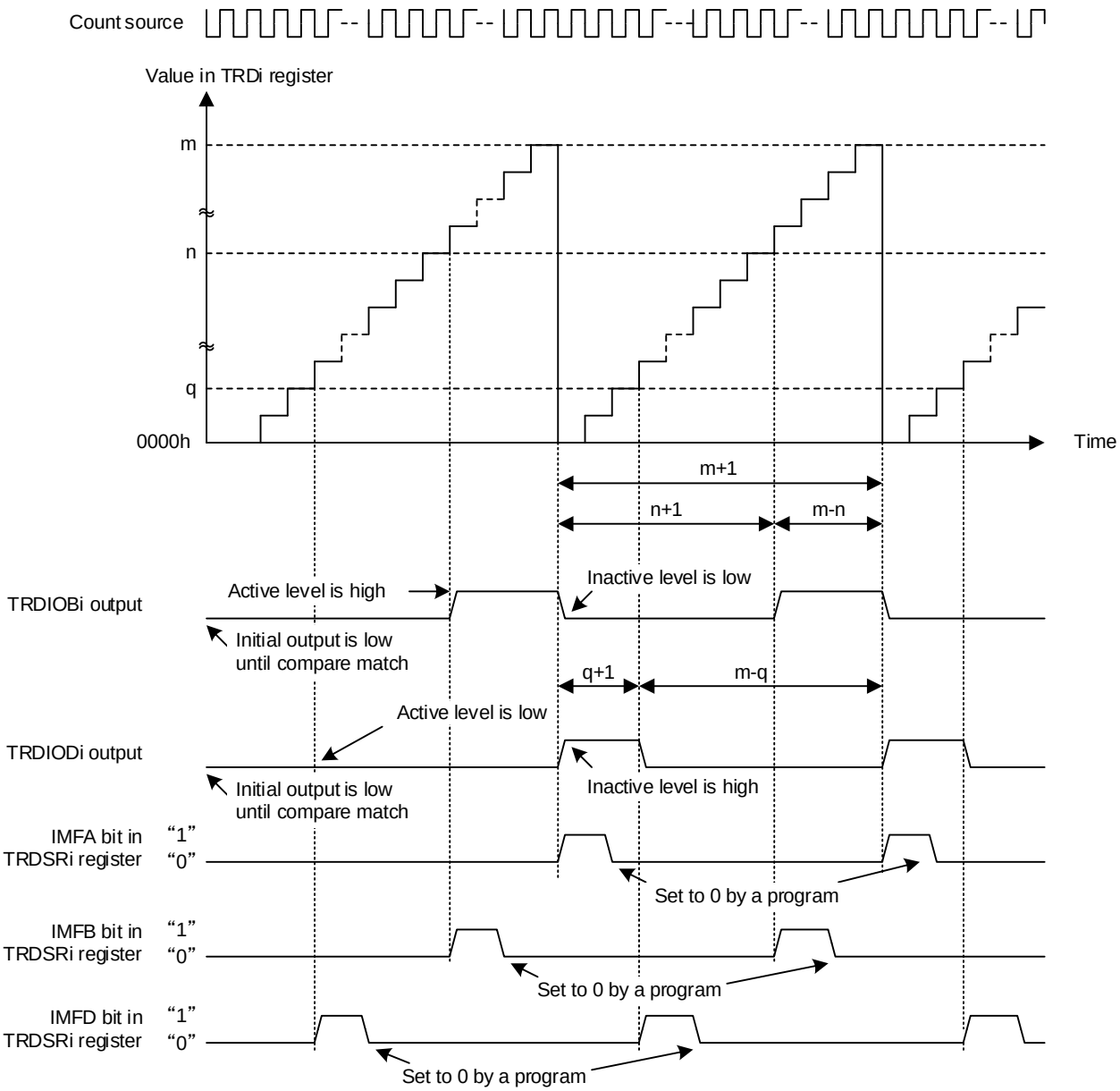
Item	Specification
Count sources **Note**	f_{CLK}, f_{PLL}, f_{IH}, f_{SUB}, f_{IL} External signal input to the TRDCLK0 pin (active edge selected by a program) Set bits TCK2 to TCK0 in the TRDCR1 register to the same value (same count source) as bits TCK2 to TCK0 in the TRDCR0 register.
Count operations	Increment
PWM waveform	PWM period: 1/fk × (m + 1) Active level width: 1/fk × (m − n) Inactive level width: 1/fk × (n + 1) fk: Frequency of count source m: Value set in the TRDGRAi register n: Value set in the TRDGRji register and TRDCMPqp register
	 (When low is selected as the active level)
Count start conditions	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. The PWM output pin holds the output level before the count stops. • When the CSELi bit in the TRDSTR register is set to 0, the count stops at compare match with the TRDGRAi register. 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) • TRDi register overflow
TRDIOA0 pin function	I/O port or TRDCLK0 (external clock) input
TRDIOA1, TRDIOC0, TRDIOC1 pin function	I/O port
TRDIOB0, TRDIOD0, TRDIOB1, 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	• Either one pin or multiple pins of TRDIOBi, TRDIODi can be selected as PWM output pin. • Active level selectable for each pin. • Initial output level selectable for each pin. • Synchronous operation (see **8.3.1 (3) Synchronous Operation**) • Pulse output forced cutoff signal input (see **8.3.1 (4) Pulse Output Forced Cutoff**) • Counter restart function can be selected by external input. • Dithering and gate function can be selected as PWM output.

Note When selecting the count source for the timer RDe, 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
p = 0 or 1, q = B or D

(1) Operation Example

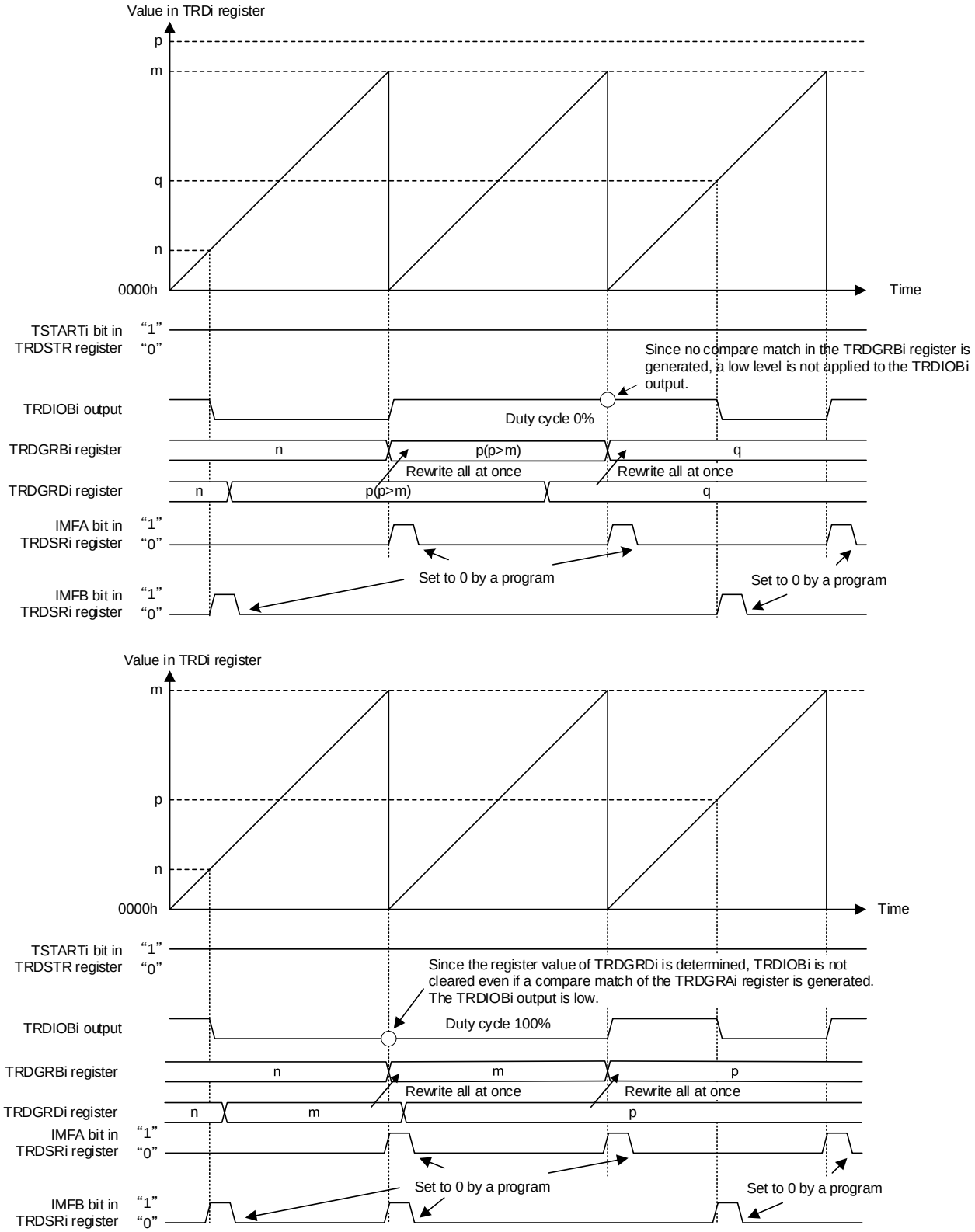
Figure 8-62. Operation Example in Extended PWM Mode



Remarks i = 0 or 1
m: Value set in TRDGRAi register
n: Value set in TRDGRBi register
q: Value set in TRDCMPBi register

The above diagram applies under the following conditions :
- Bits EBi and EDi in the TRDOER1 register are set to 0 (TRDIOBi and TRDIODi output enabled).
- Bit TOBi in the TRDOCR register is set to 0 (inactive level), the TODi bit is set to 1 (active level).
- Bit POLB in the TRDPOCRi register is set to 1 (active level is high), the POLD bit is set to 0 (active level is low).

Figure 8-63. Operation Example in Extended PWM Mode (Duty Cycle 0%, Duty Cycle 100%)



Remark i = 0 or 1
m: Value set in TRDGRAi register

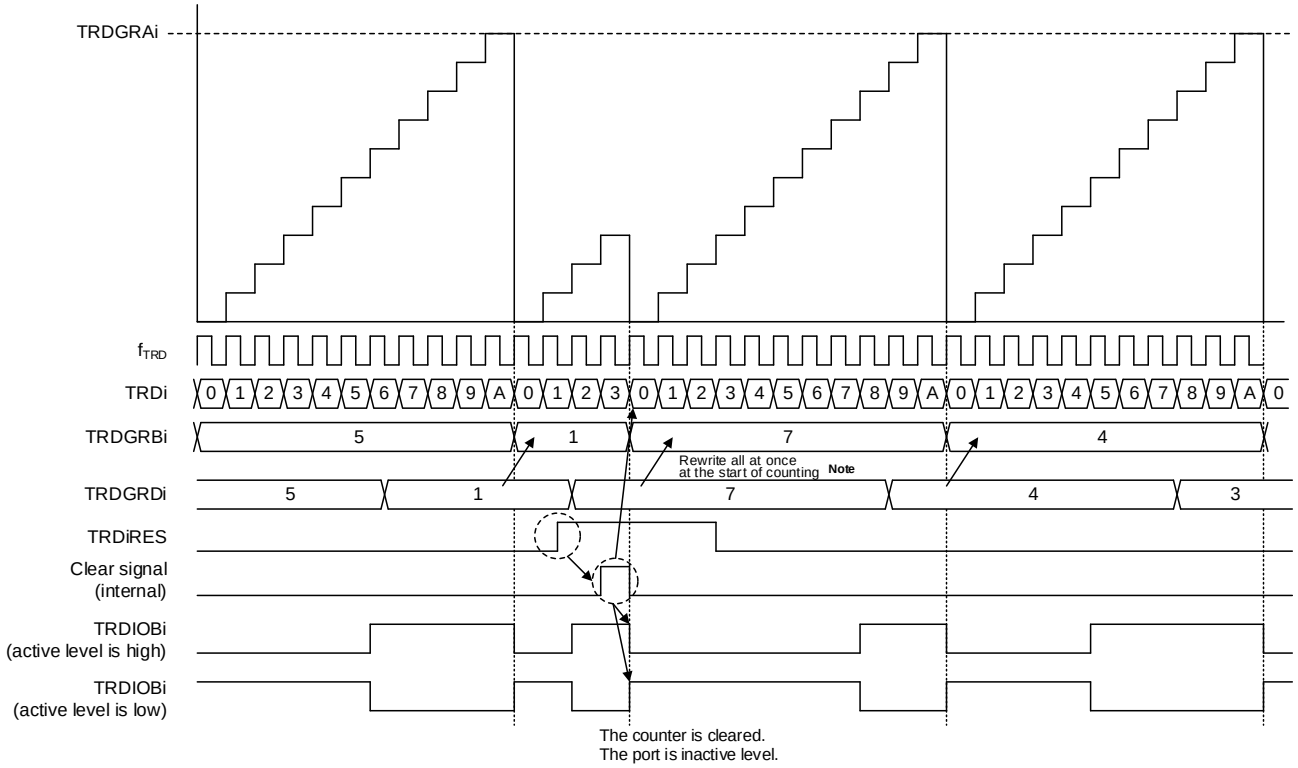
The above diagram applies under the following conditions :
- Bit EBi in the TRDOER1 register are set to 0 (TRDIOBi output enabled).
- Bit POLB in the TRDPOCRi register is set to 0 (active level is low).

(2) Counter Restart Function

In extended PWM mode, by using counter restart function, timer RDi counter (TRDi) (i = 0 or 1) can be initialized by the input of TRDiRES and the count can be started again. Select the input signal (TRDiRES) with CCLV1 and CCLV0 bits in the TRDEMRi register. The interrupt signal is not output when the counter is cleared by TRDiRES input.

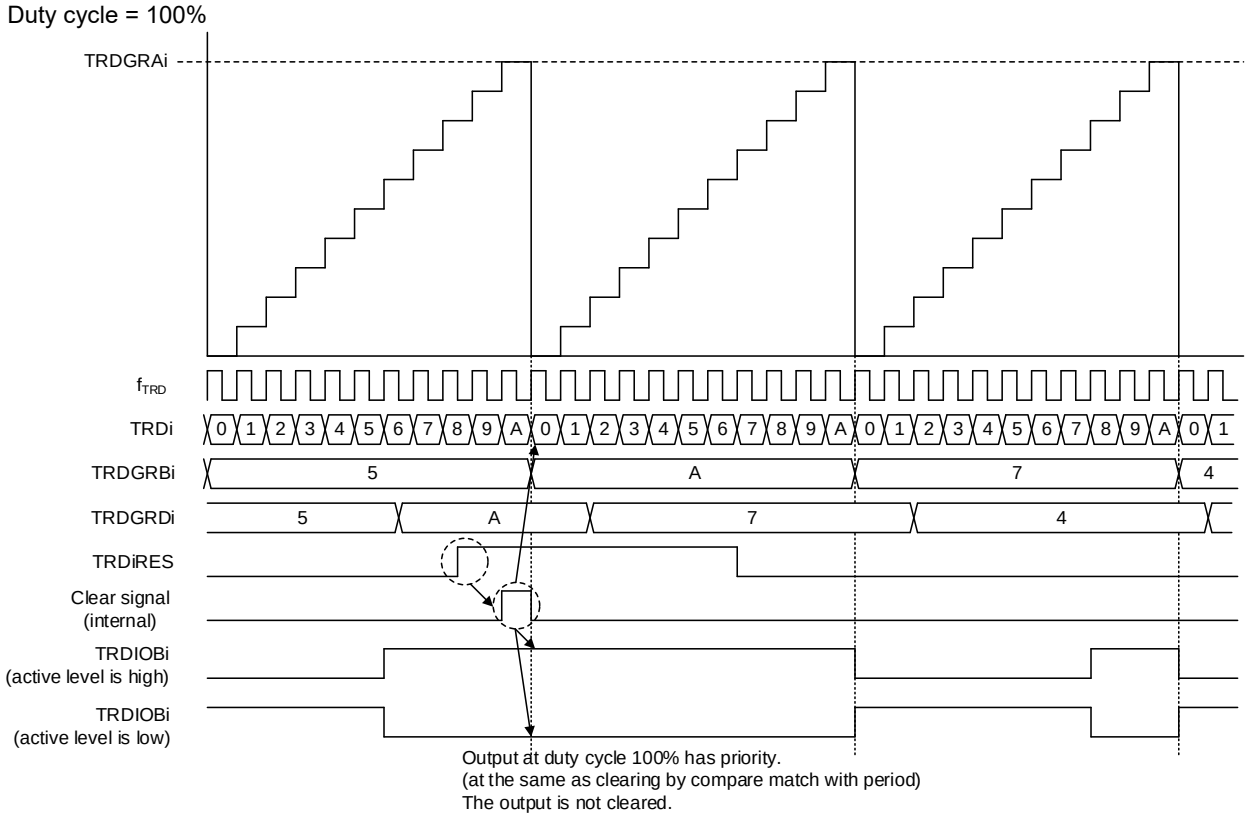
Figure 8-64 shows the Operation Example of Counter Restart Function when the counter is cleared by the rising edge.

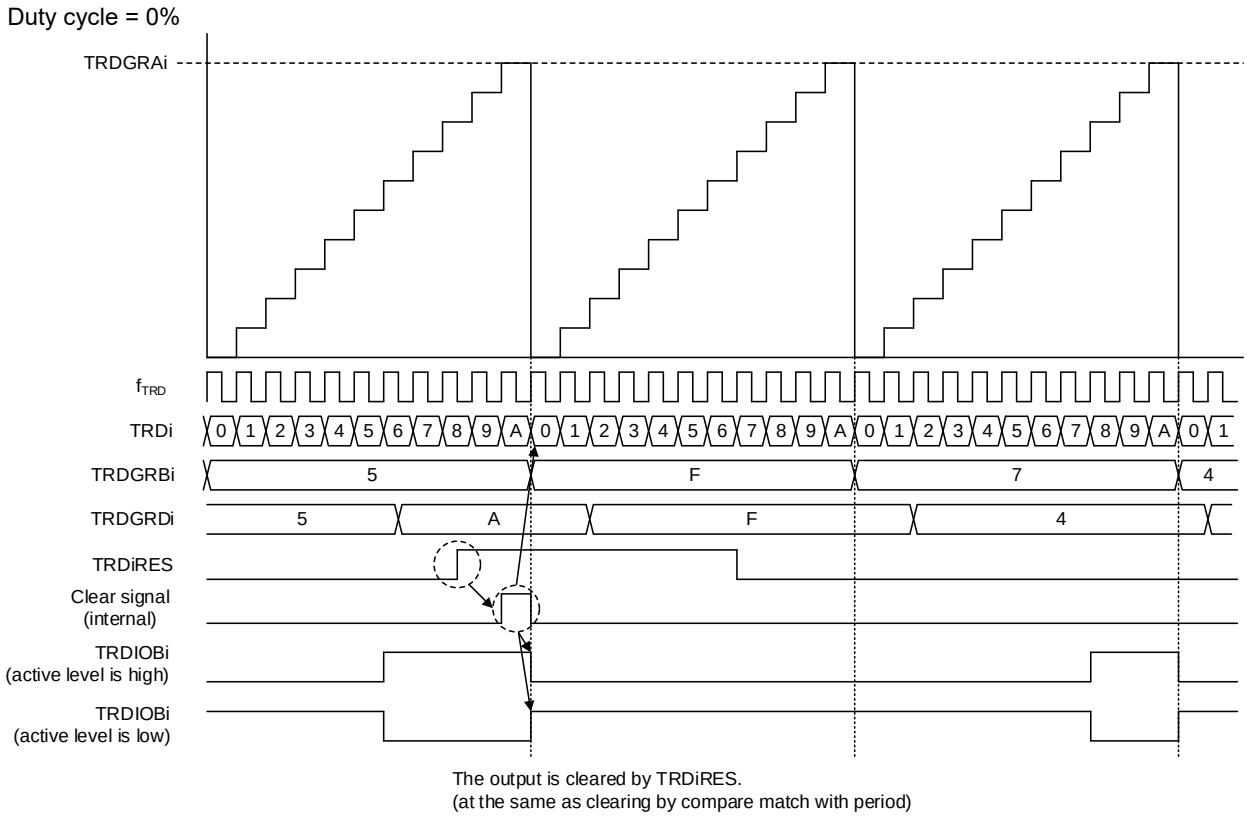
Figure 8-64. Operation Example of Counter Restart Function



Note When the RSF bit in the TRDRSFi register is set to 1.

Figure 8-65. Operation Example of Counter Restart Function (Duty Cycle 0%, Duty Cycle 100%)





(3) Dithering Function

In extended PWM mode, dithering function can be used for the output of TRDIOBi and TRDIODi (i = 0 or 1). The average resolution can be improved by extending the active pulse width of the selected period by 1 count based on the 16 periods of PWM waveform period. Select the period to extend the active pulse width by 1 count by DNR3 to DNR0 bits setting in the TRDDNRi register.

The figure below shows the PWM output operation of dithering function.

DTEN	DNR[3:0]	PWM Output Period															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0000B																
	0001B	■															
	0010B	■								■							
	0011B	■				■				■							
	0100B	■				■				■				■			
	0101B	■		■		■				■				■			
	0110B	■		■		■				■		■		■			
	0111B	■		■		■		■		■		■		■			
	1000B	■		■		■		■		■		■		■		■	
	1001B	■	■	■		■		■		■		■		■		■	
	1010B	■	■	■		■		■		■	■	■		■		■	
	1011B	■	■	■		■	■	■		■	■	■		■		■	
	1100B	■	■	■		■	■	■		■	■	■		■	■	■	
	1101B	■	■	■	■	■	■	■		■	■	■		■	■	■	
	1110B	■	■	■	■	■	■	■		■	■	■	■	■	■	■	
	1111B	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
0	–																

■ : The PWM period during which one clock is added to the active pulse.

□ : The PWM output set in registers. (TRDGRji, TRDCMPm)

Remarks DTEN: Bits DTENB and DTEND in the TRDEMRi register
DNR[3:0]: Bits in the TRDDNRi register
i = 0 or 1
j = A, B, C, or D
m = B0, D0, B1 or D1

Figure 8-66 and **Figure 8-67** shows Operation Exmple of Dithering Function.

Figure 8-66. Operation Example of Dithering Function

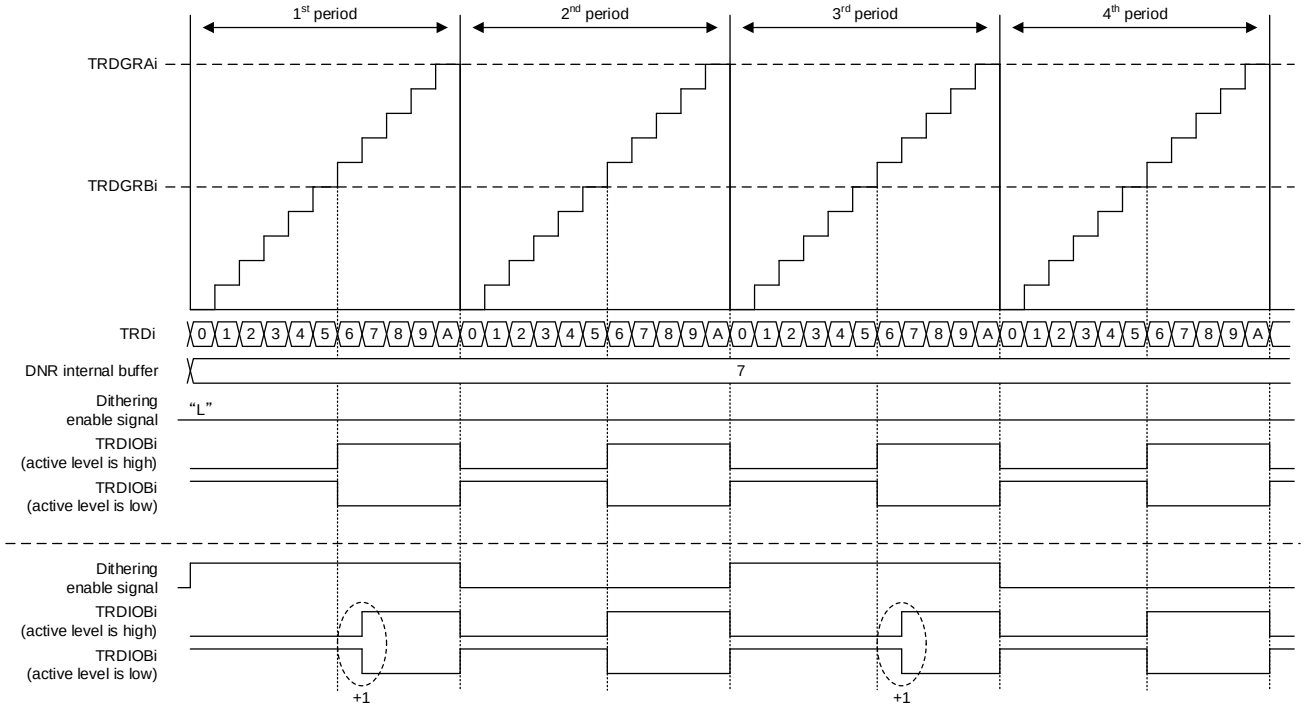
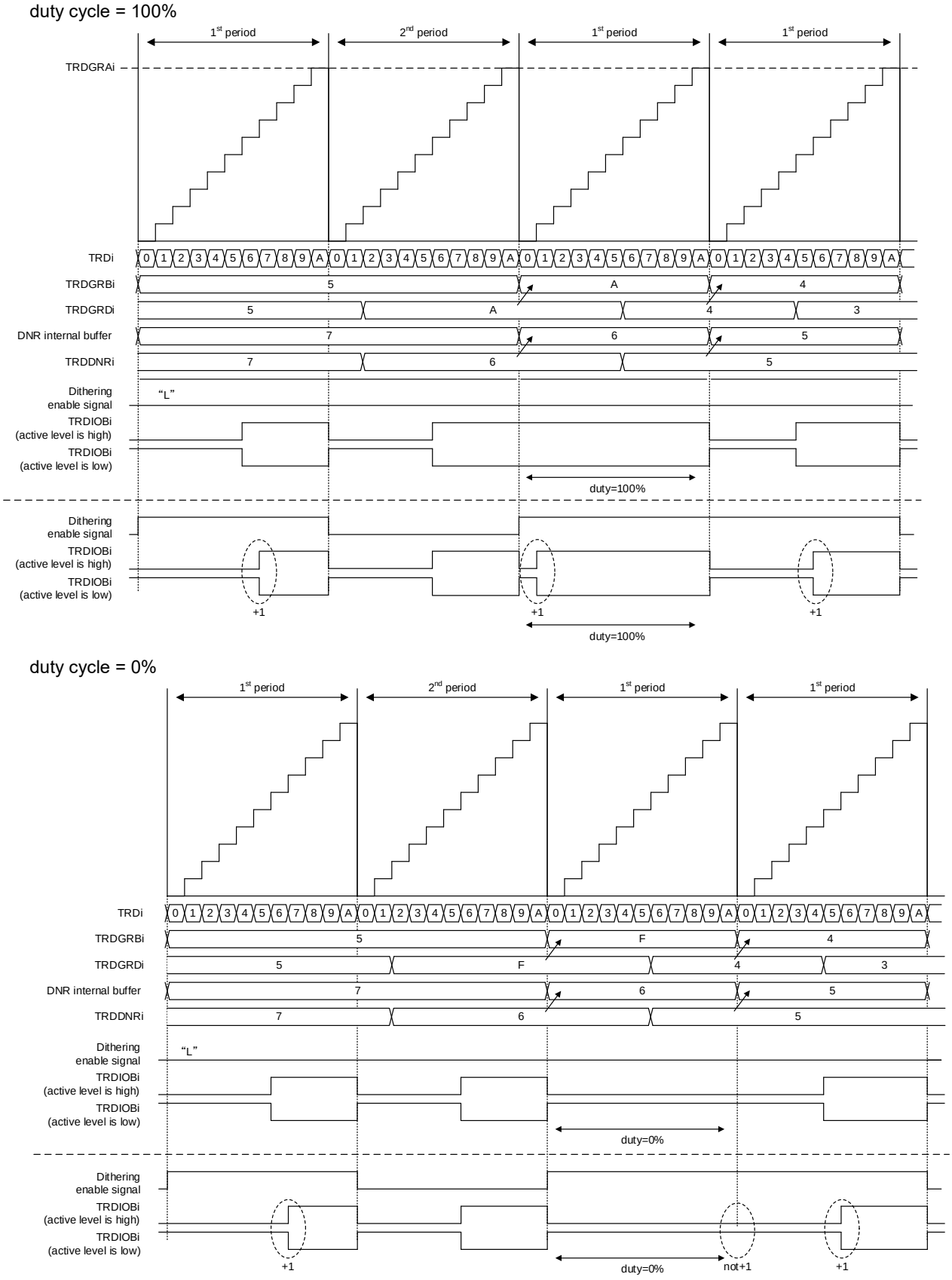


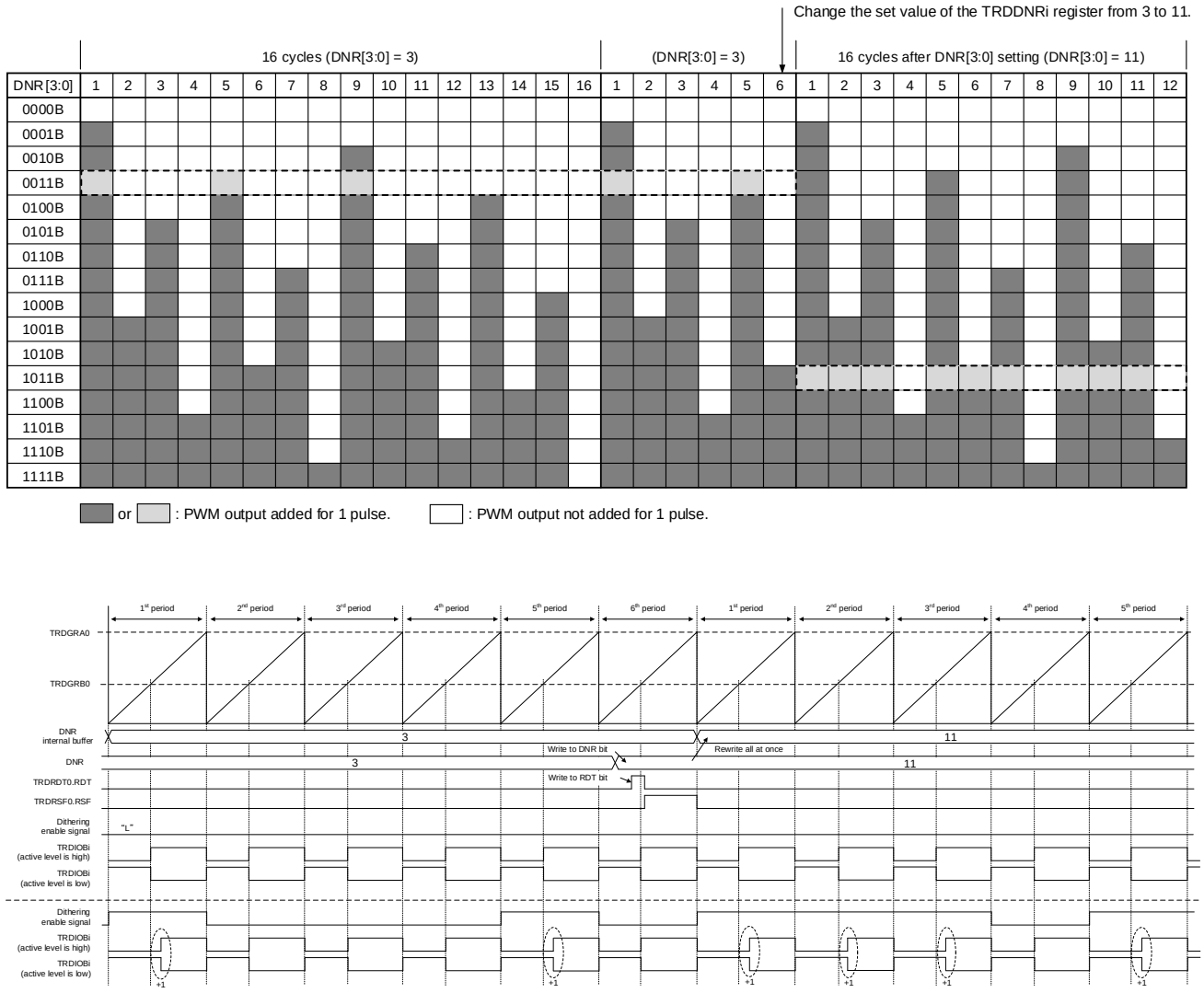
Figure 8-67. Operation Example of Dithering Function (Duty Cycle 0%, Duty Cycle 100%)



The TRDDNRi register (i = 0 or 1) can be rewritten during counting operation. If rewritten in the middle of 16 periods, the rewritten value becomes valid when TRDi register and TRDGRAi register match.

Figure 8-68 shows Operation Example of Dithering Function (Change the Setting Value from 3 to 11).

Figure 8-68. Operation Example of Dithering Function (Change the Setting Value from 3 to 11)



(4) Gate Function

In extended PWM mode, gate function can be used for the output of TRDIOBi and TRDIODi (i = 0 or 1). The outputs of TRDIOBi and TRDIODi are gated in n periods based on 16 periods. Select the gated period by GPR3 to GPR0 bits setting in the TRDGPRi register.

The figure below shows the PWM output operation of gate function.

When the GPAT bit in the TRDGPRi register is set to 0.

GTEN	GPR[3:0]	PWM Output Period															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0000B																
	0001B	■															
	0010B	■	■														
	0011B	■	■	■													
	0100B	■	■	■	■												
	0101B	■	■	■	■	■											
	0110B	■	■	■	■	■	■										
	0111B	■	■	■	■	■	■	■									
	1000B	■	■	■	■	■	■	■	■								
	1001B	■	■	■	■	■	■	■	■	■							
	1010B	■	■	■	■	■	■	■	■	■	■						
	1011B	■	■	■	■	■	■	■	■	■	■	■					
	1100B	■	■	■	■	■	■	■	■	■	■	■	■				
	1101B	■	■	■	■	■	■	■	■	■	■	■	■	■			
	1110B	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
	1111B	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
0	–	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

When the GPAT bit in the TRDGPRi register is set to 1.

GTEN	GPR[3:0]	PWM Output Period															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0000B		■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
	0001B			■	■	■	■	■	■	■	■	■	■	■	■	■	■
	0010B				■	■	■	■	■	■	■	■	■	■	■	■	■
	0011B					■	■	■	■	■	■	■	■	■	■	■	■
	0100B						■	■	■	■	■	■	■	■	■	■	■
	0101B							■	■	■	■	■	■	■	■	■	■
	0110B								■	■	■	■	■	■	■	■	■
	0111B									■	■	■	■	■	■	■	■
	1000B										■	■	■	■	■	■	■
	1001B											■	■	■	■	■	■
	1010B												■	■	■	■	■
	1011B													■	■	■	■
	1100B														■	■	■
	1101B															■	■
	1110B																■
	1111B																
0	–	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

■ : The PWM output of this cycle is not masked.

□ : The PWM output of this cycle is masked. (Gate function enables)

Remarks GTEN: Bits GTENB and GTEND in the TRDEMRi register

GPR[3:0]: Bits in the TRDGPRi register

Figure 8-69 shows Operation Example of Gate Function when GPAT bit is set to 0. PWM output waveform is gated according to the value set in GPR3 to GPR0 bits.

Figure 8-69. Operation Example of Gate Function

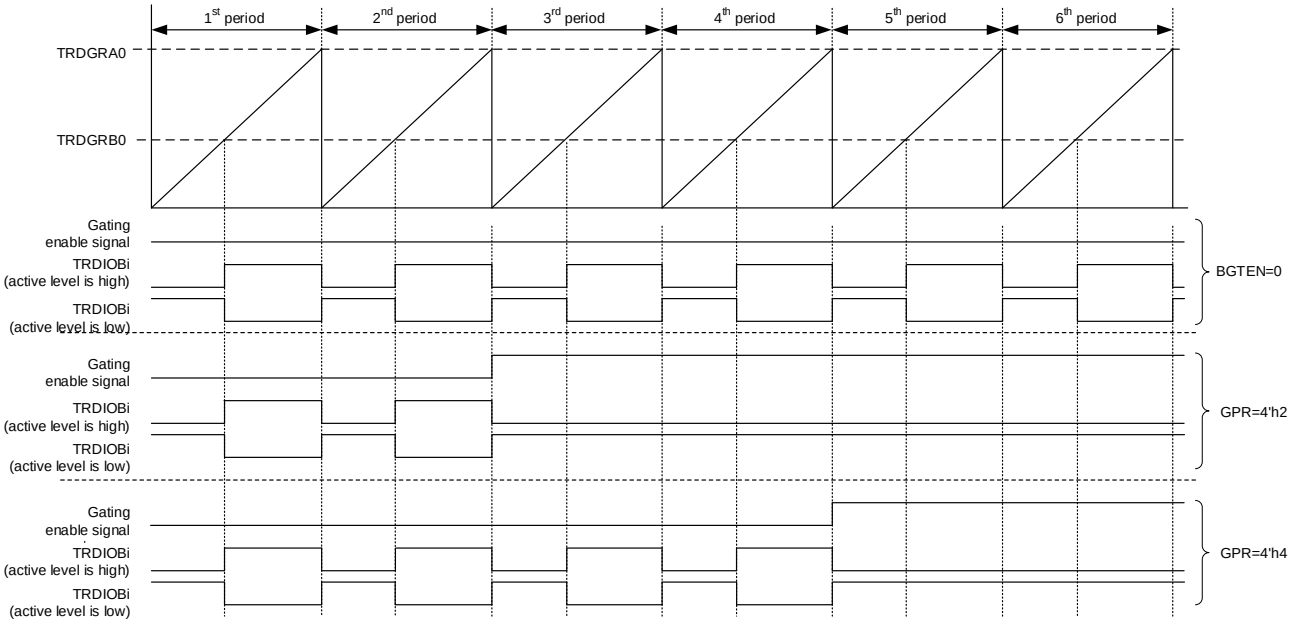
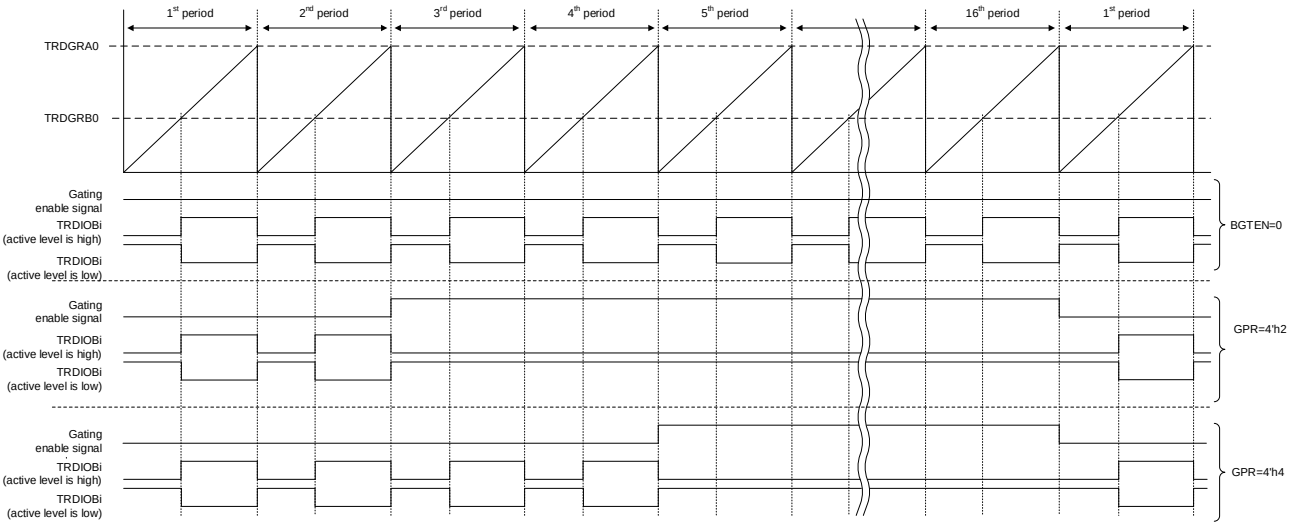


Figure 8-70 shows PWM output waveform after 16 periods.

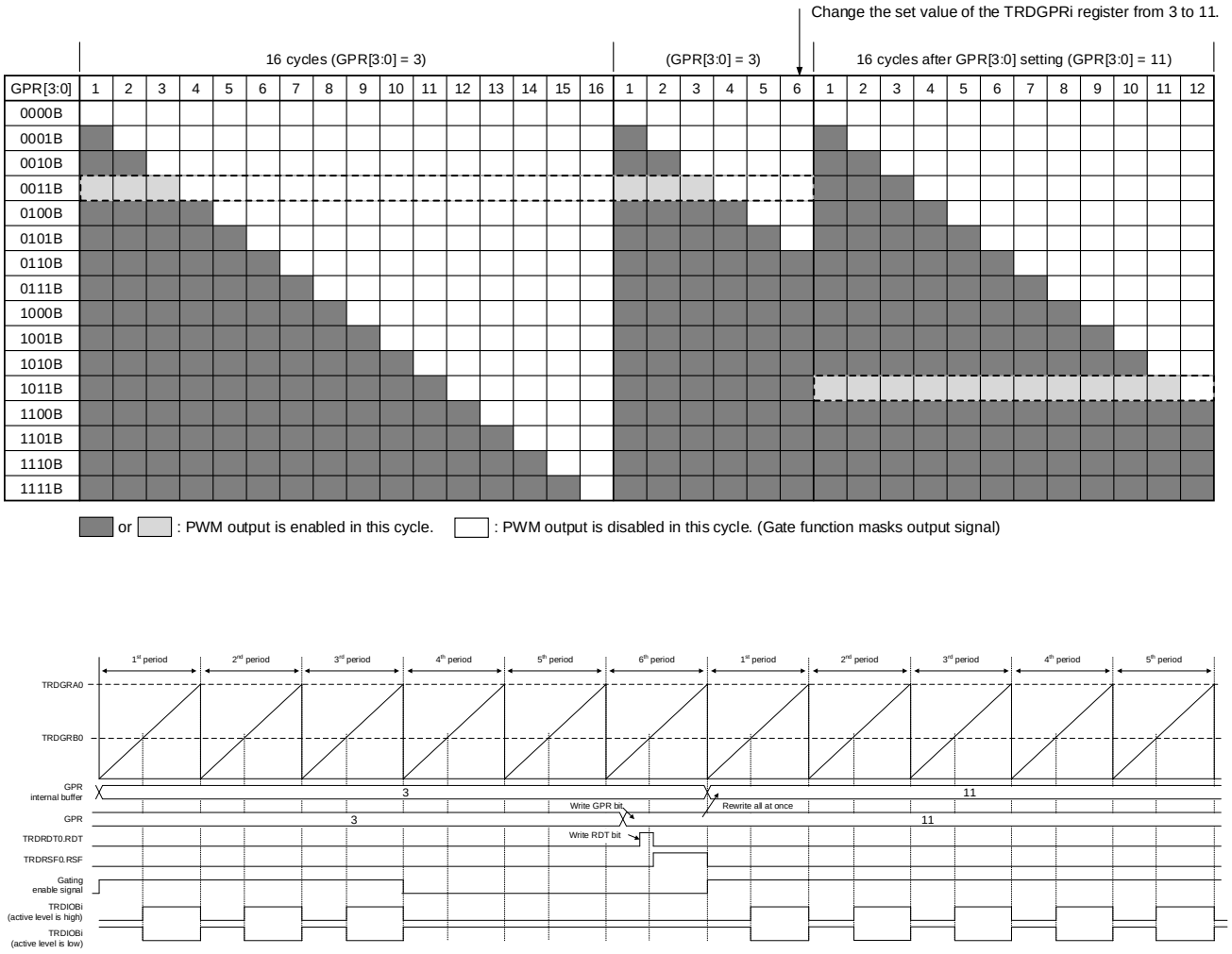
Figure 8-70. Operation Example of Gate Function After 16 Cycles



The TRDGPRi register (i = 0 or 1) can be rewritten during counting operation. If rewritten in the middle of 16 periods, the rewritten value becomes valid when TRDi register and TRDGRAi register match.

Figure 8-71 shows Operation Exmple of Dithering Function (Change the Setting Value from 3 to 11).

Figure 8-71. Operation Example of Gate Function (Change the Setting Value from 3 to 11)



8.3.9 Extended Complementary PWM Mode

In this mode, three normal-phases and three counter-phases of the symmetric or asymmetric PWM waveform are output with the same period (three-phase, triangular wave modulation, and with dead time).

Figure 8-72 shows the Block Diagram of Extended Complementary PWM Mode, **Table 8-26** lists the Extended Complementary PWM Mode Specifications, and **Figure 8-73**, **Figure 8-74**, and **Figure 8-75** show an Operation Example in Extended Complementary PWM Mode.

Figure 8-72. Block Diagram of Extended Complementary PWM Mode

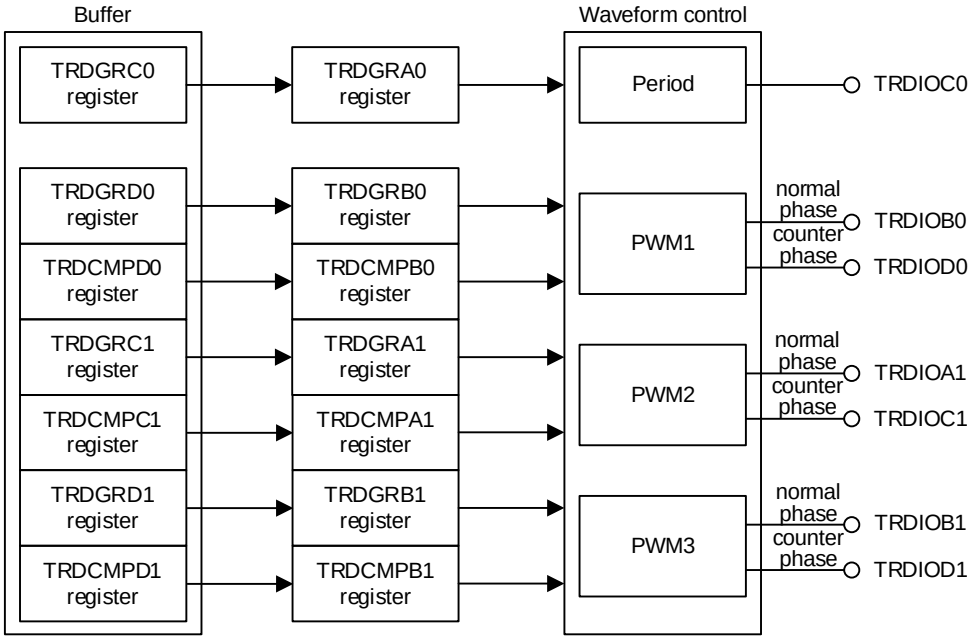
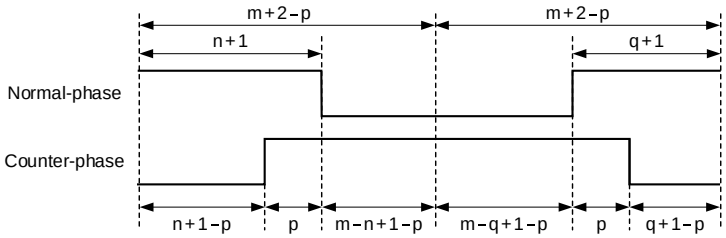


Table 8-26. Extended Complementary PWM Mode Specifications (1/2)

Item	Specification
Count sources Note 1	f_{CLK}, f_{PLL}, f_{IH}, f_{SUB}, f_{IL} External signal input to the TRDCLK0 pin (active edge selected by a program) 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. 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	Complementary PWM period : 1/fk × (m + 2 − p) × 2 Note 2 Dead time : p Active level width of normal-phase : 1/fk × {(m − n + 1 − p) + (m − q + 1 − p)} Active level width of counter-phase : 1/fk × {(n + 1 − p) + (q + 1 − p)} fk: Frequency of count source m: Value set in the TRDGRA0 register n: Value set in the TRDGRB0 register (PWM1 output) Value set in the TRDGRA1 register (PWM2 output) Value set in the TRDGRB1 register (PWM3 output) q: Value set in the TRDCMPB0 register (PWM1 output) Value set in the TRDCMPA1 register (PWM2 output) Value set in the TRDCMPB1 register (PWM3 output) p: Value set in the TRD0 register  (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	• TRD1 register underflow • Compare match with TRD0 and TRDGRA0 For details, see **8.7 Interrupt Decimation Module (TRDMBK)**.
TRDIOA0 pin function	I/O port or TRDCLK0 (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)

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

Table 8-26. Extended Complementary PWM Mode Specifications (2/2)

Item	Specification
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**) • The normal-phase and counter-phase active level and initial output level are selected individually. • Transfer timing from the buffer register selection • A/D trigger function (see **8.3.9 (2) A/D trigger generation**)

Notes 1. When selecting the count source for the timer RDe, 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 complementary PWM period is fixed.

Remark i = 0 or 1, j = A, B, C, or D

(1) Operation example

The initial output level of the TRDIOji pin (j=A, B, C, D, i=0, 1) in extended complementary PWM mode sets the inactive level to the TRDOCR register for both normal-/counter-phases. After the count starts, the active level set in bits OLS1 and OLS0 of the TRDFCR register is enabled.

If the first period after the count starts is 100% of the duty cycle, set the counter-phase to the active level to the TRDOCR register.

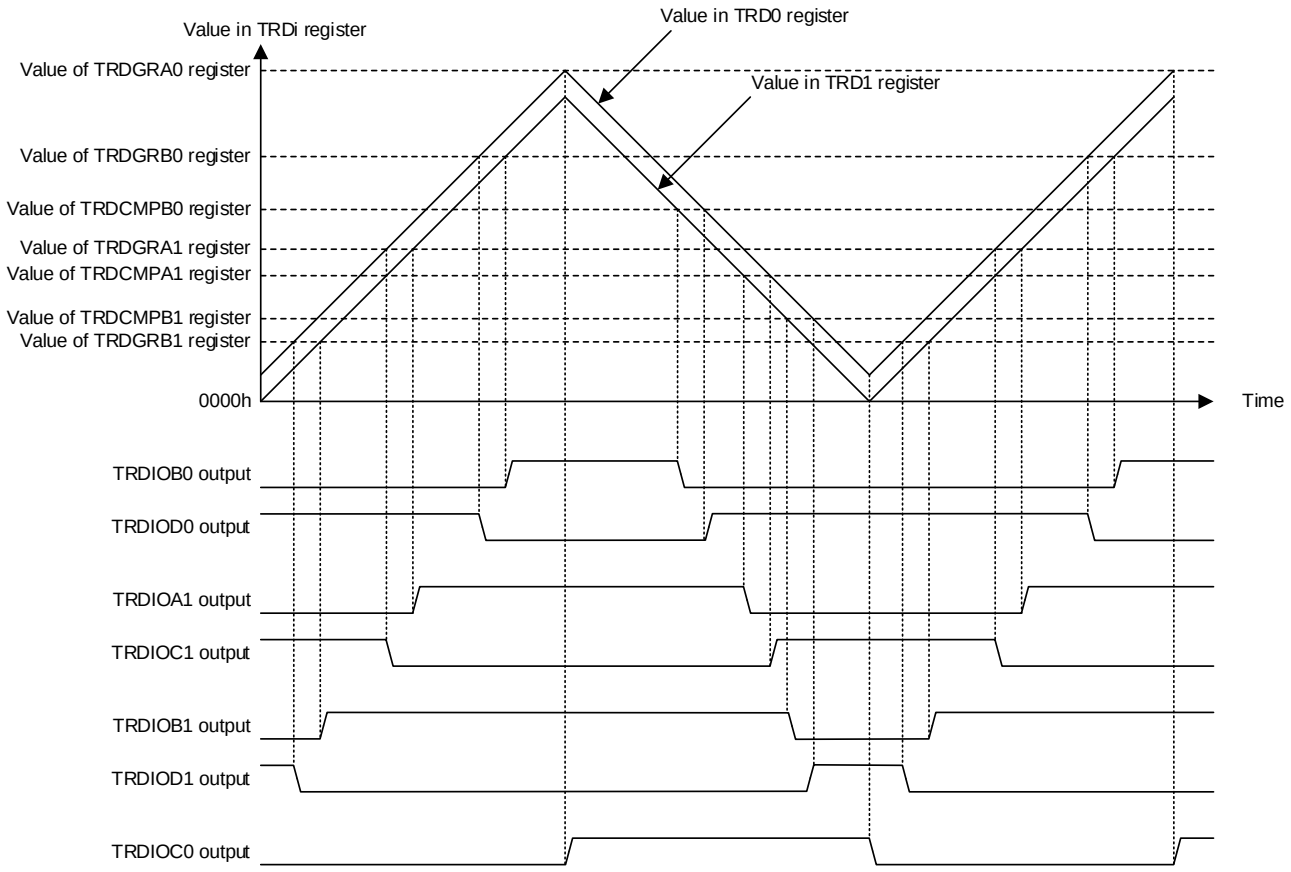
When buffer registers are transferred:

Extended complementary PWM mode uses the register reload function to update the general registers. Set the RDT bit in the TRDRDT1 register to 1 after setting the buffer registers. Transfers from the buffer register to the general register when the TRD1 counter underflows.

For details, see **8.3.1 (7) Register Reload Function**.

Remarks General registers: TRDGRB0, TRDGRA1, TRDGRB1, TRDCMPB0, TRDCMPA1, TRDCMPB1
Buffer registers: TRDGRD0, TRDGRC1, TRDGRD1, TRDCMPD0, TRDCMPC1, TRDCMPD1

Figure 8-73. Asymmetric Waveform Model in Extended Complementary PWM Mode

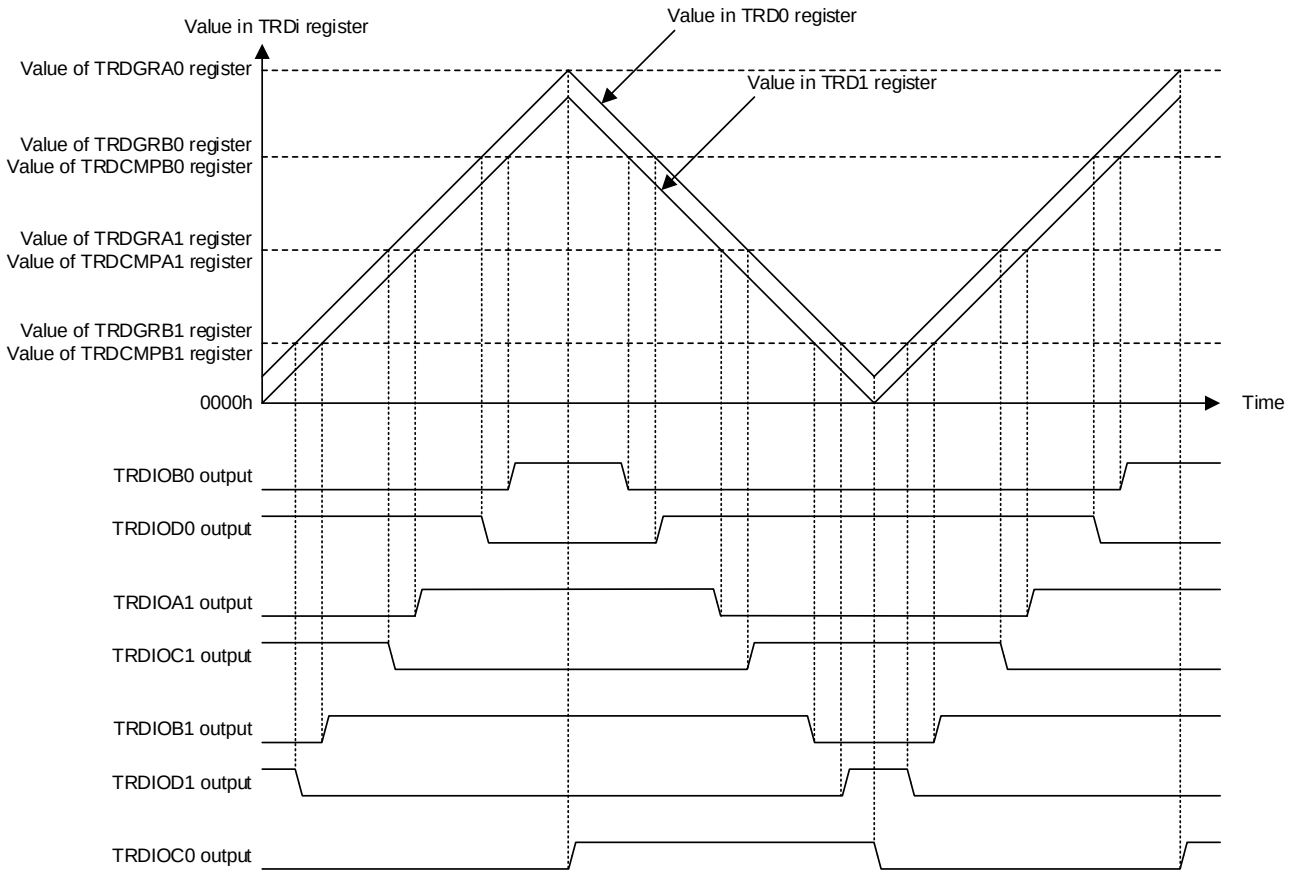


Remark i = 0 or 1

The above diagram applies under the following condition:

- TRDFCR.CPSS=1

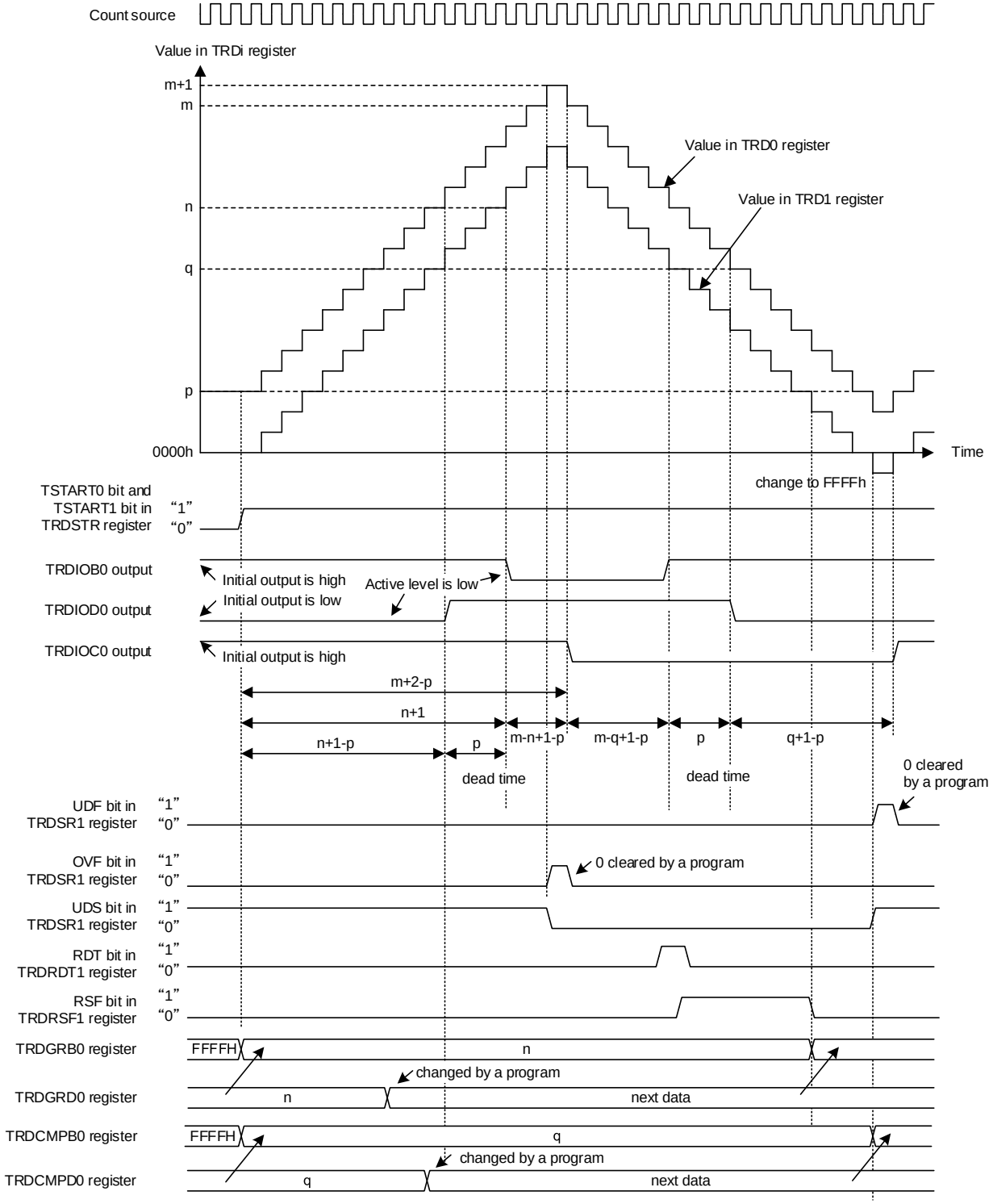
Figure 8-74. Symmetric Waveform Model in Extended Complementary PWM Mode for Asymmetric PWM Output



Remark i = 0 or 1

The above diagram applies under the following condition:
- TRDFCR.CPSS=1

Figure 8-75. Operation Example in Extended Complementary PWM Mode (Asymmetric Triangular Waveform PWM Output)



Remark i = 0 or 1
m: Value set in TRDGRA0 register
n: Value set in TRDGRB0 register
q: Value set in TRDCMPB0 register
p: Value set in TRD0 register

The above diagram applies under the following conditions :
- Bits OLS1 and OLS0 in TRDFCR register are set to 0 (active level is low in both of normal phase and counter phase)
- TRDOCR.TOB0=1, TRDOCR.TOD0=0, TRDOCR.TOC0=1

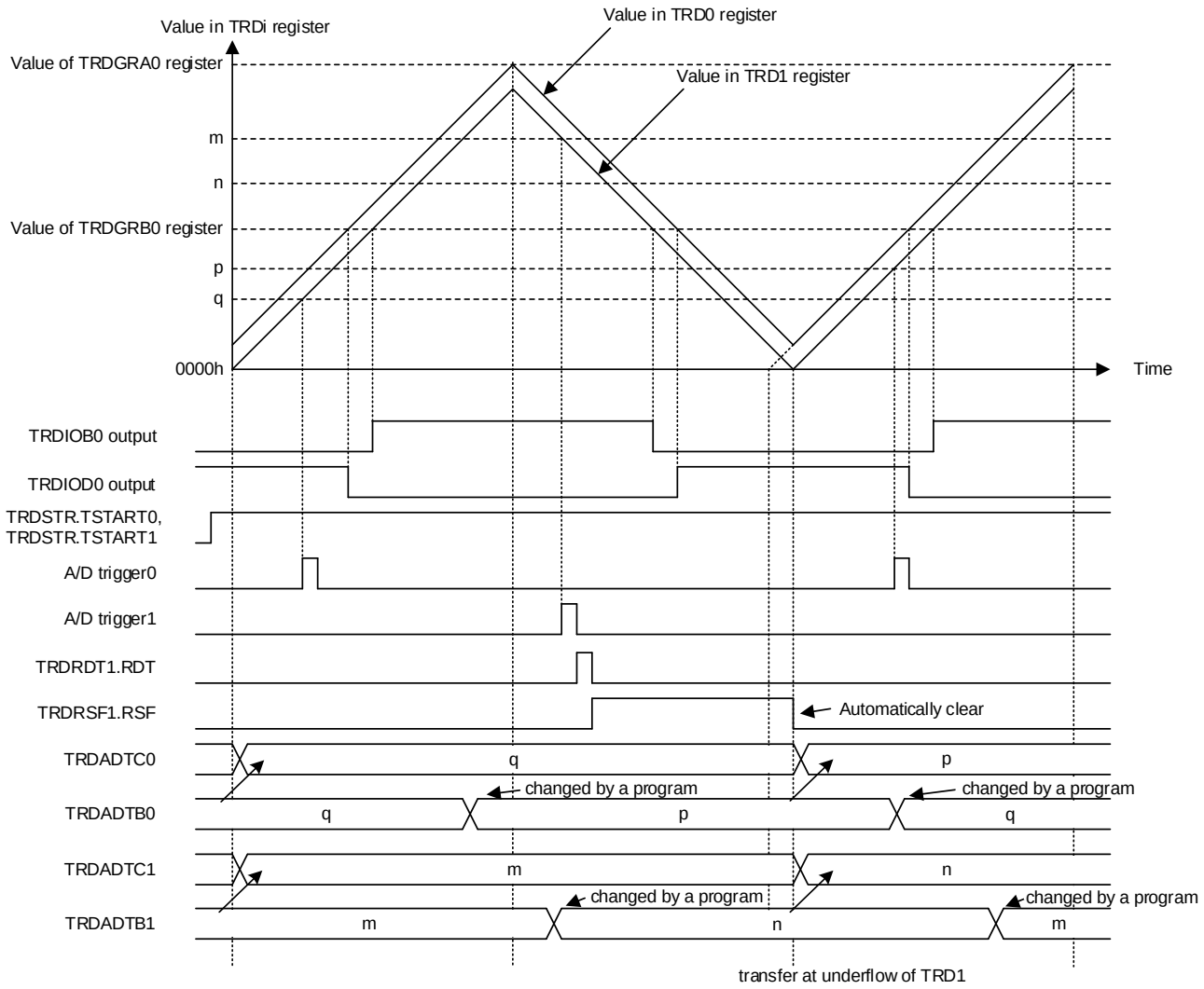
(2) A/D trigger generation

In extended complementary PWM mode, an A/D conversion trigger can be generated when the TRD1 counter and the TRDADTCi register (i=0 or 1) match, either during increment or decrement.

Select whether the ADMDi bit of the TRDADCR register is incrementing or decrementing, and enable this function with the ADEi bit in the TRDADCR register set to 1.

It is used to update data from the TRDADTBi register (buffer register) to the TRDADTCi register (general register) using register reload function.

Figure 8-76. A/D Trigger Generation Timing



Remark The above diagram applies under the following conditions:

- TRDADCR.ADMD0 = 0
- TRDADCR.ADMD1 = 1

8.4 Timer RDe Interrupt

Timer RDe generates the timer RDi (i = 0 or 1) interrupt request from six sources for each timer RD0 and timer RD1.

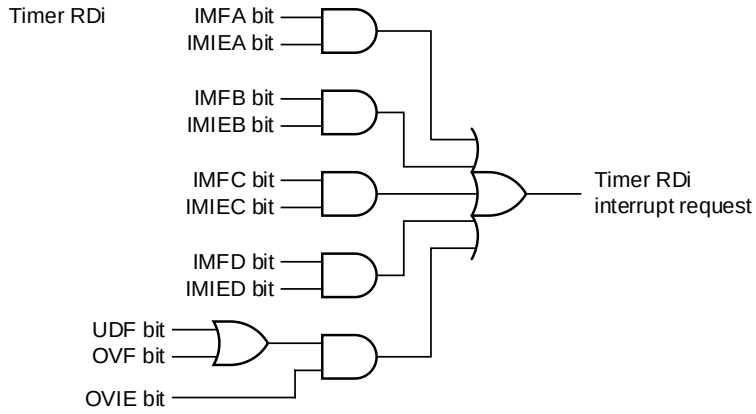
Table 8-27 lists the Registers Associated with Timer RDe Interrupt and

Figure 8-77 shows the Timer RDe Interrupt Block Diagram. The interrupt function of extended complementary PWM mode is different from the **Figure 8-77**. For details, see **8.7 Interrupt Decimation Module (TRDMBK)**.

Table 8-27. Registers Associated with Timer RDe Interrupt

	Timer RDe Status Register	Timer RDe 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) TRDPR10 (PR10H)
Timer RD1	TRDSR1	TRDIER1	TRDIF1 (IF0H)	TRDMK1 (MK0H)	TRDPR01 (PR00H) TRDPR11 (PR10H)

Figure 8-77. Timer RDe Interrupt Block Diagram



Remark i = 0 or 1
IMFA, IMFB, IMFC, IMFD, OVF, UDF: TRDSRi register bit
IMIEA, IMIEB, IMIEC, IMIED, OVIE: TRDIERi register bit

Since the interrupt source (timer RDe interrupt) is generated by a combination of multiple interrupt request sources for timer RDe, 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 at the same time.
- 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 RDe

8.5.1 SFR Read/Write Access

The timer RDe SFRs are undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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 RDe, set the TRD0EN bit in the PER1 register to 1 first. If the TRD0EN bit is 0, writes to the timer RDe 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 **Note**, TRDPTO bit in TRDOER2, TRDDFi, TRDCRi, TRDIORAi, TRDIORCi, TRDPOCRi, TRDOCR, TRDEMRi

Note. When the setting value of the corresponding bits in PWMDLY0 register is other than 00B (no output delay).

(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 8-28** 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 RDe output.

Table 8-28. TRDIOji 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 RDe complementary and reset synchronous PWM modes. In extended complementary PWM mode, outputs the initial output level set in TRDOCR register.)
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 RDe complementary and reset synchronous PWM modes. In extended complementary PWM mode, outputs the initial output level set in TRDOCR register.)

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 22 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 RDe, 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 RDe operating clock (f_{TRD}).
- The value of the TRDi register is transferred to the TRDGRji register two to three cycles of the timer RDe 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 TSTARTi bit in the TRDSTR register is 0 (count stops) if the edge selected by bits TRDIOj0 and TRDIOj1 in registers TRDIORAi and TRDIORCi are 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 RDe 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:

[Changing 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 RDe 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.

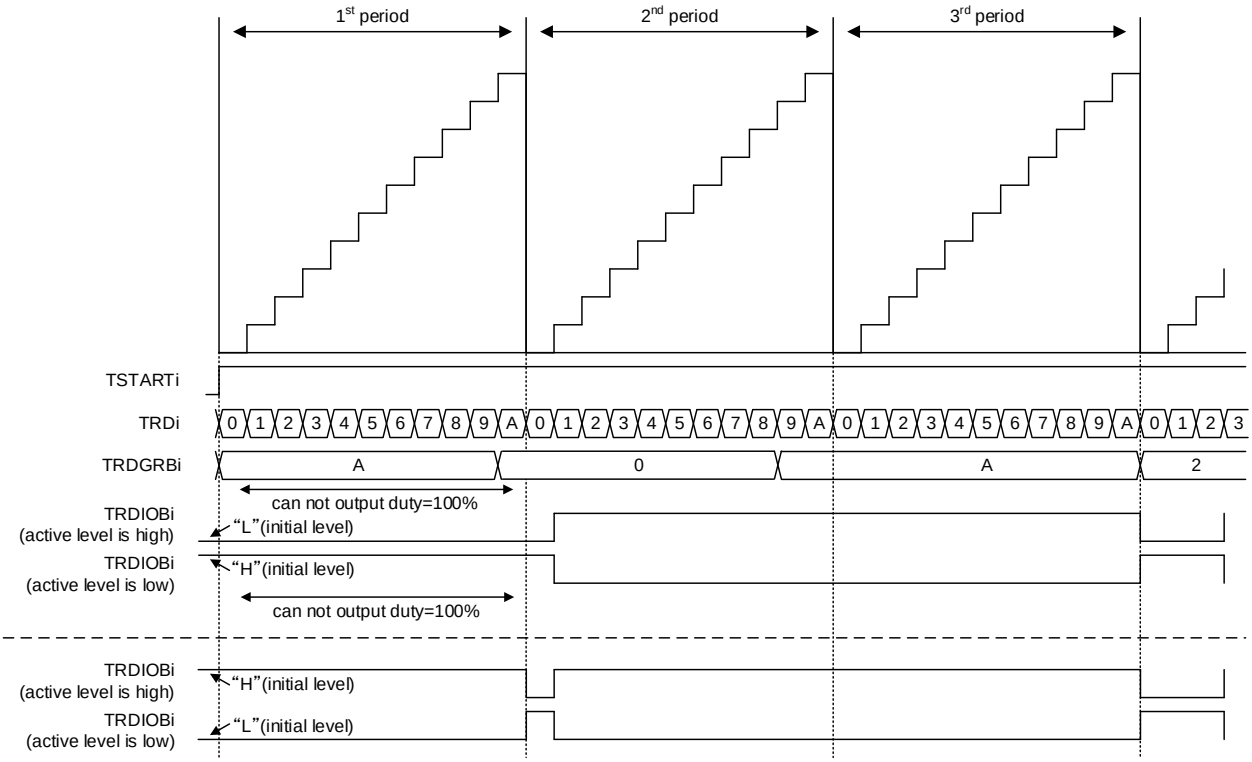
8.5.6 External Clock TRDCLK0

Set the pulse width of the external clock applied to the TRDCLK0 pin to three or more cycles of the timer RDe operating clock (f_{TRD}).

8.5.7 Timer Mode PWM Function

When initializing the general register that sets the PWM output, do not set it to the same value as the PWM cycle. If the initial output level of the terminal is set to the inactive level, 100% duty cannot be output in the cycle immediately after the start of counting operation.

Figure 8-78. Timer Mode PWM Function Limitation of Duty100% Output After Start of Counting



8.5.8 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 RDe again.

8.5.9 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.

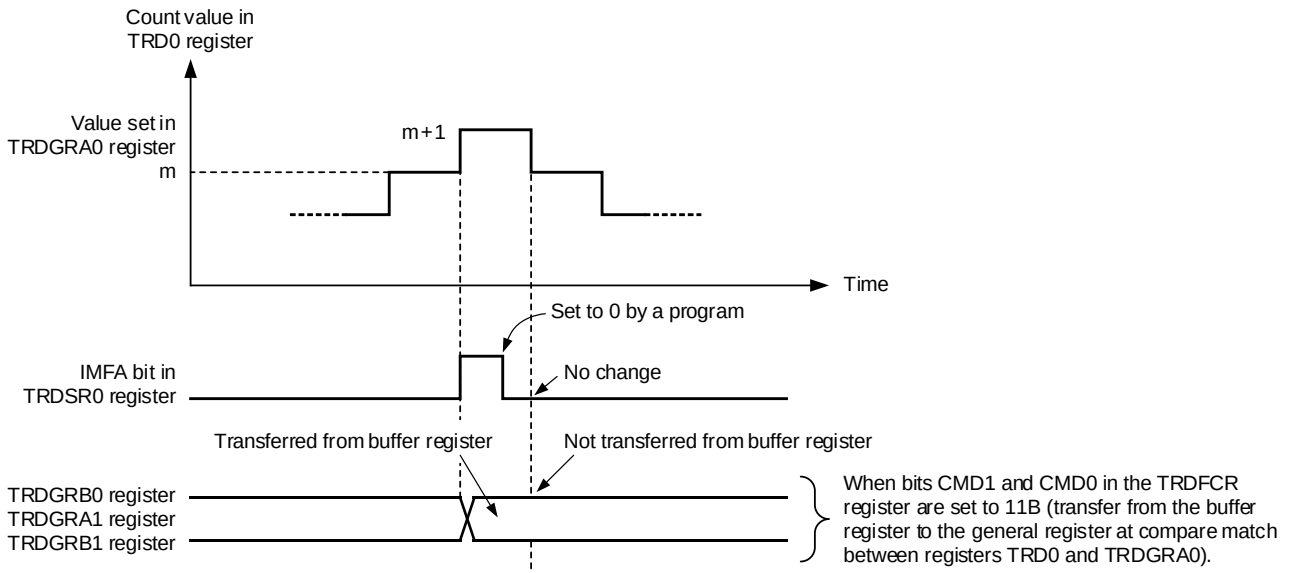
(1) Set both the TSTART0 and TSTART1 bits in the TRDSTR register to 0 (count stops).
(2) Set bits CMD1 and CMD0 in the TRDFCR register to 00B (timer mode, PWM mode, and PWM3 mode).
(3) Set bits CMD1 and CMD0 to 10B or 11B (complementary PWM mode).
(4) Set the registers associated with other timer RDe 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).

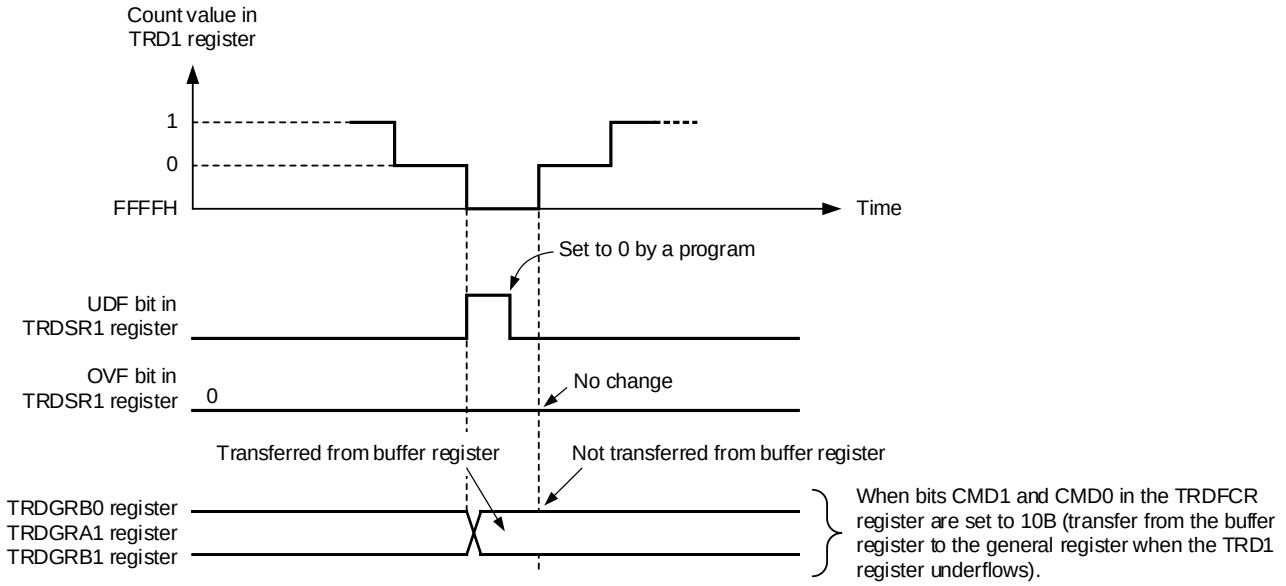
- Do not write to the TRDGRA0, TRDGRB0, TRDGRA1, or TRDGRB1 register during operation.
  When changing the PWM waveform, transfer the values written to registers TRDGRD0, TRDGRC1, and TRDGRD1 to registers TRDGRB0, TRDGRA1, and TRDGRB1 using the buffer operation.
  However, to write data to the TRDGRD0, 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 8-79. 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 8-80. Operation When TRD1 Register Underflows in Complementary PWM Mode

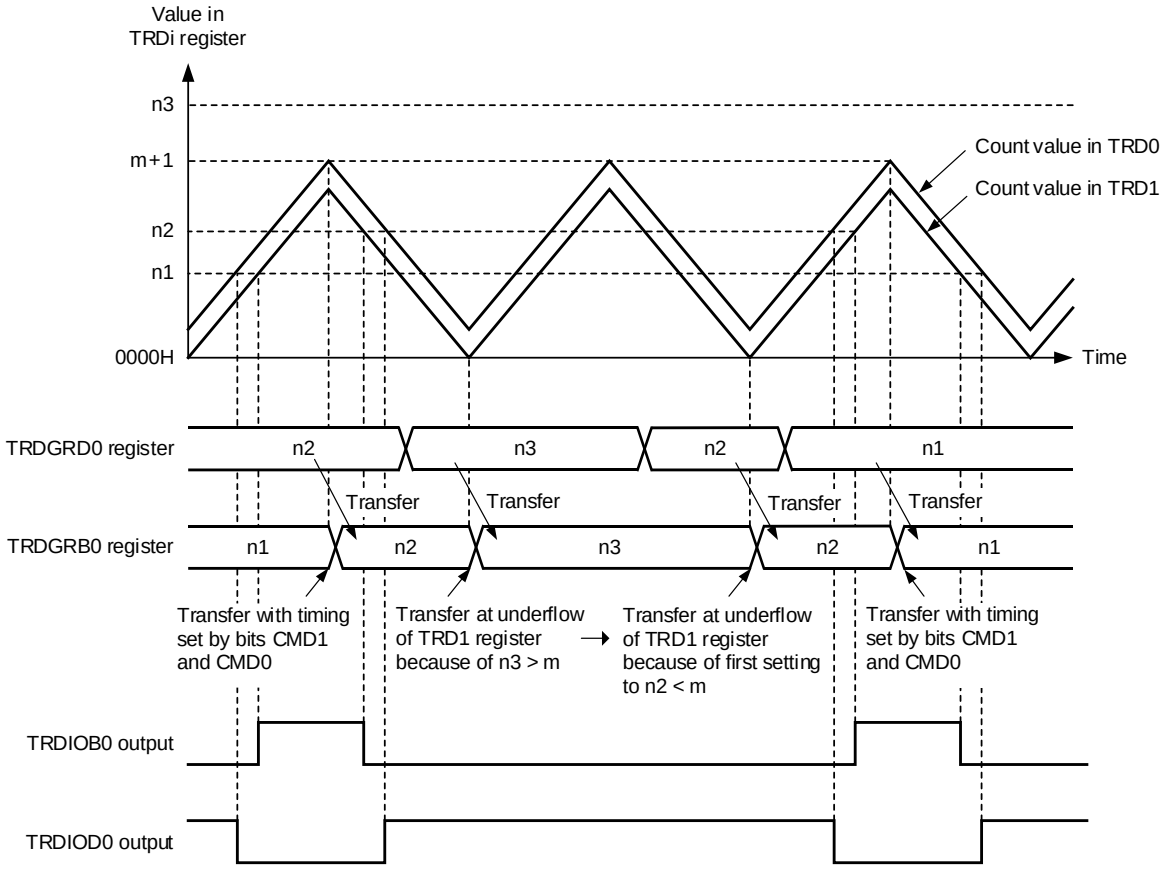


- The timing of data transfer from the buffer register to the general register should be selected using bits CMD0 and CMD1 in the TRDFCR register. However, regardless of the values of bits CMD0 and CMD1, transfer takes place with the following timing when duty cycle is 0% and duty cycle is 100%.

(a) 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 8-81. Operation When Value in Buffer Register ≥ Value in TRDGRA0 Register in Complementary PWM Mode



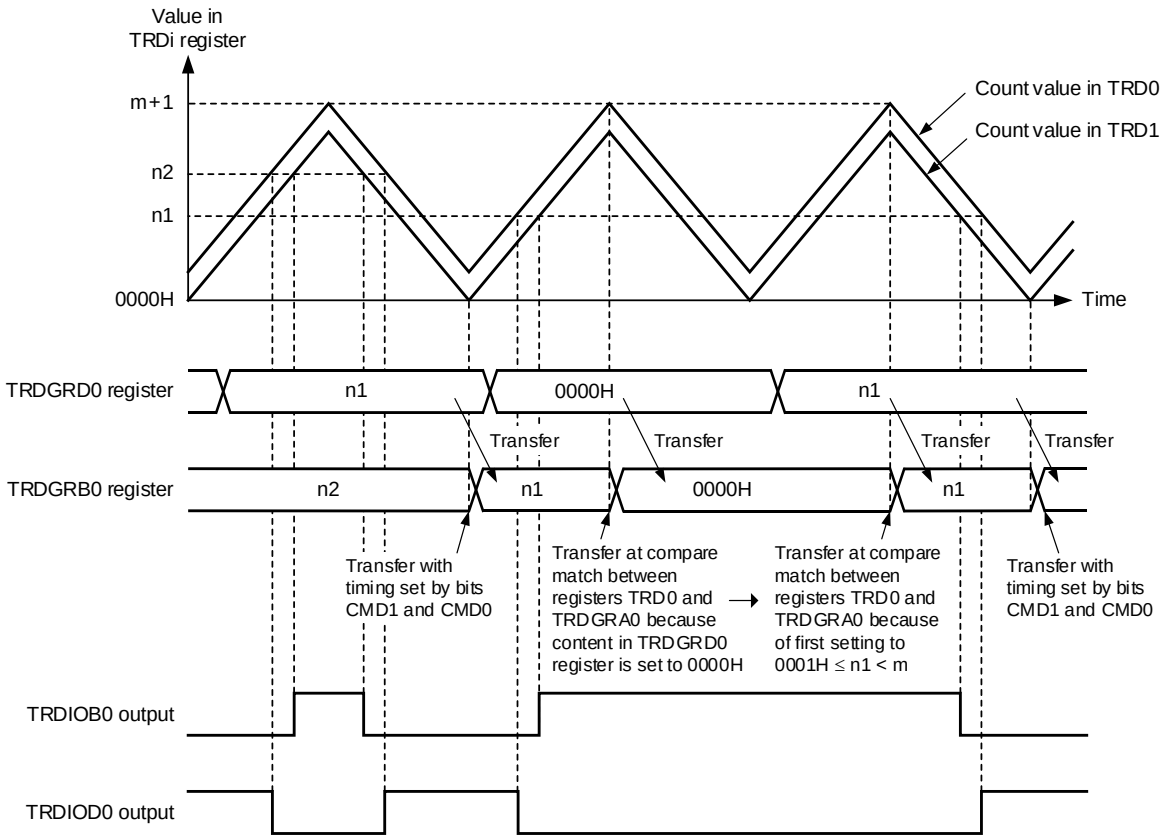
Remark m: Value set in TRDGRA0 register

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).

(b) 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 8-82. Operation When Value in Buffer Registers 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).

8.5.10 Extended PWM Mode

- To stop and restart the counting operation, follow the procedure below.
  (1) Set the TSTARTi bit of the TRDSTR register to “0” (count stop).
  (2) Reset the initial output level of the pin with the TRDOCR register.
  (3) Set the TRDi register to 0000h.
  (4) Set the TSTARTi bit of the TRDSTR register to “1” (count start).

- When using synchronous operation, follow the procedure below.
  (1) Set the TRDMR register TRDSYNC bit to "1". Note
  (2) Set the same value in the TRDCR0 register and TRDCR1 register.
  Set the TRDi register to 0000h.
  Set the same value for TRDGRA0 and TRDGRA1.
  (3) Set the TSTARTi bit of the TRDSTR register to “1” (count start).
  When rewriting the compare register by register reload function, set TRDRDT0.RDT and TRDRDT1.RDT at the same time with 16-bit access.

  Note When using the count / restart function in synchronous operation, TRD0 and TRD1 are cleared by inputting TRD0RES. The input for TRD1RES is ignored.

- When initializing the general register that sets the PWM output, do not set it to the same value as the PWM cycle.
  If the initial output level of the terminal is set to the inactive level, 100% duty cannot be output in the cycle immediately after the start of counting operation. To output 100% duty from the first cycle, set the initial output level of the terminal to the active level. See **8.5.7 Timer Mode PWM Function** for details.

- The pulse width of the signal input to the TRD0RES pin or TRD1RES pin should be at least 3 cycles of the timer RDe operating clock.

- When using the first cycle with the gate enabled, set the initial output level of the terminal to the inactive level.

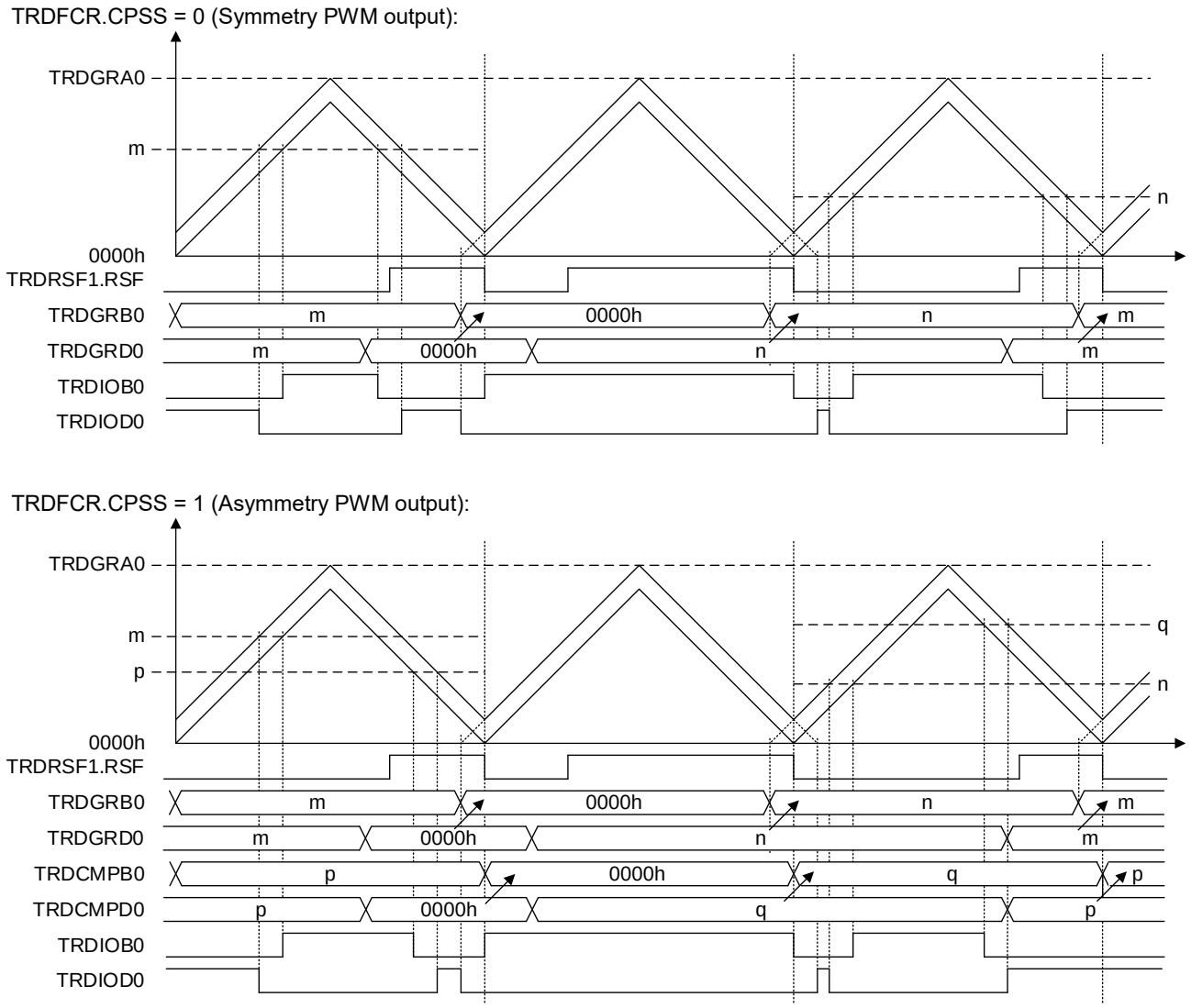
- When using dithering function, follow the procedure below.
  (1) Set the TSTARTi bit of the TRDSTR register to “0” (count stop).
  (2) Reset the initial output level of the pin with the TRDOCR register.
  (3) Set the TRDi register to 0000h.
  (4) Set the DNR[3:0] bit in the TRDDNRi register to “0000B” (initialize dithering number).
  (5) Set the TSTARTi bit of the TRDSTR register to “1” (count start).

8.5.11 Extended Complementary PWM Mode

- Duty cycle 100% setting

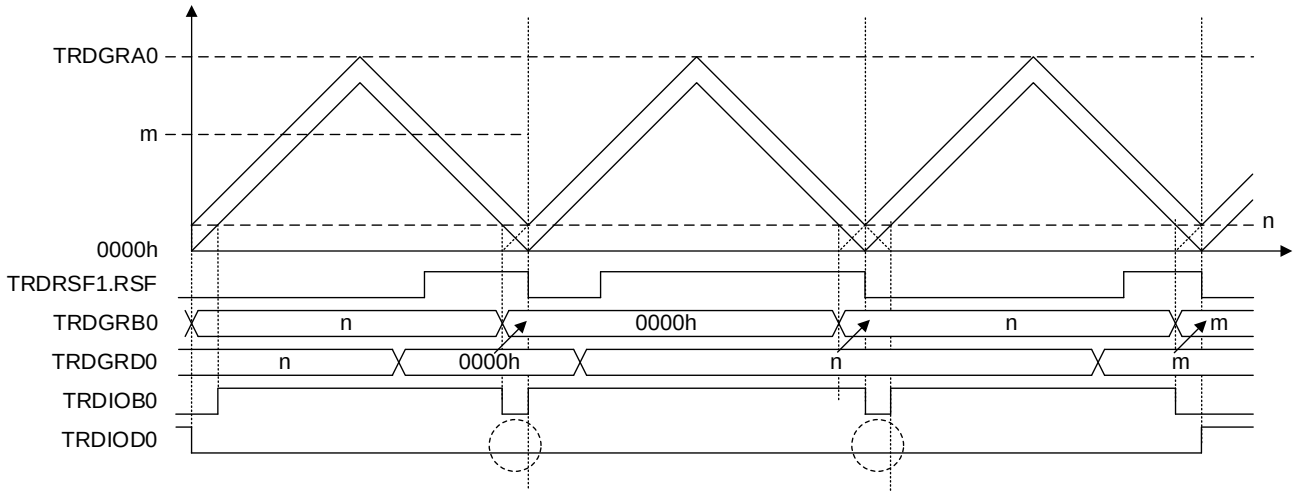
  When setting duty cycle 100%, set 0000H to related compare register. If the CPSS bit in the TRDFCR register is set to 0, only the TRDGRji register can be set. If the CPSS bit is set to 1, set the TRDGRji register and the TRDCMPm register. If buffer registers are set by the reload timing and the RSF bit in the TRDRSF1 register is 1, duty cycle 100% can be output.

Figure 8-83. Operation Example (Duty Cycle 100%)



- Notes on duty cycle 100% setting
  Depending on the PWM duty ratio and dead-time settings, the counter-phase has the same inactive level output as when the PWM duty is 100%. Note that TRDIOC0 is also affected by the setting of dead-time in the same way as the counter-phase.

Figure 8-84. Notes on Duty Cycle 100% Setting



Remark n: dead time

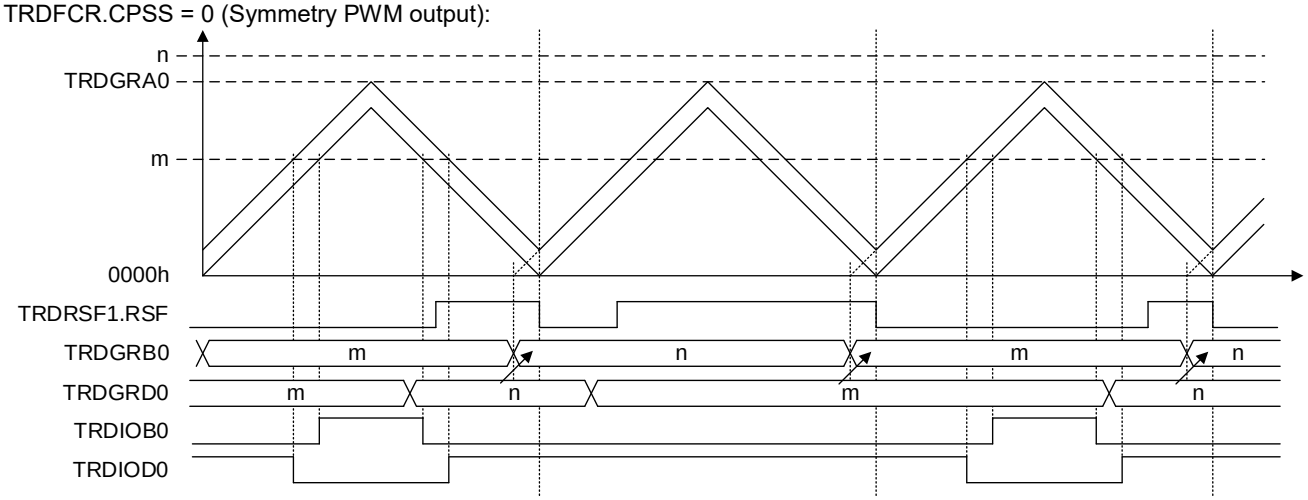
The above diagram applies under the following condition:
- TRDFCR.CPSS=0

- Duty cycle 0% setting

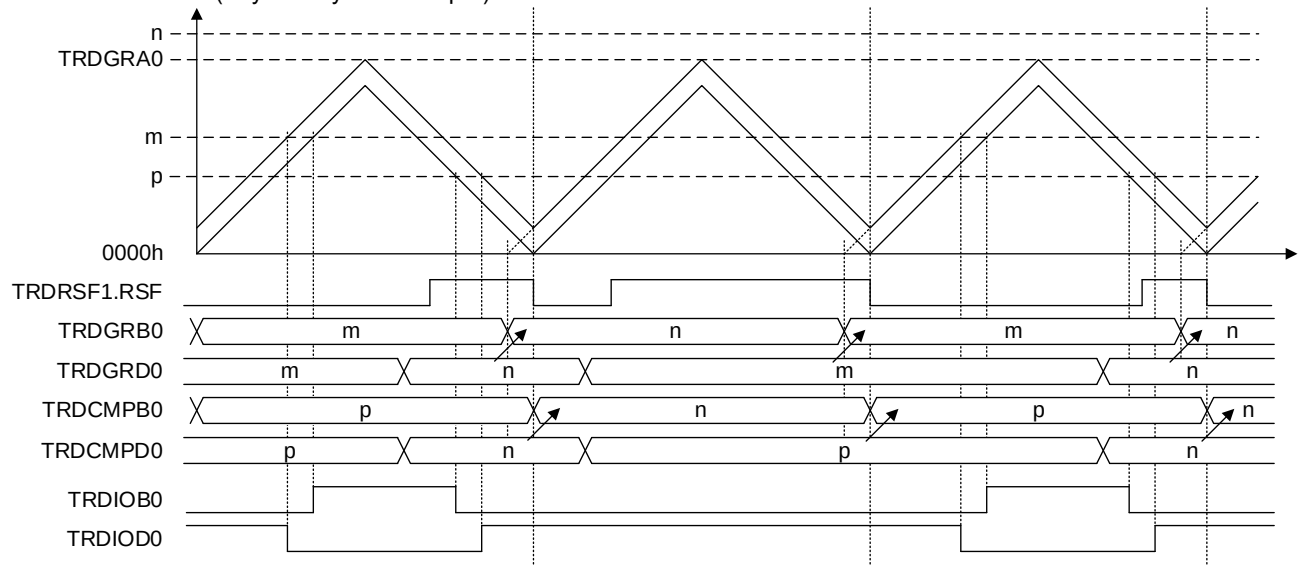
  When setting duty cycle 0%, set the related compare register to a value greater than or equal to the setting value of the TRDGRA0 register. If the CPSS bit in the TRDFCR register is set to 0, only the TRDGRji register can be set. If the CPSS bit is set to 1, set the TRDGRji register and the TRDCMPm register. If buffer registers are set by the reload timing and the RSF bit in the TRDRSF1 register is 1, duty cycle 0% can be output.

Figure 8-85. Operation Example (Duty Cycle 0%)

TRDFCR.CPSS = 0 (Symmetry PWM output):

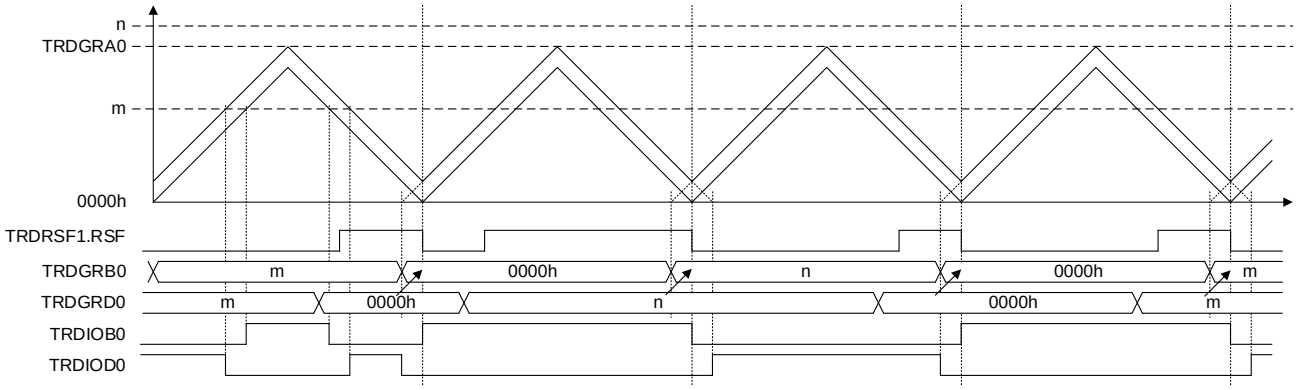


TRDFCR.CPSS = 1 (Asymmetry PWM output):



- Operation example with duty cycle 100%, duty cycle 0%
  In this mode, duty cycle 0% and 100% can be output continuously.

Figure 8-86. Operation Example (Duty Cycle 100%, Duty Cycle 0%)



The above diagram applies under the following condition:
TRDFCR.CPSS=0

- Set the buffer registers (TRDGRD0, TRDGRC1, TRDGRD1, TRDCMPD0, TRDCMPC1, TRDCMPD1) within the following range except when outputting with duty 100% and duty 0%.
  Setting range: > Value set in TRD0 register
  < Value set in TRDGRA0 register − value set in TRD0 register
- When extended complementary PWM mode is used for motor control, make sure OLS0 = OLS1.

- To stop and restart the counting operation, follow the procedure below.
  (1) Set the TSTART0 and TSTART1 bits of the TRDSTR register to “0” (count stop).
  (2) Set the CMD1 and CMD0 bits of the TRDFCR register to "00B" (timer mode or PWM3 mode).
  (3) Set bits CMD1, CMD0, and EPWM in the TRDFCR register to "10B" and "1" (extended complementary PWM mode).
  (4) Set the TRD0 register to the dead-time setting value, and set the TRD1 register to 0000H.
  (5) Set the registers TRDGRC0, TRDGRD0, TRDGRC1, and TRDGRD1. In asymmetric waveform model, also set the registers TRDGRm and TRDCMPm (m=B0, D0, A1, B1, C1, or D1). Also, set the paired registers for TRDGRB0 and TRDGRD0, TRDGRA1 and TRDGRC1, TRDGRB1 and TRDGRD1, TRDCMPB0 and TRDCMPD0, TRDCMPA1 and TRDCMPC1, and TRDCMPB1 and TRDCMPD1 to the same values.
  (6) To use A/D trigger function, set the TRDADTB0 and TRDADTB1 registers.
  (7) Reset the initial output level of the pin with the TRDOCR register.
  (8) Set the TSTART0 and TSTART1 bits to “1” (count start).In this mode, duty cycle 0% and 100% can be output continuously.

- Do not set the general registers TRDGRA0, TRDGRB0, TRDGRA1, TRDGRB1, TRDCMPB0, TRDCMPA1, and TRDCMPB1 during timer counting. To set these registers, set the buffer registers TRDGRD0, TRDGRC1, TRDGRD1, TRDCMPD0, TRDCMPC1, and TRDCMPD1, then set the RDT bit in the TRDRDT1 register to 1 (use the register reload function). The PWM period (TRDGRA0 register value) cannot be changed.

8.6 PWM Option Unit A (PWMOPA)

The PWM option unit is used to cutoff and release the output from timer RDe and ports with the comparator 0 output, external interrupt 0 (INTP0), and event link controller (ELC) as trigger signals. The PWM option unit is a different function from the pulse forced cutoff incorporated in timer RDe.

Table 8-29. Functional Differrence between Pulse Forced Cutoff and Output Forced Cutoff

	Pulse Forced Cutoff of Timer RDe	Output Forced Cutoff of PWM Option Unit
Mode supporting forced cutoff	• PWM function • Reset synchronous PWM mode • Complementary PWM mode • PWM3 mode • Ectended PWM mode • Extended complementary PWM mode	• Supports all output modes for timer RDe • Port output can also be cut off.
Cutoff source	• ELC input Note • Low-level input of INTP0	• ELC input Note • INTP0 • Comparator 0 output Note
Cutoff release	• Release via software (while stopping Timer RDe counting)	• Released via hardware • Released via software (stopping counting is unnecessary)
Pin that can be cut off	Selected from among the pins set for timer RDe output from P13(P15)/TRDIOA0, P125(P11)/TRDIOB0, P14/TRDIOC0, P120(P12)/TRDIOD0, P15/TRDIOA1, P17/TRDIOB1, P16/TRDIOC1, P30/TRDIOD1.	Selected from among P13(P15)/TRDIOA0, P125(P11)/TRDIOB0, P14/TRDIOC0, P120(P12)/TRDIOD0, P15/TRDIOA1, P17/TRDIOB1, P16/TRDIOC1, P30/TRDIOD1. Port output can also be cut off.
Port state selection at cutoff	• High-impedance output • Low-level output • High-level output	• High-impedance output • Low-level output • High-level output However, when port output is cut off, only high-impedance output can be selected.

Note The functions of ELC and comparator are only available in the RL78/F25.

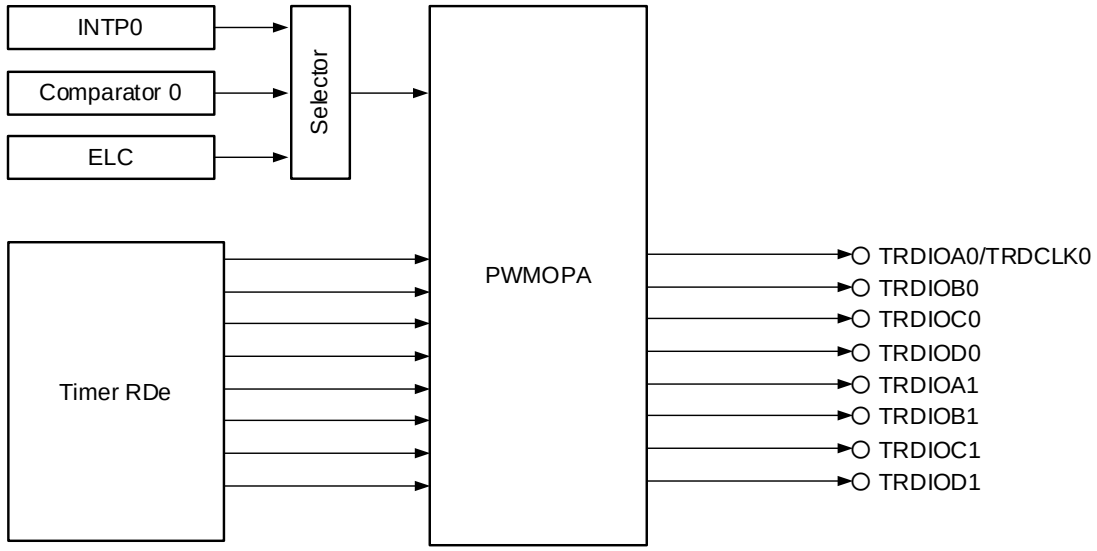
Caution When pulse forced cutoff and output forced cutoff are used simultaneously, the same cutoff source should not be selected.

8.6.1 Overview of PWM Option Unit A

The PWM option unit A has the following functions.

- Comparator 0, external interrupt 0, and the event link controller can be selected as an output cutoff source.
- When comparator 0 or external interrupt 0 is an output forced cutoff source, the edge to become a cutoff source can be selected.
- Software release and hardware release can be selected for output forced cutoff release.
- High-level, low-level, and high-impedance can be selected for the output level at cutoff.

Figure 8-87. PWMOPA Control Logic



8.6.2 Registers Controlling PWM Option Unit A

Table 8-30 lists the registers controlling the PWM option unit A.

Table 8-30. Registers Controlling PWMOPA

Address	Register Name	Symbol	After Reset	Access Size
F02C1H	Peripheral enable register 1	PER1	00H	1, 8
F0248H	PWMOPA control register 0	OPCTL0	00H Note	1, 8
F0249H	PWMOPA cutoff control register 0	OPDF0	00H Note	8
F024AH	PWMOPA cutoff control register 1	OPDF1	00H Note	8
F024BH	PWMOPA edge selection register	OPEDGE	00H Note	8
F024CH	PWMOPA status register	OPSR	00H Note	1, 8

Note If it is necessary to read the initial value, set the PWMOPEN bit in the PER1 register to 1 before reading.

8.6.2.1 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 PWM option unit A, be sure to set bit 2 (PWMOPEN) 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 8-88. 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	CTSUEN	CMPEN Note	TRD0EN	DTCEN	PWMOPEN	0	TRJ0EN

PWMOPEN	Control of PWMOPA input clock supply
0	Stops input clock supply. • SFR used by PWMOPA cannot be written. • PWMOPA is in the reset status.
1	Enables input clock supply. • SFR used by PWMOPA can be read and written.

Note Only for RL78/F25.

Cautions 1. Be sure to clear the following bits to 0.
RL78/F22: bits 1, 5, and 7
RL78/F25: bit 1

2. When setting PWMOPA, be sure to set the PWMOPEN bit to 1 first. If PWMOPEN = 0, writing to a control register of PWMOPA is ignored, and all read values are default values.

8.6.2.2 PWMOPA Control Register 0 (OPCTL0)

PWM option unit A is controlled by this register.

The OPCTL0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H

Figure 8-89. Format of PWMOPA Control Register 0 (OPCTL0)

Address: F0248H After Reset: 00H [Note 10]

Symbol	7	6	5	4	3	2	<1>	0
OPCTL0	0	HAZAD_SET	IN_EG	IN_SEL1	IN_SEL0	ACT	HZ_REL	HS_SEL

Bit 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

HAZAD_SET	Output cutoff hazard control selection [Note 1]	R/W
0	Hazard measure disabled	R/W
1	Hazard measure enabled	

IN_EG	Output forced cutoff source edge/output forced cutoff release edge selection [Notes 2,3]	R/W
0	Rising edge: Output forced cutoff Falling edge: Output forced cutoff release	R/W
1	Rising edge: Output forced cutoff release Falling edge: Output forced cutoff	

IN_SEL1	IN_SEL0	Cutoff source selection [Notes 2, 4, 5]	R/W
0	0	No output cutoff source selection	R/W
0	1	Comparator 0 output	
1	0	INTP0 pin input	
1	1	Event input from ELC	

ACT	When software release is selected: Software release timing selection	R/W
0	When HZ_REL is set to 1 via software, forced cutoff is released and pulse output is resumed.	R/W
1	When HZ_REL is set to 1, forced cutoff is released and pulse output is resumed at the following timing. • Timer RDe complementary PWM mode: Output forced cutoff is released at the TRDIOC0 edge timing selected in the OPEDGE register and pulse output is resumed. • Timer RDe reset synchronous PWM mode: Output forced cutoff is released when the TRD0 count becomes 0000H. • Other than the modes above: TRDIOj0 (j = A, B, C, D) forced cutoff is released when the TRD0 count becomes 0000H. TRDIOj1 forced cutoff is released when the TRD1 count becomes 0000H. [Note 6]	

HZ_REL	When software release is selected: Output cutoff release control	R/W
0	Output forced cutoff continues (if forced cutoff is released, this bit becomes 0). [Note 7]	R/W
1	Output forced cutoff is released and pulse output is resumed. [Note 8]	
• The value that can be read from or written to the HZ_REL bit differs depending on the state. • Normal state: 1 or 0 is written and only 0 can be read. • Output forced cutoff state: Only 1 can be written and only 1 can be read.		

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

HS_SEL	Output forced cutoff release mode selection	R/W
0	Released via hardware When releasing output forced cutoff via hardware, the cutoff release timing varies depending on the operating mode of timer RDe. • Timer RDe complementary PWM, extended complementary PWM modes: After a cutoff release source is detected, output forced cutoff is released at the TRDIOC0 edge timing selected in OPEDGE. • Timer RDe reset synchronous PWM mode: After a cutoff release source is detected, output forced cutoff is released when the TRD0 count becomes 0000H. Note 6 • Other than the timer RDe modes above: After a cutoff release source is detected, TRDIOj0 (j = A, B, C, D) output forced cutoff is released when the TRD0 count becomes 0000H. TRDIOj1 output forced cutoff is released when the TRD1 count becomes 0000H. Note 9	R/W
1	Released via software	

Notes 1. Do not change it while timer RDe is operating.

2. To set, set the IN_SEL1 and IN_SEL0 bits at least three clocks after the IN_EG bit has been set.

3. It is enabled when comparator 0 output or INTP0 pin input is selected as an output cutoff source.

4. To release output forced cutoff with an ELC source, make sure to select software release (set the HS_SEL bit to 1). There is no restriction on output cutoff release with external interrupt 0 (INTP0) and comparator 0.

5. Set the input enabled level period of the comparator 0 output and INTP0 to one clock or longer.

6. When all of bit 15 through bit 0 of the counter becomes 0 while timer RD0 and timer RD1 are operating, the timer RD0 and timer RD1 count values become 0000H.

7. If timer RDe operates in the output compare function, PWM function, or PWM3 mode, the operation at output cutoff release differs between cases where 2 channels are used and 1 channel is used.

- If timer RDe is used with 2 channels
  If the HZ_REL bit is set to 1 via software, all the cutoff state bits (HZOF0, HZOF1) become 0 (cutoff release) and the HZ_REL bit becomes 0.
- If timer RDe is used with 1 channel
  If the HZ_REL bit is set to 1 via software, the cutoff state bits (HZOF0, HZOF1) corresponding to the used timer RDe channel become 0 and the HZ_REL bit becomes 0.

8. It cannot be set to 1 if forced cutoff has not occurred.

9. When timer RDe operates in the output compare function, PWM function, or PWM3 mode, cutoff cannot be released for channels not operating when the forced cutoff state is released (the cutoff state bits (HZOF0, HZOF1) do not become 0).

10. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) and TRD0EN = 0, PWMOPEN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1, PWMOPEN = 1 before reading.

8.6.2.3 PWMOPA Cutoff Control Register 0 (OPDF0)

This register is the PWM output TRDIOj0 (j = A, B, C, D) pulse cutoff control register of PWMOPA.

The OPDF0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-90. Format of PWMOPA Cutoff Control Register 0 (OPDF0)

Address: F0249H After Reset: 00H [Note]

Symbol	7	6	5	4	3	2	1	0
OPDF0	DFD01	DFD00	DFC01	DFC00	DFB01	DFB00	DFA01	DFA00

DFD01	DFD00	TRDIOD0 pin output forced cutoff control	R/W
0	0	Forced cutoff prohibited	R/W
0	1	High-impedance output	
1	0	Low-level output	
1	1	High-level output	

DFC01	DFC00	TRDIOC0 pin output forced cutoff control	R/W
0	0	Forced cutoff prohibited	R/W
0	1	High-impedance output	
1	0	Low-level output	
1	1	High-level output	

DFB01	DFB00	TRDIOB0 pin output forced cutoff control	R/W
0	0	Forced cutoff prohibited	R/W
0	1	High-impedance output	
1	0	Low-level output	
1	1	High-level output	

DFA01	DFA00	TRDIOA0 pin output forced cutoff control	R/W
0	0	Forced cutoff prohibited	R/W
0	1	High-impedance output	
1	0	Low-level output	
1	1	High-level output	

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) and TRD0EN = 0, PWMOPEN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1, PWMOPEN = 1 before reading.

Cautions
1. **When the TRDIOj0 (j = A, B, C, D) pin is used for port output with forced cutoff enabled, select high-impedance output.**
2. **Do not change the register value in the forced cutoff state.**

8.6.2.4 PWMOPA Cutoff Control Register 1 (OPDF1)

This register is the PWM output TRDIOj1 (j = A, B, C, D) pulse cutoff control register of PWMOPA.

The OPDF1 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-91. Format of PWMOPA Cutoff Control Register 1 (OPDF1)

Address: F024AH After Reset: 00H [Note]

Symbol	7	6	5	4	3	2	1	0
OPDF1	DFD11	DFD10	DFC11	DFC10	DFB11	DFB10	DFA11	DFA10

DFD11	DFD10	TRDIOD1 pin output forced cutoff control	R/W
0	0	Forced cutoff prohibited	R/W
0	1	High-impedance output	
1	0	Low-level output	
1	1	High-level output	

DFC11	DFC10	TRDIOC1 pin output forced cutoff control	R/W
0	0	Forced cutoff prohibited	R/W
0	1	High-impedance output	
1	0	Low-level output	
1	1	High-level output	

DFB11	DFB10	TRDIOB1 pin output forced cutoff control	R/W
0	0	Forced cutoff prohibited	R/W
0	1	High-impedance output	
1	0	Low-level output	
1	1	High-level output	

DFA11	DFA10	TRDIOA1 pin output forced cutoff control	R/W
0	0	Forced cutoff prohibited	R/W
0	1	High-impedance output	
1	0	Low-level output	
1	1	High-level output	

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) and TRD0EN = 0, PWMOPEN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1, PWMOPEN = 1 before reading.

Cautions
1. **When the TRDIOj1 (j = A, B, C, D) pin is used for port output with forced cutoff enabled, select high-impedance output.**
2. **Do not change the register value in the forced cutoff state.**

8.6.2.5 PWMOPA Edge Selection Register (OPEDGE)

This register selects the timing for cutoff release when timer RDe is set to complementary PWM mode or extended complementary PWM mode and output forced cutoff is released by hardware.

The OPEDGE register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-92. Format of PWMOPA Edge Selection Register (OPEDGE)

Address: F024BH After Reset: 00H [Note]

Symbol	7	6	5	4	3	2	1	0
OPEDGE	0	0	0	0	0	0	EG1	EG0

Bits 7 to 2	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

EG1	EG0	Output forced cutoff release edge selection	R/W
0	0	Cutoff released at the rising edge of TRDIOC0	R/W
0	1	Cutoff released at the falling edge of TRDIOC0	
1	0	Cutoff released at both edges of TRDIOC0	
1	1	Input edge of TRDIOC0 disabled, cutoff retained	

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) and TRD0EN = 0, PWMOPEN = 0 in the PER1 register. If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1, PWMOPEN = 1 before reading.

8.6.2.6 PWMOPA Status Register (OPSR)

This register displays statuses of output forced cutoff and cutoff sources.

The OPSR register can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H

Figure 8-93. Format of PWMOPA Status Register (OPSR)

Address: F024CH After Reset: 00H [Note 3]

Symbol	7	6	5	4	3	2	1	0
OPSR	0	0	0	0	0	HZOF1	HZOF0	HZIF0

Bits 7 to 3	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

HZOF1	Cutoff State [Note 1]	R/W
0	Normal timer output (TRDIOA1, TRDIOB1, TRDIOC1, TRDIOD1)	R
1	Cutoff state (TRDIOA1, TRDIOB1, TRDIOC1, TRDIOD1)	

HZOF0	Cutoff State [Note 1]	R/W
0	Normal timer output (TRDIOA0, TRDIOB0, TRDIOC0, TRDIOD0)	R
1	Cutoff state (TRDIOA0, TRDIOB0, TRDIOC0, TRDIOD0)	

HZIF0	Output cutoff source state [Notes 1, 2]	R/W
0	State in which output cutoff source has not exceeded threshold	R
1	State in which output cutoff source has exceeded threshold	

Notes

1. If the output cutoff source has exceeded the threshold before selecting the INTP0 and comparator 0 cutoff sources with the IN_SEL1 and IN_SEL0 bits in the OPCTL0 register, after the IN_SEL1 and IN_SEL0 bits are set, the HZIF0 bit is set to 1, but the HZOF0 and HZOF1 bits are not set.

2. Effective when INTP0 and comparator 0 cutoff sources are selected.

3. The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) and TRD0EN = 0, PWMOPEN = 0 in the PER1 register.
If it is necessary to read the initial value, set f_{CLK} to f_{IH} and TRD0EN = 1, PWMOPEN = 1 before reading.

8.6.3 Operation

Output forced cutoff and return of the timer RDe output pin TRDIOji (j = A, B, C, D; i = 0, 1) can be controlled by using the INTP0 input, event input from ELC, or comparator 0 output as a trigger. When INTP0 input or comparator 0 output is used as a cutoff source, the edge to trigger output forced cutoff or output forced cutoff release can be selected.

8.6.3.1 Forced Cutoff

Output Pulse output from the timer RDe output pin TRDIOji (j = A, B, C, D; i = 0, 1) can be cut off by using the INTP0 input, event input from ELC, or comparator 0 output as a trigger.

When an output forced cutoff source is detected, the output of timer RDe is forcibly cut off, and the output value specified in the OPDF0/OPDF1 register is output. For detailed operation, see **Figure 8-95**.

Forced cutoff release can be selected from hardware and software by using the setting value of the HS_SEL bit of the OPCTL0 register.

8.6.3.2 Hardware Release (HS_SEL = 0)

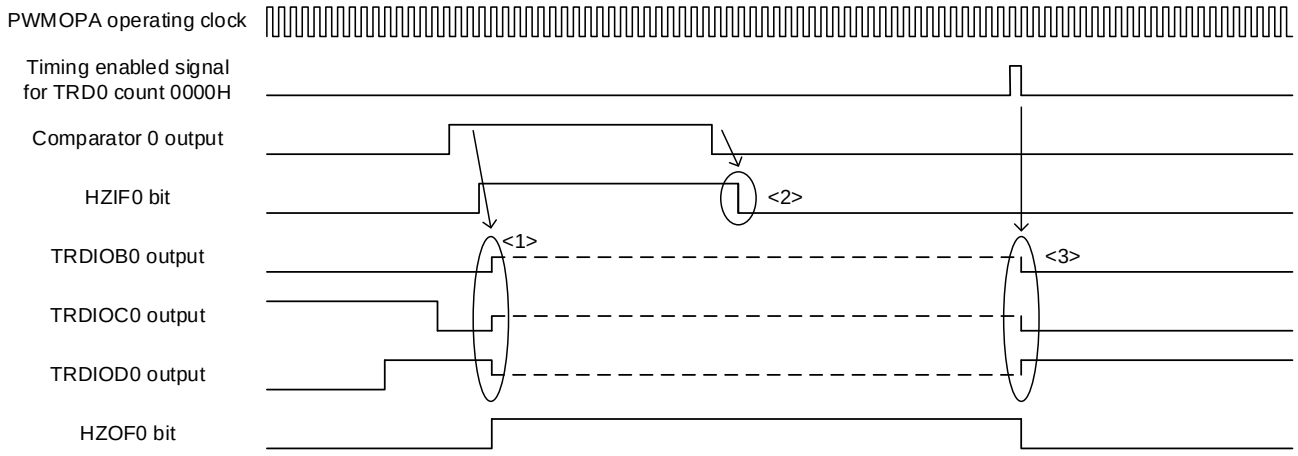
The timing of output forced cutoff release varies depending on the function of timer RDe.

(1) Output other than the timer RDe complementary PWM mode or extended complementary PWM mode

- Timer RDe is in the output compare function, PWM function, PWM3 mode, or extended PWM mode:
  Output forced cutoff of TRDIOA0, TRDIOB0, TRDIOC0, and TRDIOD0 is released when the TRD0 count value becomes 0000H after an output forced cutoff release source is detected. Output forced cutoff of TRDIOA1, TRDIOB1, TRDIOC1, and TRDIOD1 is released when the TRD1 count value becomes 0000H.
- Timer RDe is in the reset synchronous PWM mode:
  Forced cutoff of all TRDIOji (j = A, B, C, D; i = 0, 1) pins is released when the TRD0 count value becomes 0000H after an output forced cutoff release source is detected.

Figure 8-94. Operation Example of Output Forced Cutoff or Output Forced Cutoff Release by Hardware

(an example of cutoff of TRDIOB0, TRDIOC0, and TRDIOD0 pins)



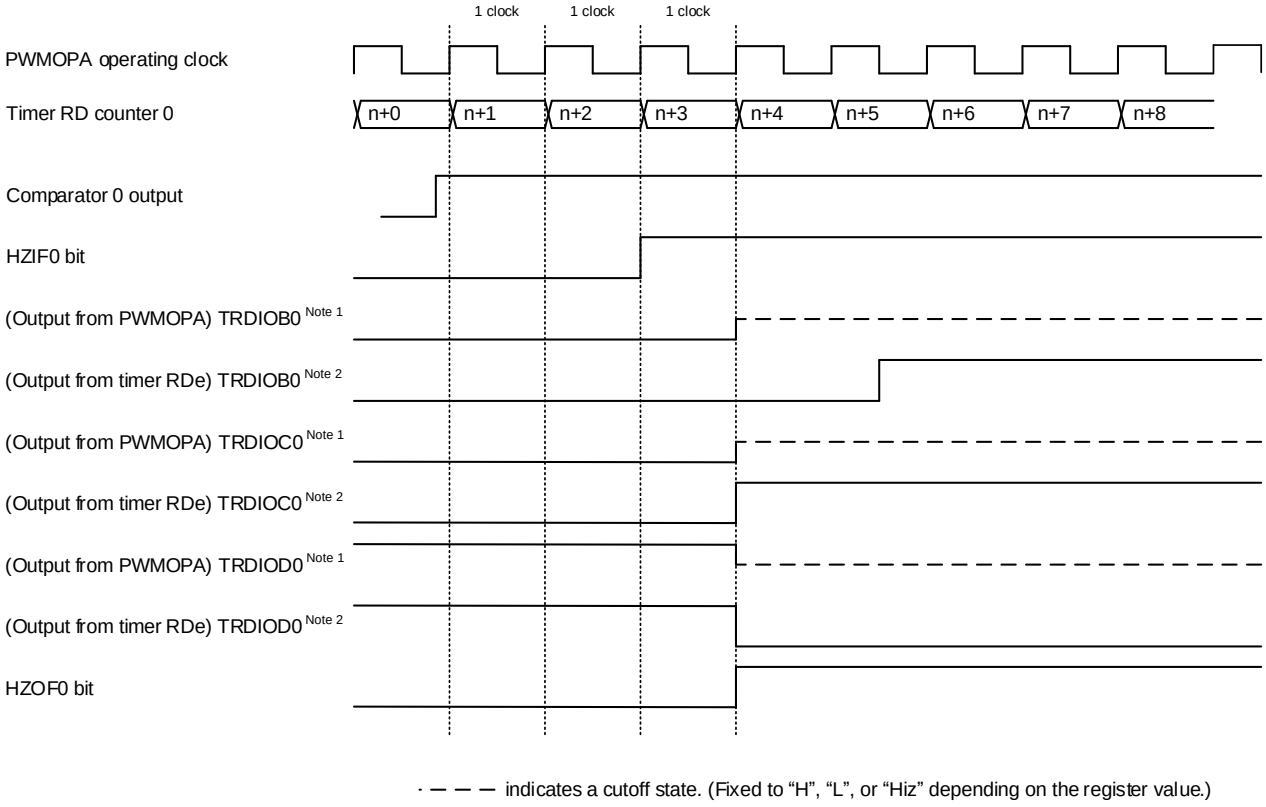
<1> The TRDIOB0, TRDIOC0, and TRDIOD0 pin outputs enter the output forced cutoff state when the rising edge of the comparator 0 output signal is detected.

<2> The HZIF0 bit is cleared after the falling edge of the comparator 0 output signal is detected.

<3> The forced cutoff state is released when the TRDi count value becomes 0000H.

Conditions: HS_SEL = 0 (Forced cutoff release mode: Released via hardware)
INSEL[1:0] = 01B (Cutoff source: Comparator 0 output)
IN_EG = 0 (Output forced cutoff at rising edge, cutoff release at falling edge)

Figure 8-95. Cutoff Detailed Timing Diagram

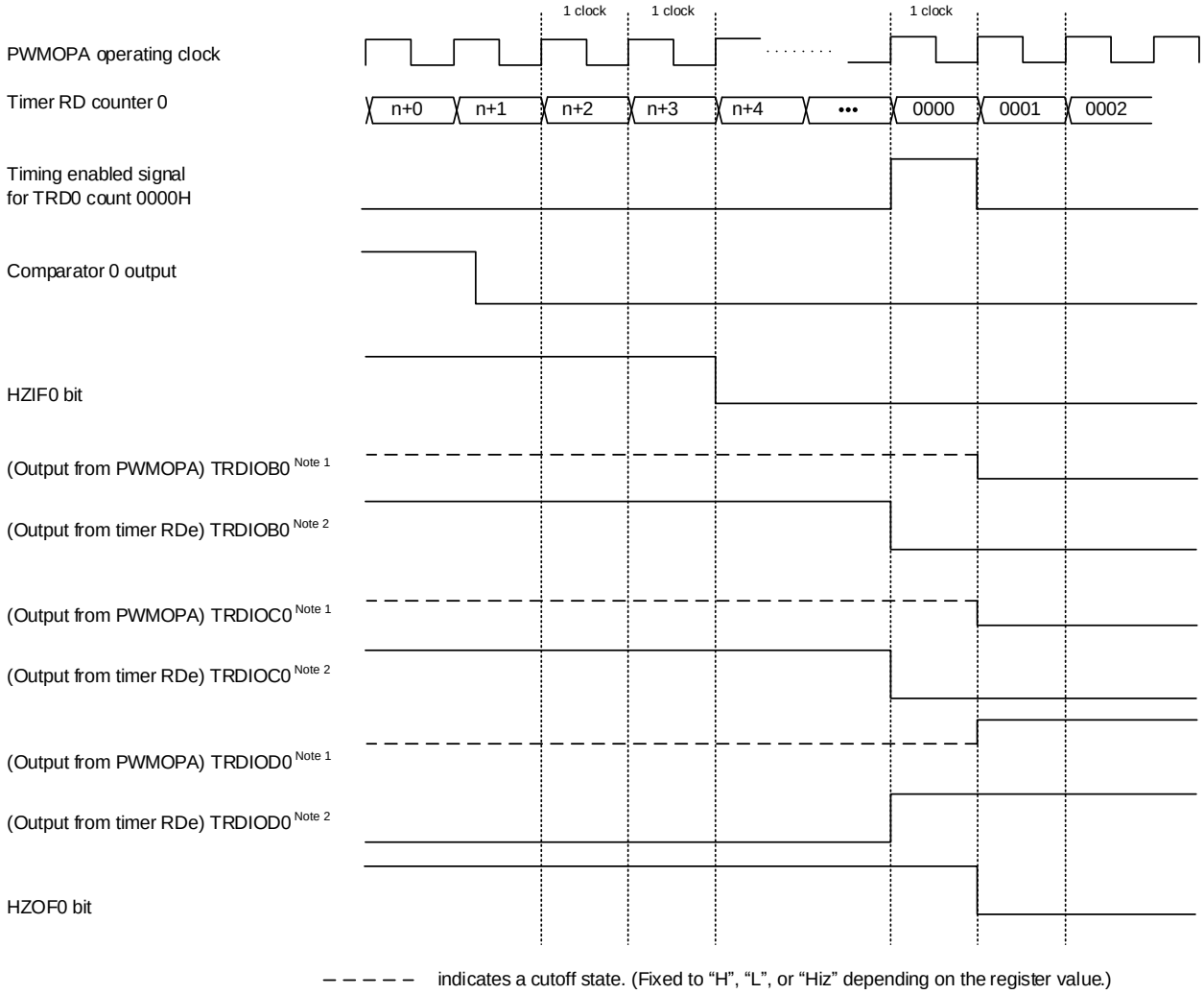


· — — — indicates a cutoff state. (Fixed to “H”, “L”, or “Hiz” depending on the register value.)

Notes 1. (Output from PWMOPA) TRDIOj0 (j = B, C, or D) indicates the state of the multiplexed timer RDe function pin.

2. (Output from Timer RDe) TRDIOj0 indicates input to PWMOPA from the timer RDe.

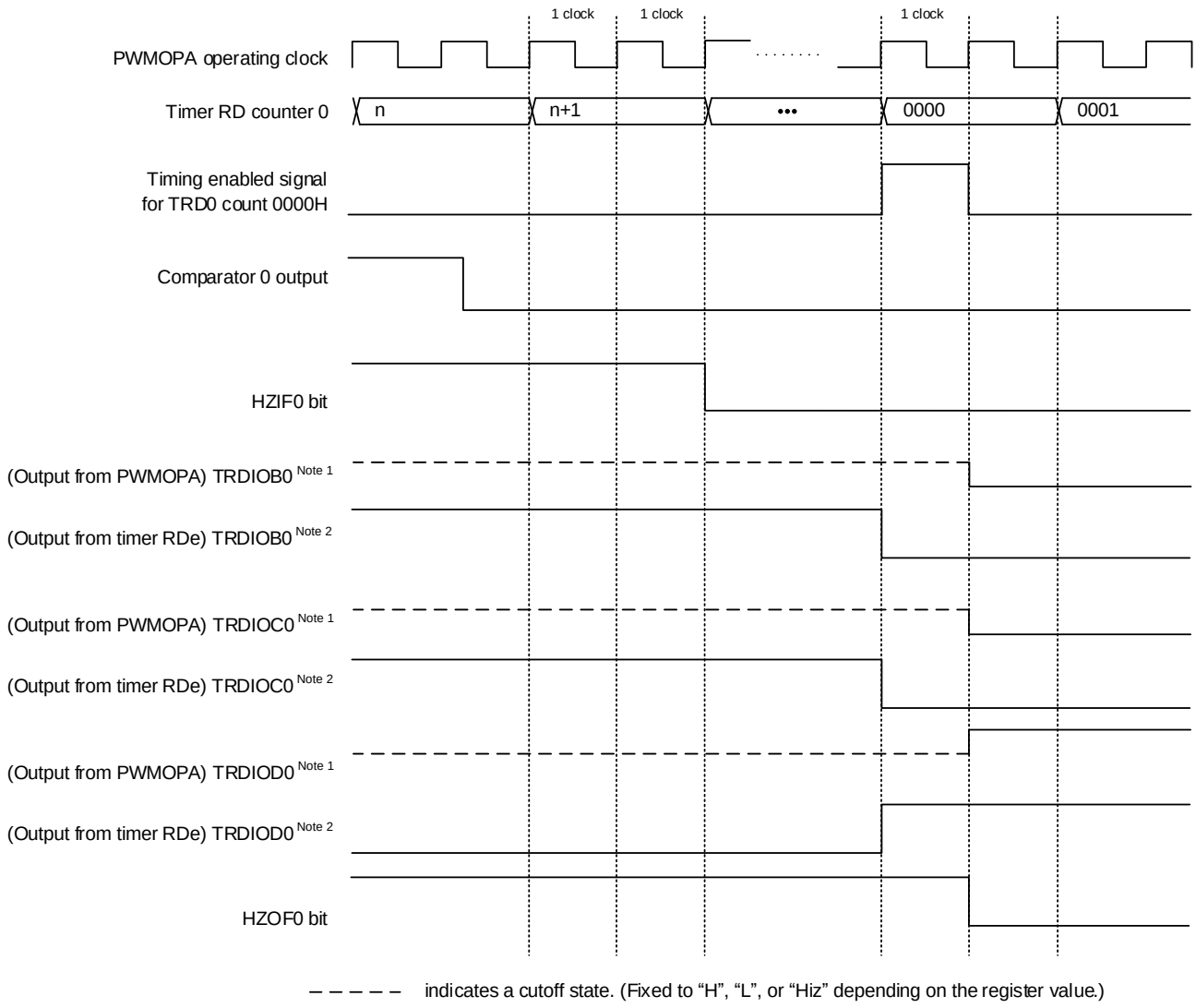
Figure 8-96. Cutoff Release Detailed Timing Diagram (Timer RDe Count Source = f_{CLK})



Notes 1. (Output from PWMOPA) TRDIOj0 (j = B, C, or D) indicates the state of the multiplexed timer RDe function pin.

2. (Output from Timer RDe) TRDIOj0 indicates input to PWMOPA from the timer RDe.

Figure 8-97. Cutoff Release Detailed Timing Diagram (Timer RDe Count Source = f_{CLK}/2)

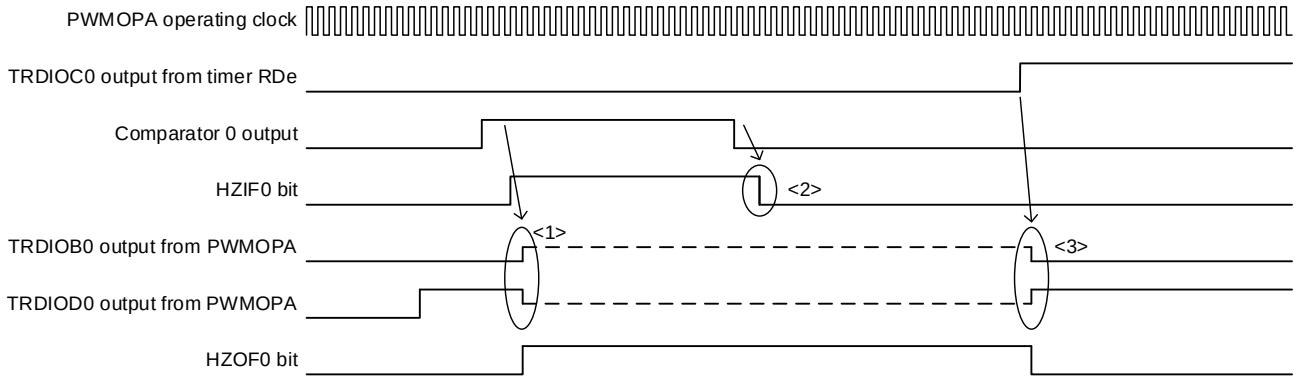


Notes 1. (Output from PWMOPA) TRDIOj0 (j = B, C, or D) indicates the state of the multiplexed timer RDe function pin.

2. (Output from Timer RDe) TRDIOj0 indicates input to PWMOPA from the timer RDe.

(2) Output of the timer RDe complementary PWM mode or extended complementary PWM mode:
When an output forced cutoff source is detected, the output of timer RDe is forcibly cut off, and the output value specified in the OPDF0/OPDF1 register is output.

Figure 8-98. Operation Example of Hardware Cutoff Release Function (an example of TRDIOB0 and TRDIOD0)

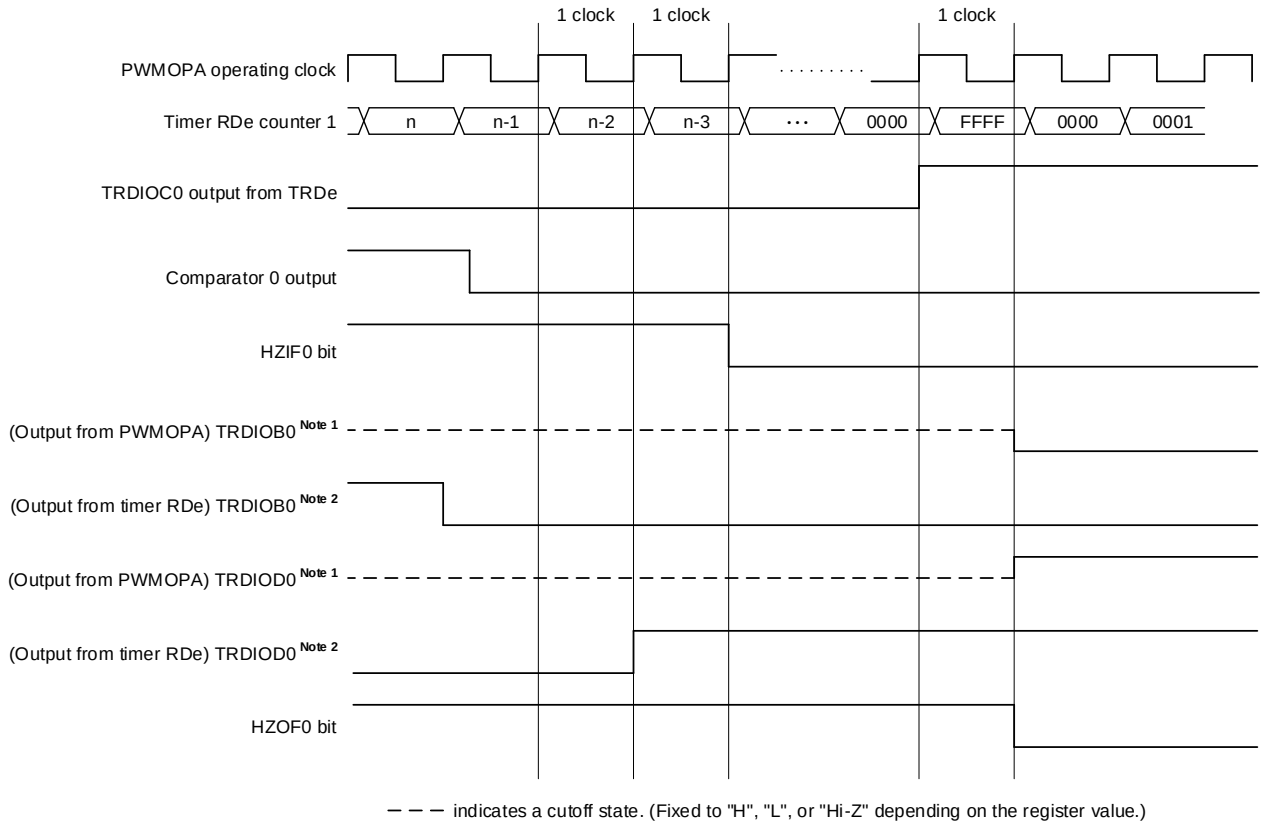


<1> The TRDIOB0 and TRDIOD0 pin outputs enter the output forced cutoff state when the rising edge of the comparator 0 output signal is detected.
<2> The HZIF0 bit is cleared after the falling edge of the comparator 0 output signal is detected.
<3> The forced cutoff state is released at the rising edge of TRDIOC0.

Conditions: HS_SEL = 0 (Forced cutoff release mode: Released via hardware)
INSEL[1:0] = 01B (Cutoff source: Comparator 0 output)
IN_EG = 0 (Output forced cutoff at rising edge, cutoff release at falling edge)

For the cutoff detailed timing diagram, see **Figure 8-95**.

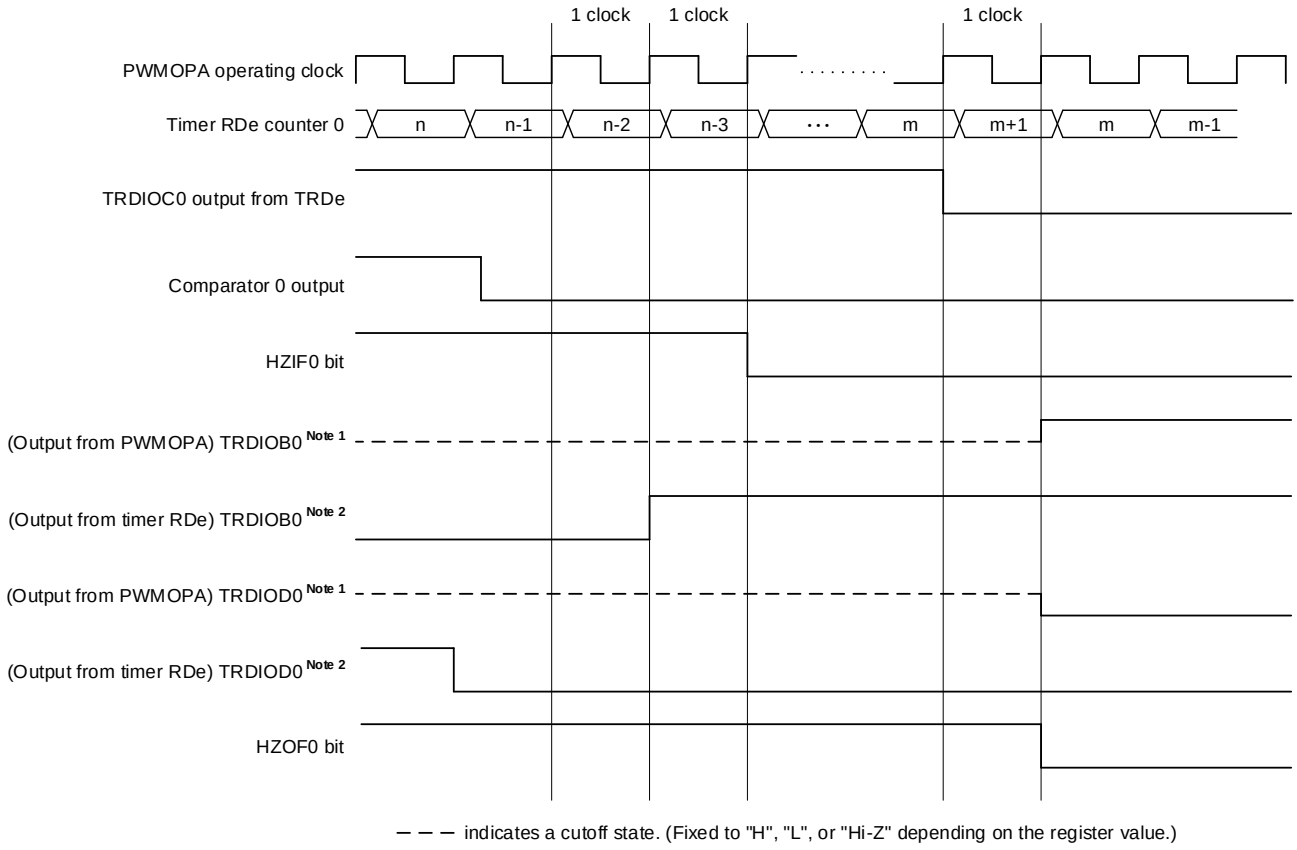
Figure 8-99. Cutoff Release Detailed Timing Diagram (Timer RDe Count Source = f_{CLK}, TRD1 decremented) (an example of TRDIOB0 and TRDIOD0)



Notes 1. (Output from PWMOPA) TRDIOj0 (j = B or D) indicates the state of the multiplexed timer RDe function pin.

2. (Output from Timer RDe) TRDIOj0 indicates input to PWMOPA from the timer RDe.

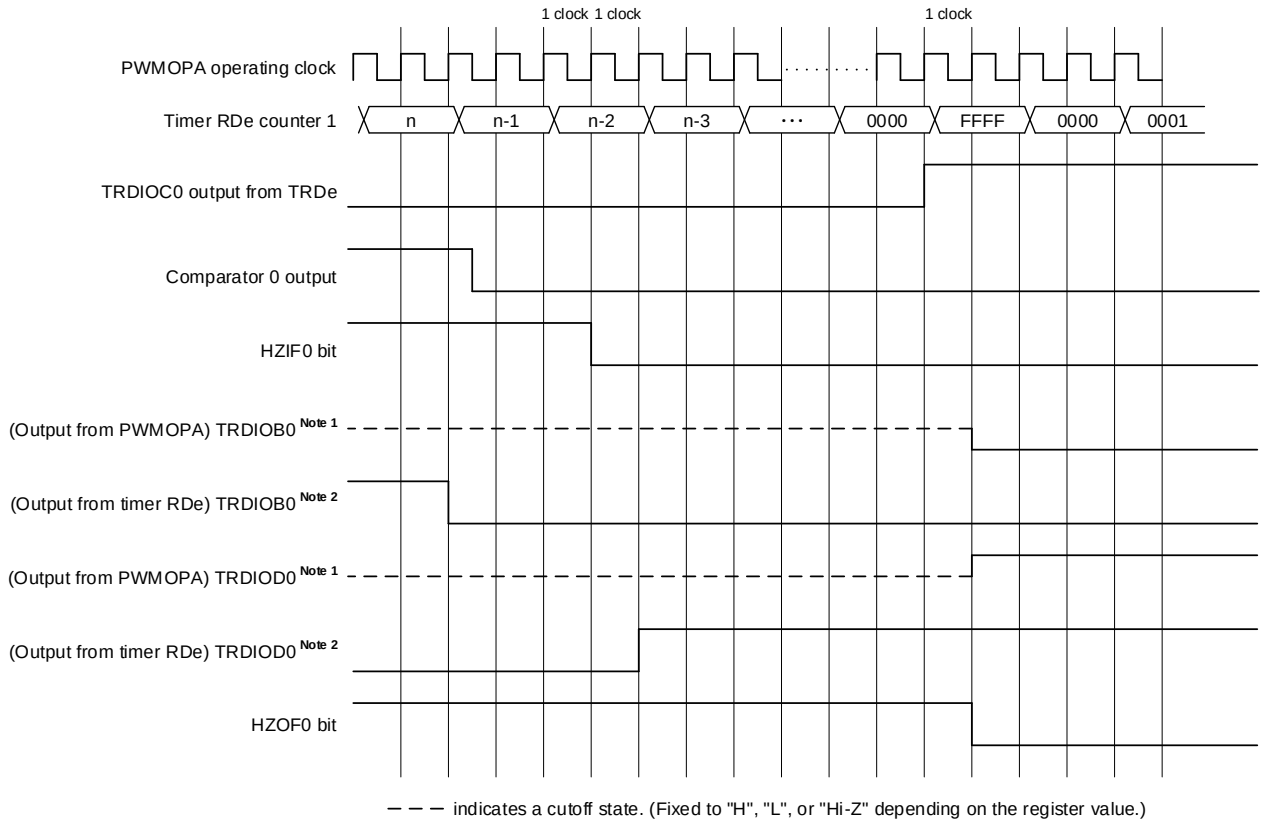
Figure 8-100. Cutoff Release Detailed Timing Diagram
(Timer RDe Count Source = f_{CLK}, Timer RD0 Count = TRDGRA0) (an example of TRDIOB0 and TRDIOD0)



Remark m: TRDGRA0 value

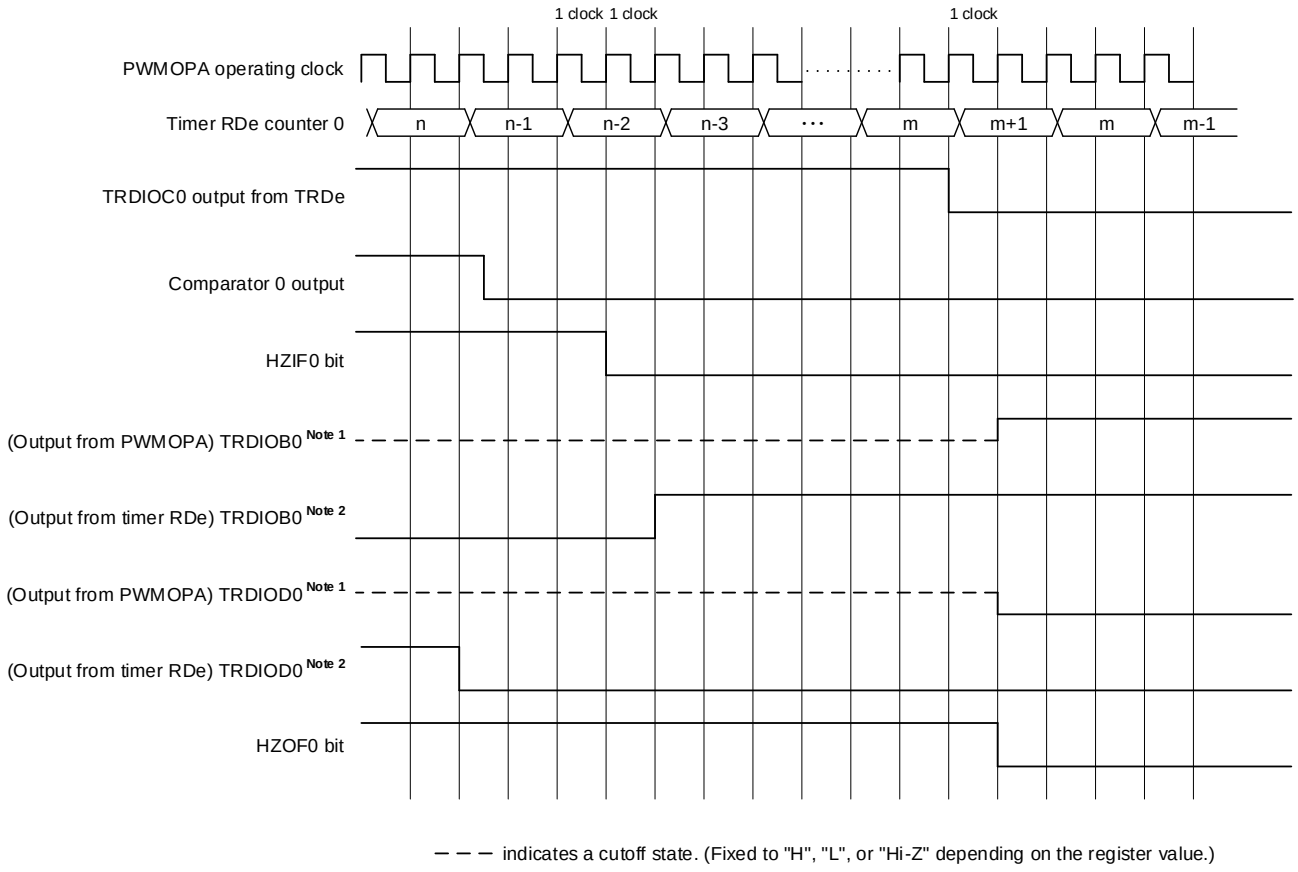
Notes 1. (Output from PWMOPA) TRDIOj0 (j = B or D) indicates the state of the multiplexed timer RDe function pin.
2. (Output from Timer RDe) TRDIOj0 indicates input to PWMOPA from the timer RDe.

Figure 8-101. Cutoff Release Detailed Timing Diagram
(Timer RDe Count Source = f_{CLK}/2, TRD1 Decremented) (an example of TRDIOB0 and TRDIOD0)



Notes 1. (Output from PWMOPA) TRDIOj0 (j = B or D) indicates the state of the multiplexed timer RDe function pin.
2. (Output from Timer RDe) TRDIOj0 indicates input to PWMOPA from the timer RDe.

Figure 8-102. Cutoff Release Detailed Timing Diagram
(Timer RDe Count Source = f_{CLK}/2, Timer RD0 Count = TRDGRA0) (an example of TRDIOB0 and TRDIOD0)



Remark m: TRDGRA0 value

Notes 1. (Output from PWMOPA) TRDIOj0 (j = B or D) indicates the state of the multiplexed timer RDe function pin.
2. (Output from Timer RDe) TRDIOj0 indicates input to PWMOPA from the timer RDe.

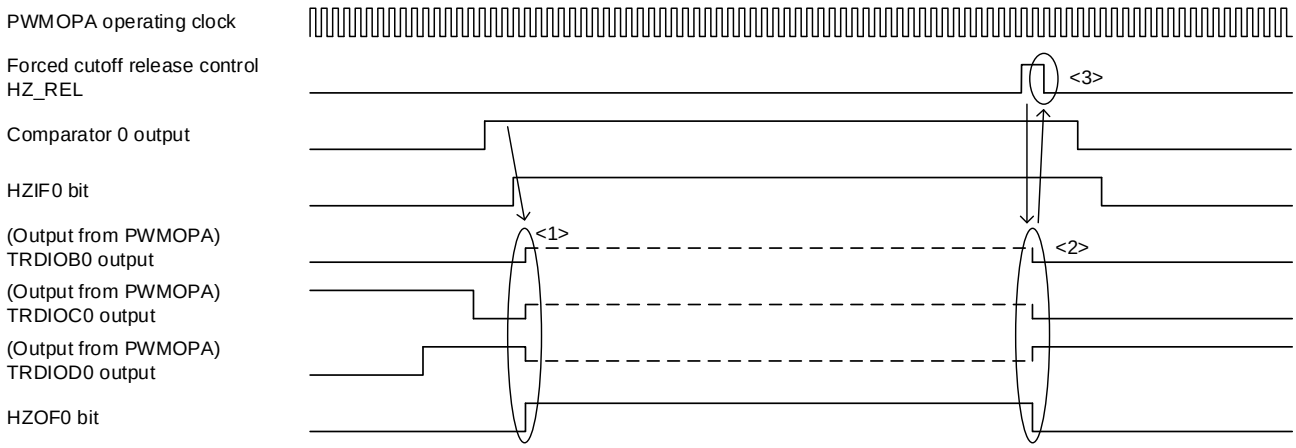
8.6.3.3 Software Cutoff Release (HS_SEL = 1)

The timing of output forced cutoff release varies depending on the setting of the ACT bit of the OPCTL0 register.

(1) Immediate release via software (ACT = 0)

If ACT is set to 0, forced cutoff is released immediately when the HZ_REL bit of the OPCTL0 register is set to 1.
After forced cutoff, the HZ_REL bit automatically becomes 0.

Figure 8-103. Operation Example of Cutoff Release by Software (an example of TRDIOB0, TRDIOC0, and TRDIOD0)

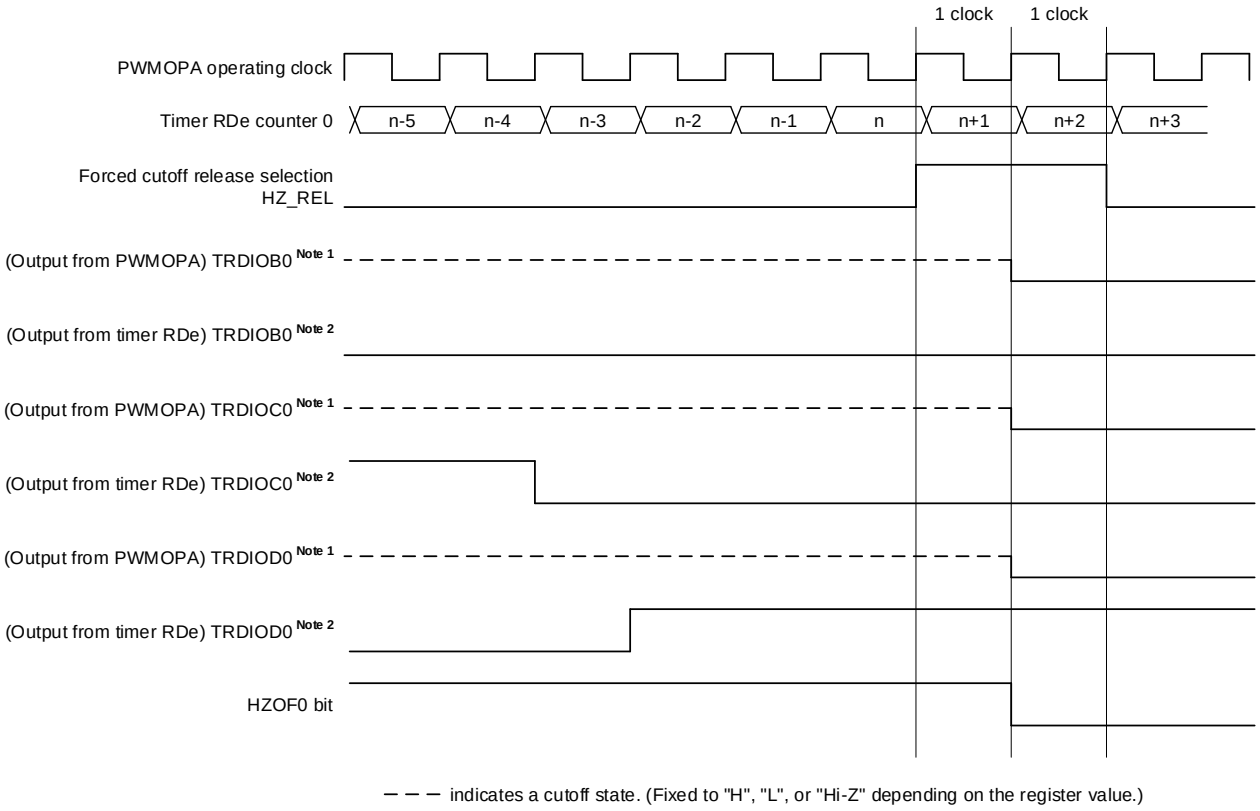


<1> The TRDIOB0, TRDIOC0, and TRDIOD0 pin outputs enter the output forced cutoff state when the rising edge of the comparator 0 output signal is detected.
<2> The HZ_REL bit is set to 1 to release forced cutoff immediately.
<3> After forced cutoff is released, the HZ_REL bit becomes 0.

Conditions: HS_SEL = 1 (Forced cutoff release mode: Released via software)
INSEL[1:0] = 01B (Cutoff source: Comparator 0 output)
IN_EG = 0 (Output forced cutoff at rising edge, cutoff release at falling edge)

For the cutoff detailed timing diagram, see **Figure 8-95**.

Figure 8-104. Cutoff Release Detailed Timing Diagram (an example of TRDIOB0, TRDIOC0, and TRDIOD0)



Notes 1. (Output from PWMOPA) TRDIOj0 (j = B, C, or D) indicates the state of the multiplexed timer RDe function pin.

2. (Output from Timer RDe) TRDIOj0 indicates input to PWMOPA from the timer RDe.

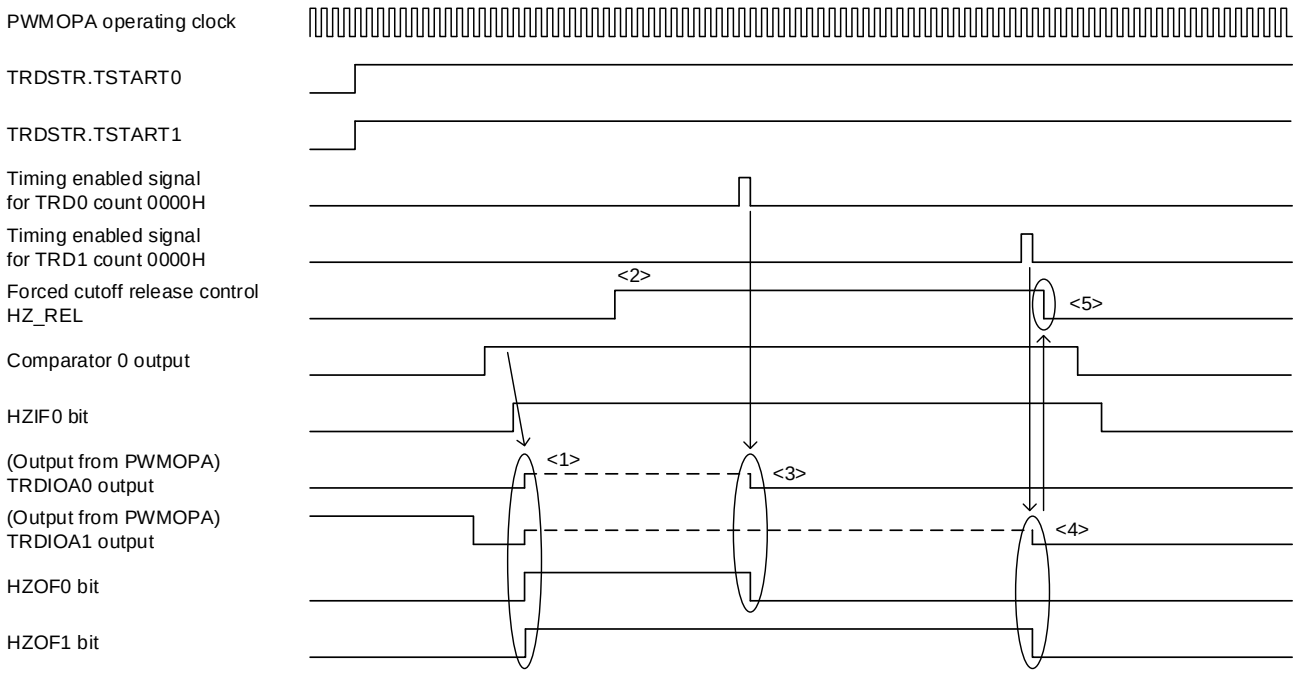
(2) Immediate release via software (ACT = 1)

If ACT is set to 1, forced cutoff can be released via the signal from timer RDe after the HZ_REL bit of the OPCTL0 register is set to 1. After forced cutoff is released, the HZ_REL bit automatically becomes 0.

Hardware release resumes output using the release signal from timer RDe as a trigger after an output forced cutoff release source is detected. Software release resumes output using the release signal from timer RDe as a trigger after the HZ_REL bit is set to 1. The release timing is the same.

- Timer RDe is in the output compare function, PWM function, PWM3 mode, or extended PWM mode:
  After the HZ_REL bit is set to 1, output forced cutoff of TRDIOA0, TRDIOB0, TRDIOC0, and TRDIOD0 is released when the TRD0 count value becomes 0000H. Output forced cutoff of TRDIOA1, TRDIOB1, TRDIOC1, and TRDIOD1 is released when the TRD1 count value becomes 0000H.

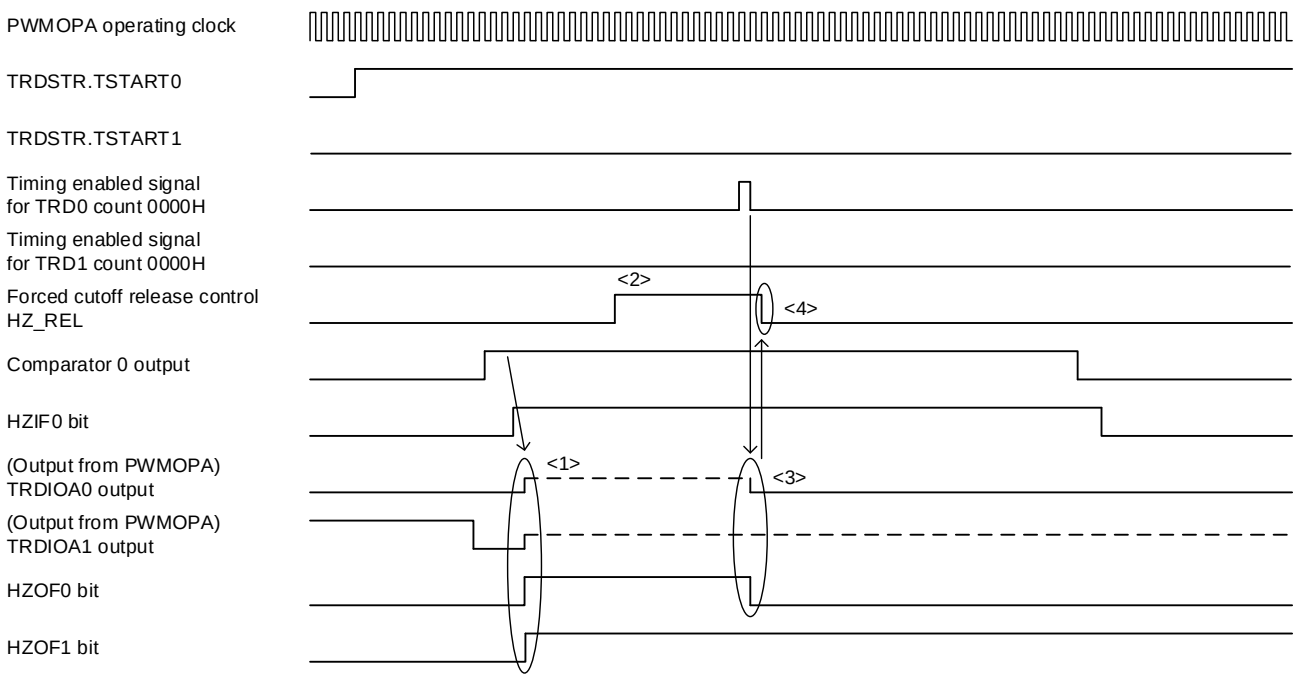
Figure 8-105. Operation Example of Cutoff Release by Software (Timer RDe, 2-channel count)



<1> The TRDIOA0 and TRDIOA1 pin outputs enter the output forced cutoff state when the rising edge of the comparator 0 output signal is detected.
<2> After the HZ_REL bit is set to 1, it waits until each of the counter values becomes 0000H.
<3> The TRDIOA0 forced cutoff state is released when the TRD0 count value becomes 0000H.
<4> The TRDIOA1 forced cutoff state is released when the TRD1 count value becomes 0000H.
<5> After forced cutoff of each channel is released, the HZ_REL bit becomes 0 automatically.

Conditions: HS_SEL = 1 (Forced cutoff release mode: Released via software)
INSEL[1:0] = 01B (Cutoff source: Comparator 0 output)
IN_EG = 0 (Output forced cutoff at rising edge, cutoff release at falling edge)

Figure 8-106. Operation Example of Cutoff Release by Software (Timer RDe, 1-channel count)



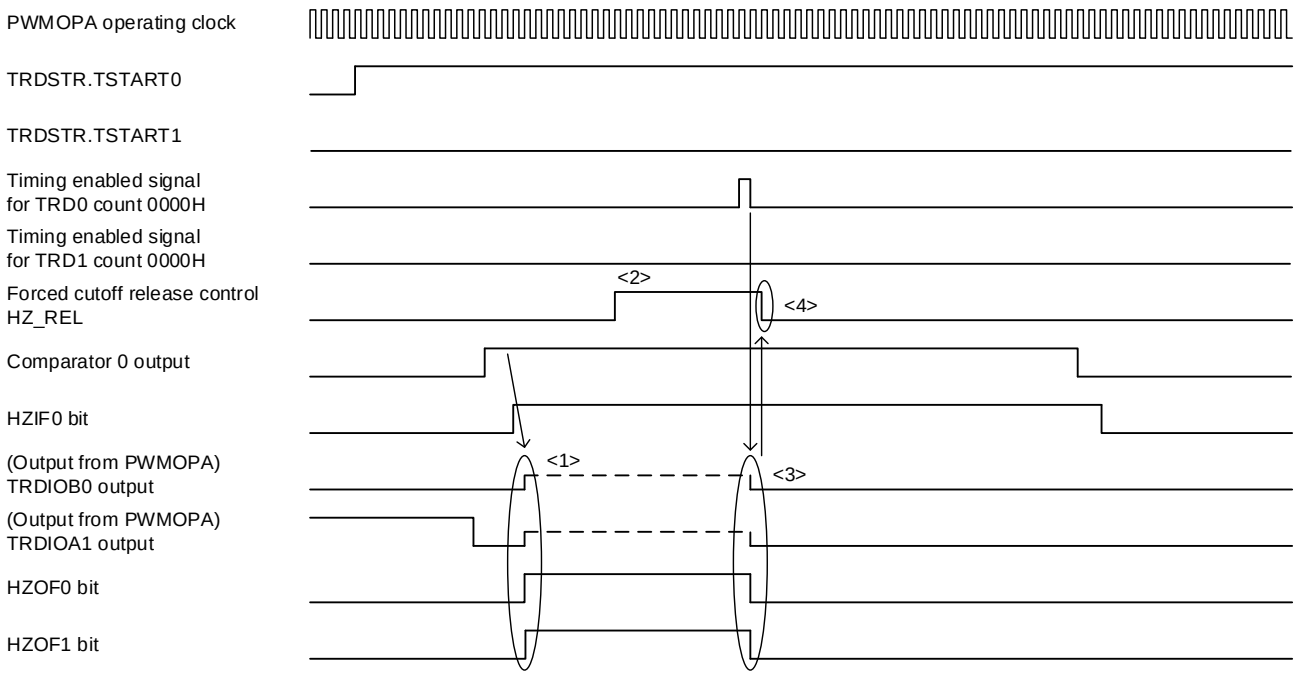
<1> The TRDIOA0 and TRDIOA1 pin outputs enter the output forced cutoff state when the rising edge of the comparator 0 output signal is detected.
<2> After the HZ_REL bit is set to 1, it waits until the counter value becomes 0000H.
<3> The TRDIOA0 forced cutoff state is released when the TRD0 count value becomes 0000H.
<4> After forced cutoff is released, the HZ_REL bit becomes 0 automatically.

Conditions: HS_SEL = 1 (Forced cutoff release mode: Released via software)
INSEL[1:0] = 01B (Cutoff source: Comparator 0 output)
IN_EG = 0 (Output forced cutoff at rising edge, cutoff release at falling edge)

For the cutoff detailed timing, see **Figure 8-95**.
For the cutoff release detailed timing, see **Figure 8-96** and **Figure 8-97**.
For the timing when HZ_REL bit becomes 0 automatically, see **Figure 8-104**.

- Timer RDe is in the reset synchronous PWM mode:
  Output forced cutoff of all TRDIOji (i = 0 or 1; j = A, B, C, or D) pins is released when the TRD0 count value becomes 0000H after the HZ_REL bit is set to 1.

Figure 8-107. Operation Example of Output Cutoff Release by Software



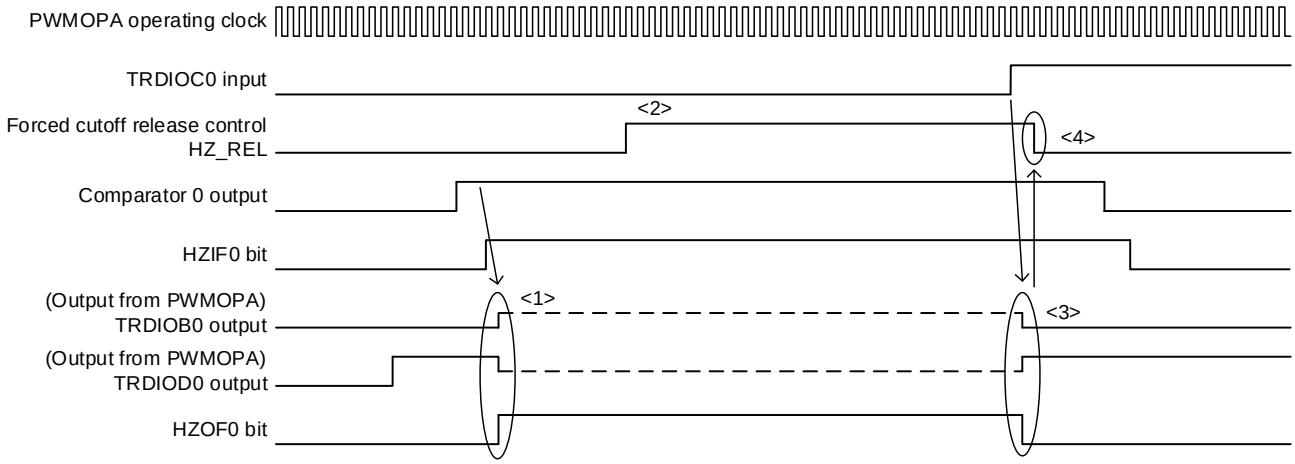
<1> The TRDIOB0 and TRDIOA1 pin outputs enter the output forced cutoff state when the rising edge of the comparator 0 output signal is detected.
<2> After the HZ_REL bit is set to 1, it waits until the counter value of timer RD becomes 0000H.
<3> The TRDIOB0 and TRDIOA1 forced cutoff states are released when the TRD0 count value becomes 0000H.
<4> After forced cutoff is released, the HZ_REL bit becomes 0 automatically.

Conditions: HS_SEL = 1 (Forced cutoff release mode: Released via software)
INSEL[1:0] = 01B (Cutoff source: Comparator 0 output)
IN_EG = 0 (Output forced cutoff at rising edge, cutoff release at falling edge)

For the cutoff detailed timing, see **Figure 8-95**.
For the cutoff release detailed timing, see **Figure 8-96** and **Figure 8-97**.
For the timing when HZ_REL bit becomes 0 automatically, see **Figure 8-104**.

- Timer RDe is in the complementary PWM mode or extended complementary PWM mode:
  If the OPEDGE register is set after the HZ_REL bit is set to 1, the output forced cutoff state of timer RDe is released at both edges, rising edge, or falling edge of the TRDIOC0 selected.

Figure 8-108. Operation Example of Cutoff Release by Software (an example of TRDIOB0 and TRDIOD0)



<1> The TRDIOB0 and TRDIOD0 pin outputs enter the output forced cutoff state when the rising edge of the comparator 0 output signal is detected.
<2> After the HZ_REL bit is set to 1, it waits for the TRDIOC0 rising signal.
<3> When the rising edge of TRDIOC0 is detected, The forced cutoff state is released.
<4> When forced cutoff is released, the HZ_REL bit becomes 0 automatically.

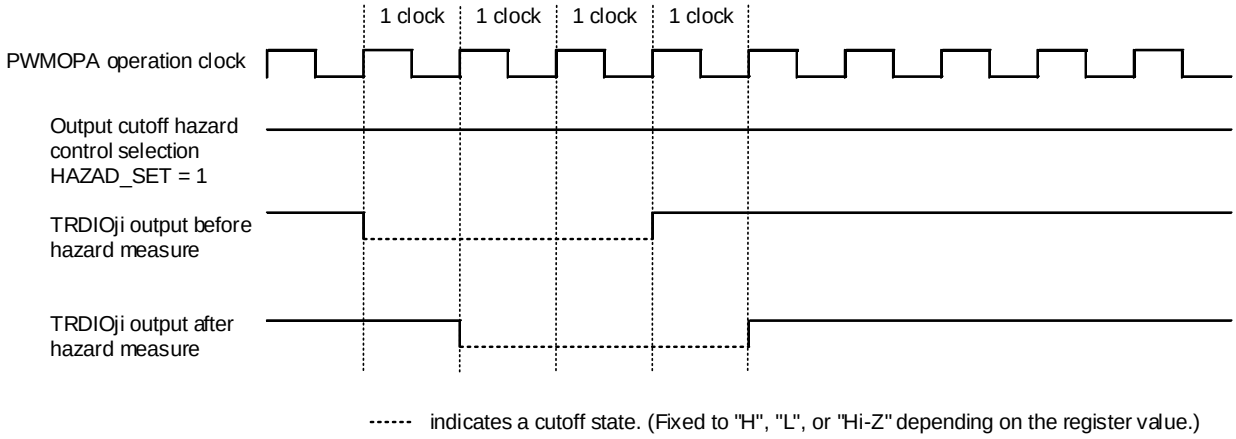
Conditions: HS_SEL = 0 (Forced cutoff release mode: Released via hardware)
INSEL[1:0] = 01B (Cutoff source: Comparator 0 output)
IN_EG = 0 (Output forced cutoff at rising edge, cutoff release at falling edge)

For the cutoff detailed timing, see **Figure 8-95**.
For the cutoff release detailed timing, see **Figure 8-99**, **Figure 8-100**, **Figure 8-101**, and **Figure 8-102**.
For the timing when HZ_REL bit becomes 0 automatically, see **Figure 8-104**.

8.6.3.4 Hazard Measures

A hazard may occur if switching between the TRDIOji pin and port is performed when a cutoff state occurs, cutoff release occurs, or while timer RDe is operating. Hazard risks can be handled by setting the hazard control selection bit (HAZAD_SET) to 1. However, for the timer RDe output when hazard control is enabled, a delay of one clock occurs from the timer RDe output when it is disabled.

Figure 8-109. Hazard Control Timing Diagram



Remark j = A, B, C, or D; i = 0 or 1

8.6.3.5 Output Cutoff Source Detected State and Output Cutoff Source Undetected State

Whether it is the output cutoff source detected state (HZIF0 = 1) or output cutoff source undetected state (HZIF0 = 0) is determined based on the level of the signals (INTP0, comparator 0 output) selected with the cutoff source selection bits (OPCTL0.IN_SEL1, OPCTL0.IN_SEL0).

If the output cutoff/output cutoff release edge bit (OPCTL0.IN_EG) is set to 0, the high-level becomes the output cutoff source detected state, and the low-level becomes the output cutoff source undetected state.

If the output cutoff/output cutoff release edge bit (OPCTL0.IN_EG) is set to 1, the low-level becomes the output cutoff source detected state, and the high-level becomes the output cutoff source undetected state.

Remark If the output cutoff source has exceeded the threshold before selecting the INTP0 and comparator 0 cutoff sources with the IN_SEL1 and IN_SEL0 bits in the OPCTL0 register, after the IN_SEL1 and IN_SEL0 bits are set, the HZIF0 bit is set to 1, but the HZOF0 and HZOF1 bits are not set.

8.6.3.6 Timing When Timer RDe Counter Value becomes 0000H

When releasing output forced cutoff via hardware, the output cutoff release conditions vary depending on the operating mode of timer RDe.

(1) Timing when the count value becomes 0000H if timer RDe is in the output compare function

- Count value = 0000H and timer RDe starts counting.
  Output cutoff is not released.
- Timer RDe is counting and 0000H is written to the counter with software.
  Output cutoff is released.
- The counter overflows and becomes 0000H.
  Output cutoff is released.
- The counter becomes 0000H at compare match with TRDGRAi register
  Output cutoff is released.

(2) Timing when the count value becomes 0000H if timer RDe is in the PWM function and extended PWM mode

- Count value = 0000H and timer RDe starts counting.
  Output cutoff is not released.
- Timer RDe is counting and 0000H is written to the counter with software.
  Output cutoff is released.
- The counter becomes 0000H at compare match with TRDGRAi register
  Output cutoff is released.

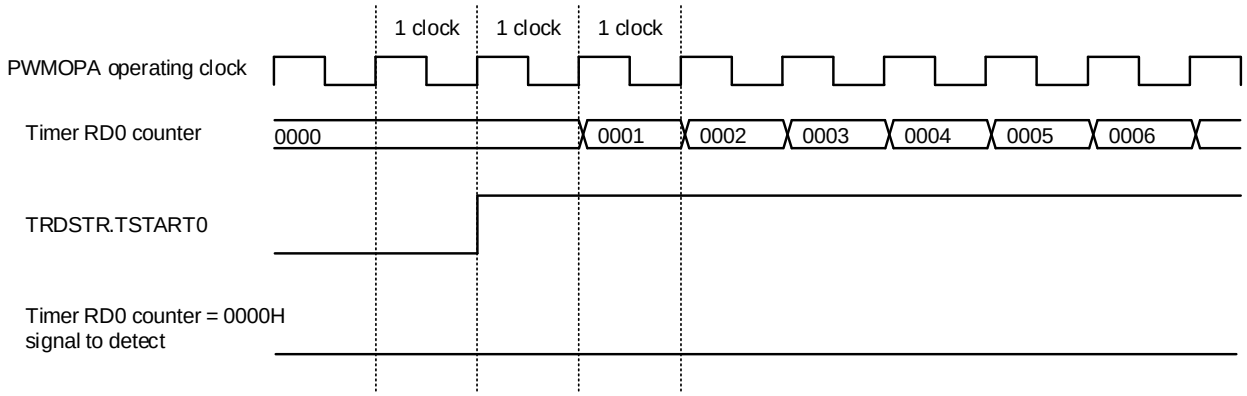
(3) Timing when the count value becomes 0000H if timer RDe is in the reset synchronous PWM mode

- Count value = 0000H and timer RDe starts counting.
  Output cutoff is not released.
- Timer RDe is counting and 0000H is written to the counter with software.
  Output cutoff is released.
- The counter becomes 0000H at compare match with TRDGRA0 register
  Output cutoff is released.

(4) Timing when the count value becomes 0000H if timer RDe is in the PWM3 mode

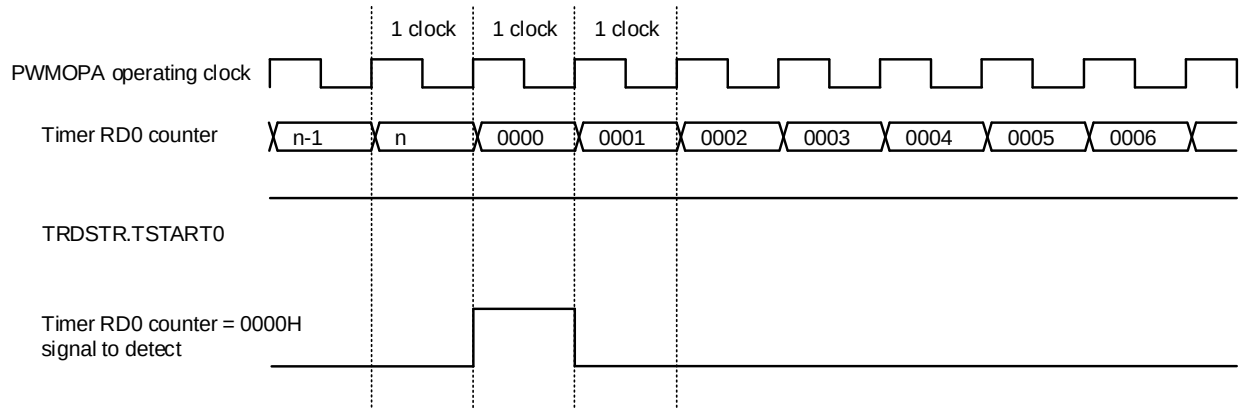
- Count value = 0000H and timer RDe starts counting.
  Output cutoff is not released.
- Timer RDe is counting and 0000H is written to the counter with software.
  Output cutoff is released.
- The counter becomes 0000H at compare match with TRDGRA0 register
  Output cutoff is released.

Figure 8-110. Judgement Timing for Count Value = 0000H (Timer RDe count starts when count value = 0000H)



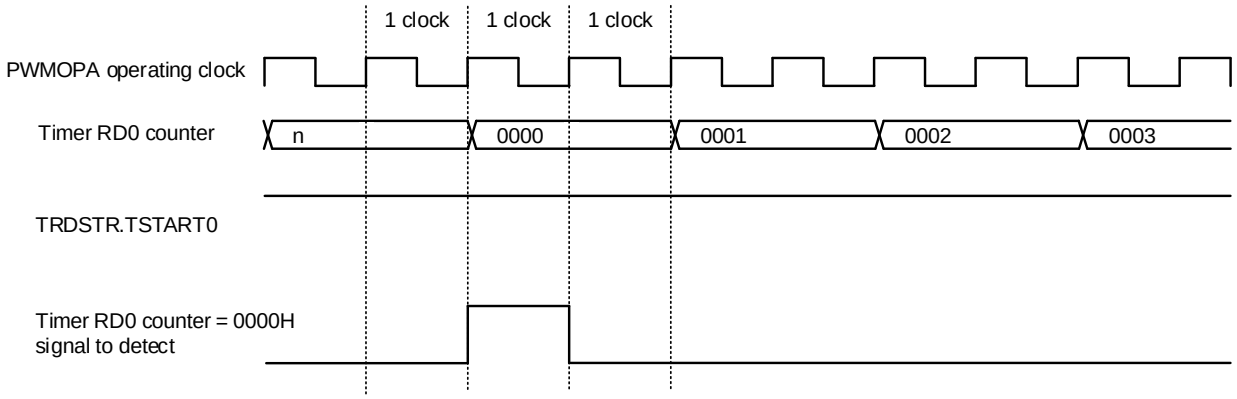
Remark The count value at the start of counting = 0000H is not judged.

Figure 8-111. Judgemen Timing for Count Value = 0000H
(count value becomes 0000H while counting with count source = operating clock)



Remark It is judged that the count value becomes 0000H.

Figure 8-112. Judgement Timing for Conut Value = 0000H
(count value becomes 0000H while counting with count source = operating clock/2)

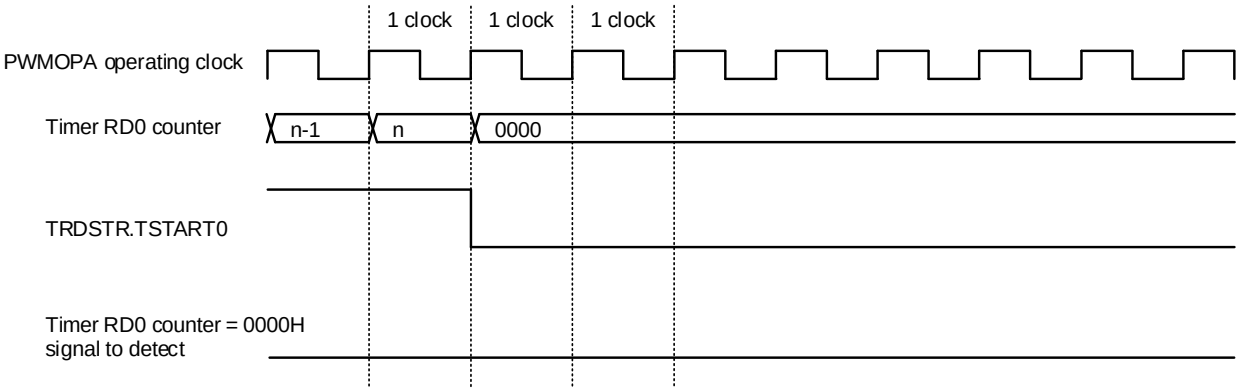


Remark It is judged that the count value becomes 0000H.

(5) When timer RDe count value = 0000H and timer RDe is stopped

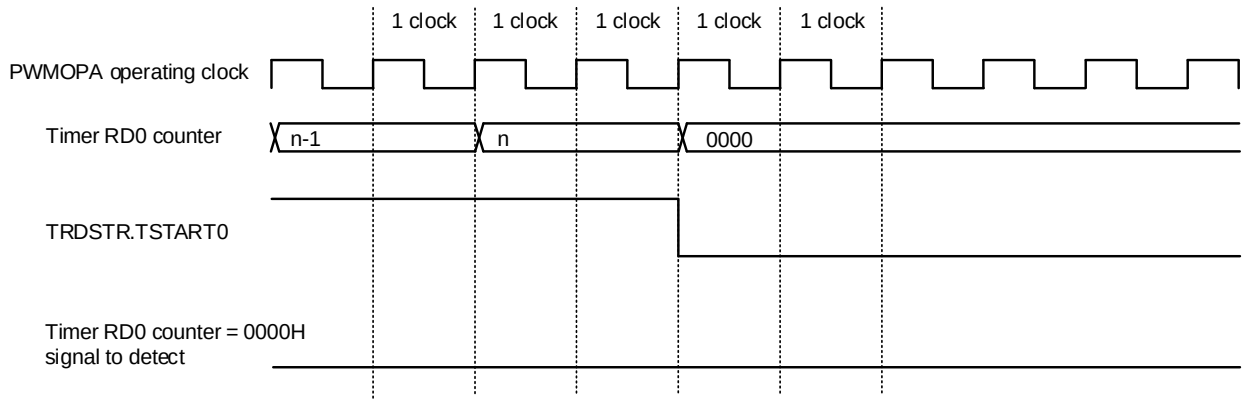
- If timer RDe stops simultaneously when the timer RDe count vlue becomes 0000H, it is not cutoff release timing.

Figure 8-113. Judgement Timing for Count Value = 0000H
(count source = operating clock, count stops simultaneously when timer RDe counter value becomes 0000H)



Remark Count value = 0000H is not judged.

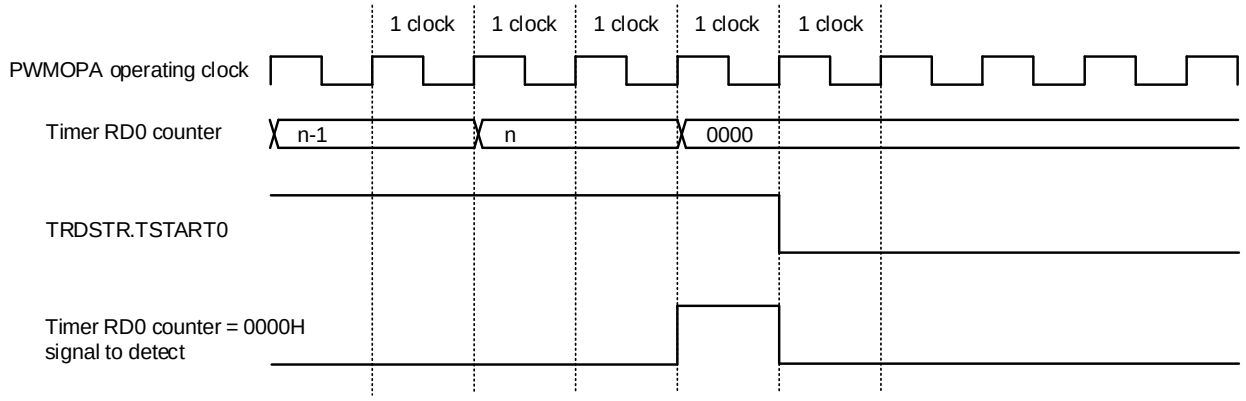
Figure 8-114. Judgement Timing for Count Value = 0000H
(count source = f_{CLK}/2, count stops simultaneously when timer RDe counter value becomes 0000H)



Remark Count value = 0000H is not judged.

- If timer RDe stops at the next one cycle after the timer RDe count value becomes 0000H, forced cutoff is released.

Figure 8-115. Judgement Timing for Count Value = 0000H
(count source = f_{CLK}/2, count stops at the next timing after timer RDe counter value becomes 0000H)



Remark It is judged that the count value becomes 0000H.

8.6.3.7 Setting Procedure

PWMOPA can operate in coordination with timer RDe.

The setting code should be added to the steps for setting up timer RDe. The procedure is as follows:

After setting the clock and mode of timer RDe:

(1) Set the PWMOPEN bit of the PER1 register to 1.
(2) Set the OPCTL0 register.
(3) Set the OPEDGE register.
(4) Set the OPDF0 and OPDF1 registers. Timer RDe starts operation.
(5) When an output cutoff factor is detected, the HZOF0 and HZOF1 bits of the OPSR register are set to "1".
(6) Release the cutoff state using the OPCTL0 register (software or hardware release can be selected with the HS_SEL bit).

Remarks 1. PWMOPA is a control module built by adding comparator 0 output, external interrupt 0 (INTP0), and event link controller (ELC) as triggers to the timer RDe cutoff function. Accordingly, operation of PWMOPA must always be performed together with timer RDe.

2. To operate the timer RDe function independently, do not set correlated registers of PWMOPA.

8.6.3.8 Cautions

(1) The following table lists the priorities when pulse output forced cutoff of timer RDe operates simultaneously with output cutoff of PWMOPA.

Table 8-31. Forced Cutoff Priorities

		Pin state control at output forced cutoff of PWMOPA			
		Prohibited	Hi-Z	Low-level	High-level
Pin state control at output forced cutoff of timer RDe	Prohibited	Prohibited	Hi-Z	Low-level	High-level
	Hi-Z	Hi-Z	Hi-Z	Low-level	High-level
	Low-level	Low-level	Hi-Z	Low-level	High-level
	High-level	High-level	Hi-Z	Low-level	High-level

(2) If timer RDe enters the pulse output forced cutoff state when PWMOPA is in the output cutoff state in the complementary PWM mode or extended complementary PWM mode, an output cutoff release edge may be entered to PWMOPA depending on the state of TRDIOC0.

(3) When output cutoff is triggered with an event link controller source, make sure to select software release (set the HS_SEL bit to 1) to release output cutoff.

(4) When output cutoff hazard control is selected, the timer RDe output via PWMOPA is delayed for one cycle of PWMOPA operating clock.

(5) If the timer RDe output pin via PWMOPA is set to timer RDe output when output cutoff hazard control is selected (HAZAD_SET bit is set to 1), switching between timer RDe output and port output is possible while timer RDe is counting.

(6) If the timer RDe output pin via PWMOPA is set to port operation, a hazard may occur upon output cutoff or cutoff release.

(7) Set the input enabled level period of comparator 0 and INTP0 to one cycle of PWMOPA operating clock or longer.

(8) When using reset synchronous PWM mode, complementary PWM mode, PWM3 mode, or extended complementary PWM mode, and when using PWM option unit A (PWMOPA), the TRDji[1,0] bits in the PWMDLY0 register for the target pin should be set to 00B (no delay) (i = 0, 1, j = A, B, C, D).

(9) When using the PWM function and extended PWM mode, if the PWM output is delayed by the PWMDLY0 register and PWM option unit A (PWMOPA) is used, the level held in the output delay circuit just before the output is cutoff is output from the target pin (output delay enabled) for the number of pulses equal to the delay setting value −1 after the forced cutoff by PWMOPA is released.

8.7 Interrupt Decimation Module (TRDMBK)

Timer RDe interrupt decimation module (TRDMBK) outputs a decimated signal with 0 to 31 in synchronization with the cycle of the input signal for a total of 3 signals, 2 input signals of interrupt request and 1 input signal of A/D trigger signal.

It can be used only in extended complementary PWM mode.

8.7.1 Overview

- 4 input signals can be enables/disables for each signal
- This circuit can be decimated 0 to 31 times with the decimation counter and compare register
- Decimation counter controls the decimation signal in synchronaization with the input signal
- Decimation module can be used only in extended complementary PWM model

Figure 8-116 shows the interrupt decimation module (TRDMBK) block diagram, and **Table 8-32** lists the input/output configuration.

Figure 8-116. TRDMBK Block Diagram

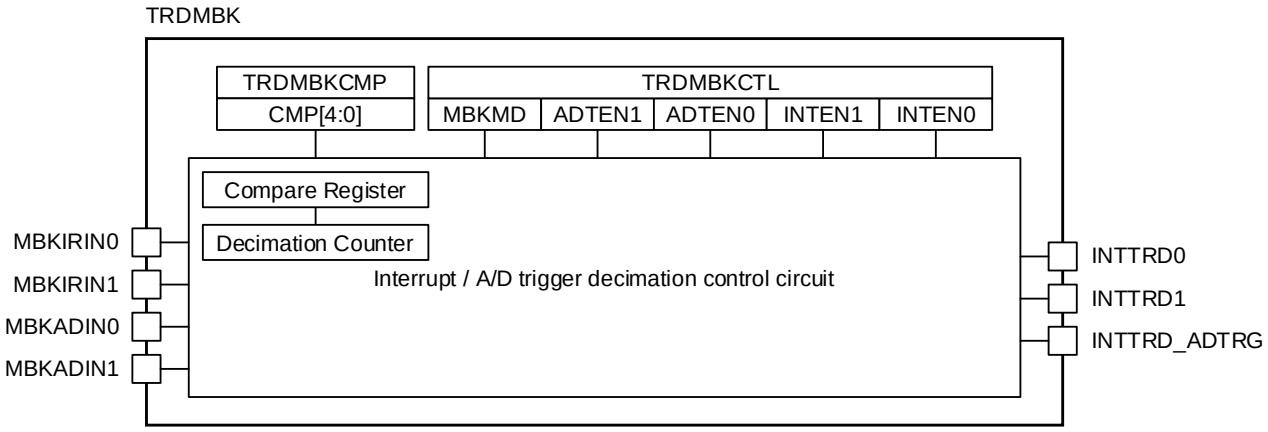


Table 8-32. TRDMBK Input/Output Configuration

Signal	Input/Output	Function
MBKIRIN0	Input	Timer RDe interrupt request signal 0 (OVF signal of TRD1)
MBKIRIN1	Input	Timer RDe interrupt request signal 1, and TRDMBK source signal (UDF signal of TRD1)
MBKADIN0	Input	Timer RDe A/D trigger signal 0 (Compare match signal with TRDADTC0)
MBKADIN1	Input	Timer RDe A/D trigger signal 1 (Compare match signal with TRDADTC1)
INTTRD0	Output	Timer RDe interrupt request signal 0 with decimation control
INTTRD1	Output	Timer RDe interrupt request signal 1 with decimation control
INTTRD_ADTRG	Output	Timer RDe A/D trigger signal with decimation control

8.7.2 TRDMBK Registers

Address	Register Name	Symbol	After Reset	Access Size
F024EH	Timer RDe decimation control register	TRDMBKCTL	00H Note	8
F024FH	Timer RDe decimation count register	TRDMBKCMP	00H Note	8

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

8.7.2.1 Timer RDe Decimation Control Register (TRDMBKCTL)

When 1 is set to any of INTEN0, INTEN1, ADTEN0, and ADTEN1, the decimation counter operation starts.

Figure 8-117. Format of Timer RDe Decimation Control Register (TRDMBKCTL)

Address: F024EH After Reset: 00H [Note]

Symbol	7	6	5	4	3	2	1	0
TRDMBKCTL	0	0	0	MBKMD	ADTEN1	ADTEN0	INTEN1	INTEN0

Bits 7 to 5	Nothing is assigned	R/W
–	The write value must be 0.	–

MBKMD	Select decimation mode	R/W
0	Disable decimation at first period after starting operation	R/W
1	Enable decimation at first period after starting operation	

ADTEN1	A/D trigger 1 (compare match signal with TRDADTC1) decimation enable	R/W
0	Disable A/D trigger 1 input (MBKADIN1)	R/W
1	Enable A/D trigger 1 input (MBKADIN1)	

ADTEN0	A/D trigger 0 (compare match signal with TRDADTC0) decimation enable	R/W
0	Disable A/D trigger 0 input (MBKADIN0)	R/W
1	Enable A/D trigger 0 input (MBKADIN0)	

INTEN1	Interrupt request 1 (UDF signal of TRD1) decimation enable	R/W
0	Disable interrupt request 1 input (MBKIRIN1)	R/W
1	Enable interrupt request 1 input (MBKIRIN1)	

INTEN0	Interrupt request 0 (OVF signal of TRD1) decimation enable	R/W
0	Disable interrupt request 0 input (MBKIRIN0)	R/W
1	Enable interrupt request 0 input (MBKIRIN0)	

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

8.7.2.2 Timer RDe Decimation Count Register (TRDMBKCMP)

Figure 8-118. Format of Timer RDe Decimation Count Register (TRDMBKCMP)

Address: F024FH After Reset: 00H [Note]

Symbol	7	6	5	4	3	2	1	0
TRDMBKCMP	0	0	0	CMP[4:0]				

Bits 7 to 5	Nothing is assigned	R/W
–	The write value must be 0.	–

CMP[4:0]	Decimation count setting	R/W
00000B	Decimation count is 0 (Decimation module is disabled)	R/W
00001B	Decimation count is 1 (Once every 2 times)	
00010B	Decimation count is 2 (Once every 3 times)	
:	:	
11110B	Decimation count is 30 (Once every 31 times)	
11111B	Decimation count is 31 (Once every 32 times)	

Note The value after reset is undefined when FRQSEL4 = 1 in the user option byte (000C2H/040C2H) 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.

8.7.3 Operation

TRDMBK contains 5-bit decimation counter. Decimation counter is decremented after reading and setting the value set in CMP[4:0] bits in the TRDMBKCMP register. When the decimation counter becomes 0, the value set in CMP[4:0] bits are read again and set, and the decimation counter is repeatedly decremented.

The TRDMBKCMP register can always be rewritten.

Figure 8-119. TRDMBK Decimation Counter Control

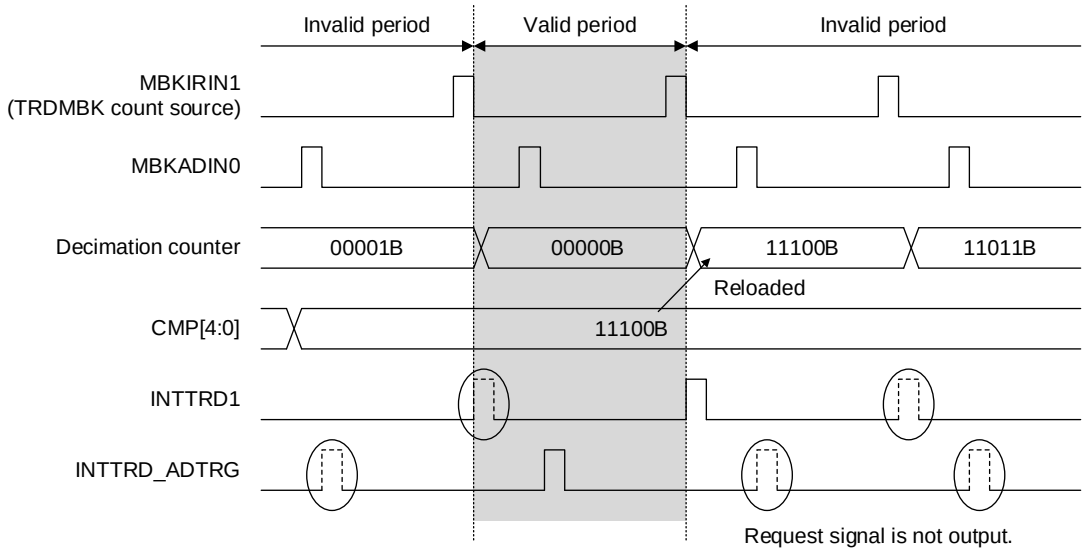
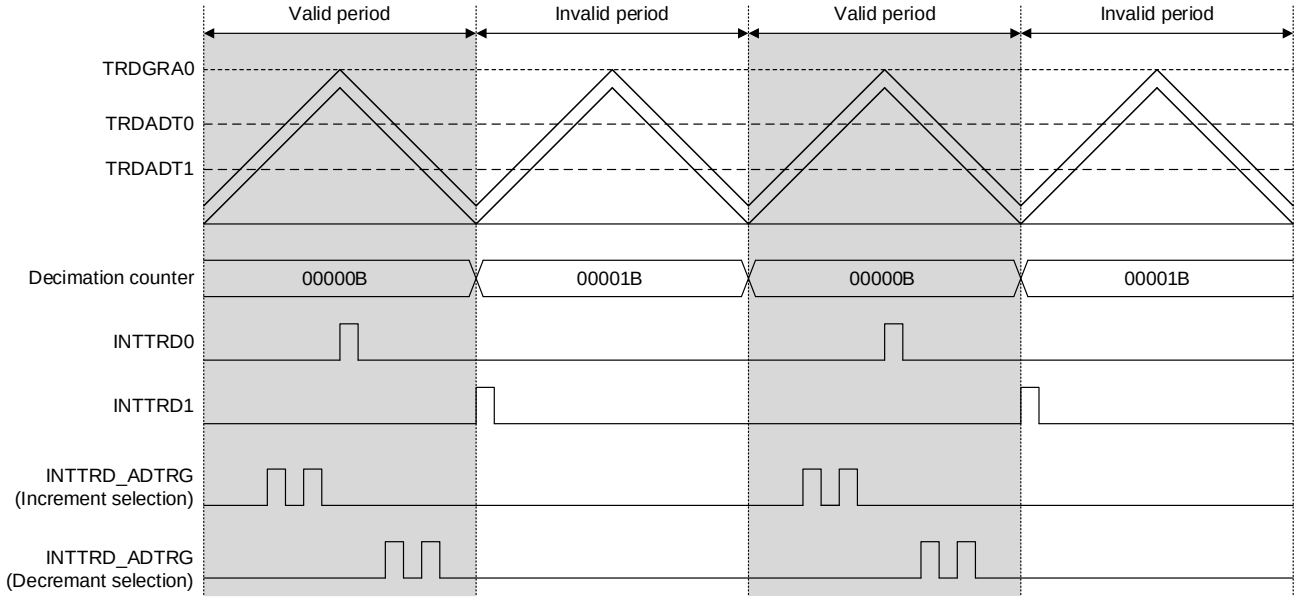


Figure 8-120. Operation Example in Extended Complementary PWM Mode (Decimated once)



With the MBKMD bit of the TRDMBKCTL register, it is possible to set whether to enable or disable the decimated output in the first cycle at the start of operation. When the MBKMD bit is set to “1”, the interrupt signal and AD trigger signal of the first cycle of the timer source are output.

Figure 8-121. Operation Example of Decimated once in Extended Complementary PWM Mode (TRDMBKCTL.MBKMD=0)

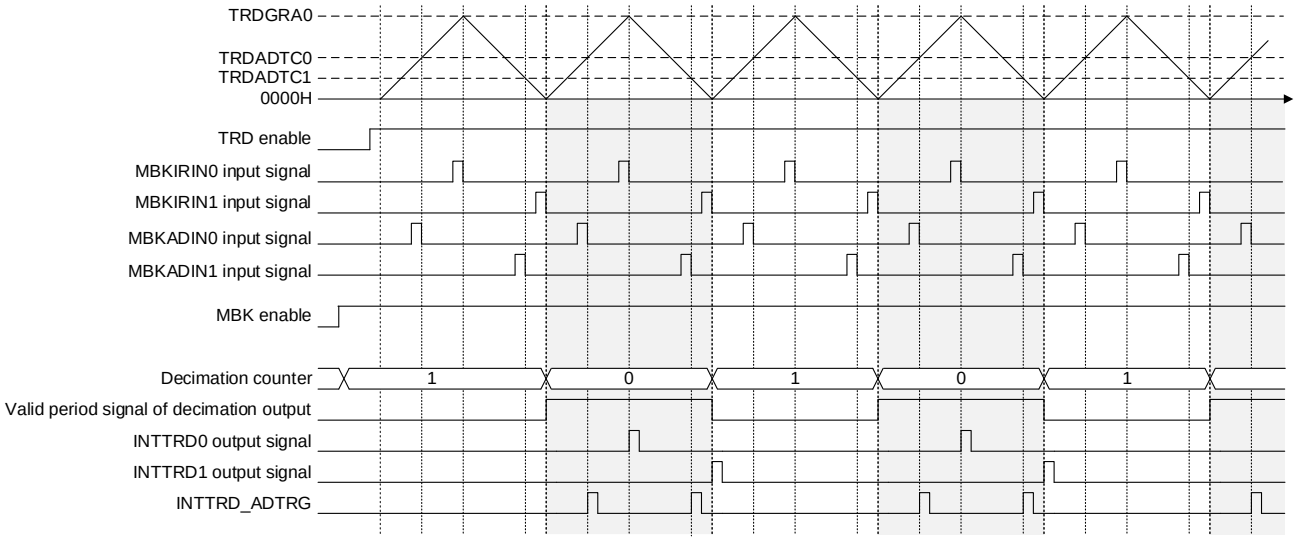
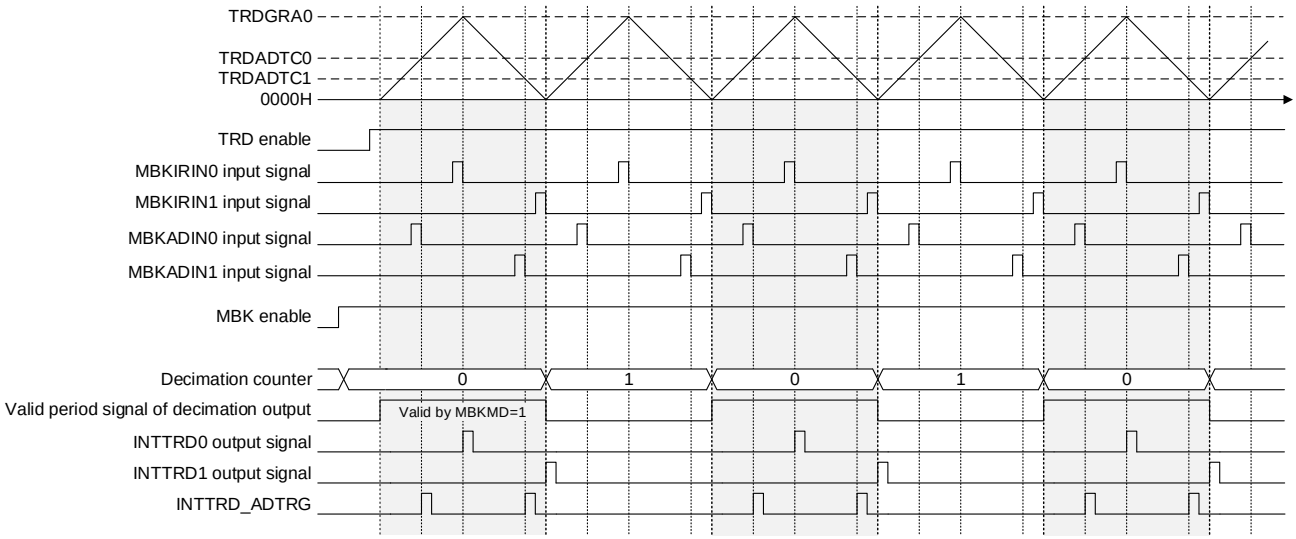


Figure 8-122. Operation Example of Decimated once in Extended Complementary PWM Mode (TRDMBKCTL.MBKMD=1)



Remarks TRD enable: It becomes 1 when TSTARTi bit in the TRDSTR register is set to 1.
MBK enable: It becomes 1 when any of INTEN0, INTEN1, ADTEN0, ADTEN1 in the TRDMBKCTL register is set to 1.

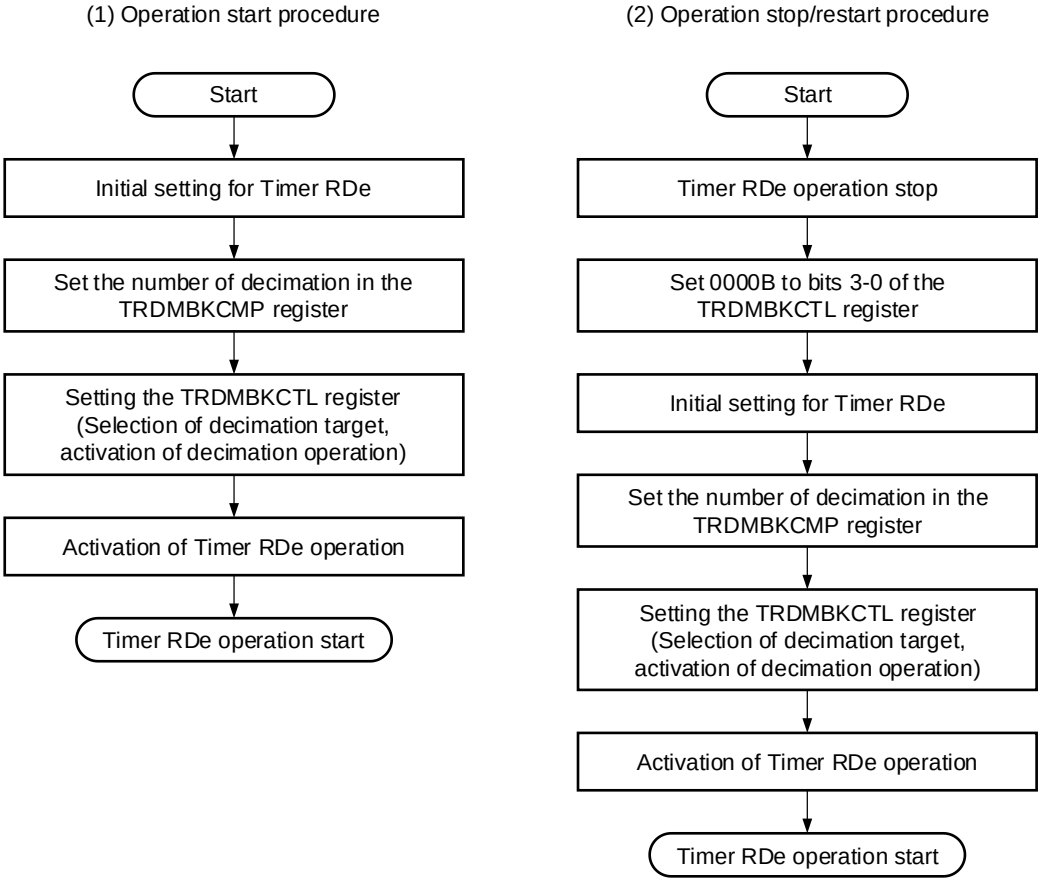
The above diagrams applied under the following conditions:

- ADMD0 bit in the TRDADCR register is 0. (A/D trigger 0 compares during up counting.)
- ADMD1 bit in the TRDADCR register is 1. (A/D trigger 1 compares during down counting.)
- Bits CMP[4:0] in the TRDMBKCMP register are 00001B. (Decimation count is 1.)

8.7.4 Setting Procedure

The procedure for setting the interrupt decimation circuit and setting the Timer RDe is shown below. The counter in the decimation circuit is initialized when bits 3 to 0 of TRDMBKCTL is set from 0000B to other than 0000B. Be sure to set bits 3 to 0 of the TRDMBKCTL register to 0000B when stopping the operation of the Timer RDe, changing the setting of the Timer RDe, or changing the setting of the decimation circuit.

Figure 8-123. Setting Procedure of the Interrupt Decimation Circuit and Timer RDe



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.

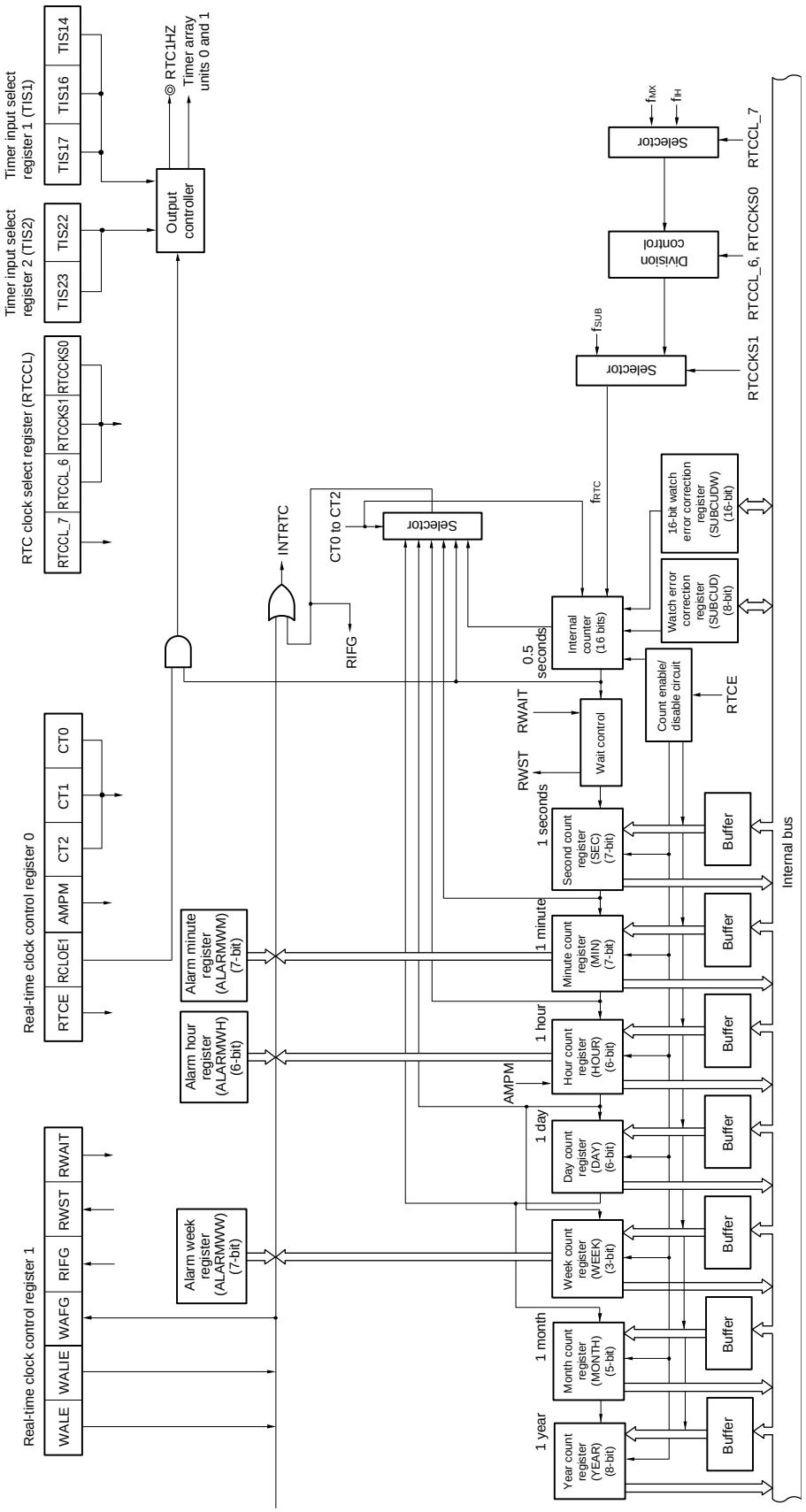
9.2 Configuration of Real-time Clock

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

Table 9-2. Real-time Clock Register Configuration

Address	Register Name	Symbol	After Reset	Access Size
F02C0H	Peripheral enable register 0	PER0	00H	1, 8
F00F3H	Operation speed mode control register	OSMC	00H	8
F0075H	Timer input select register 1	TIS1	00H	8
F007AH	Timer input select register 2	TIS2	00H	8
F02C5H	RTC clock select register	RTCCL	00H	1, 8
FFF54H	16-bit watch error correction register	SUBCUDW	0000H	16
FFF92H	SEC count register	SEC	00H	8
FFF93H	MIN count register	MIN	00H	8
FFF94H	HOUR count register	HOUR	12H	8
FFF95H	WEEK count register	WEEK	00H	8
FFF96H	DAY count register	DAY	01H	8
FFF97H	MONTH count register	MONTH	01H	8
FFF98H	YEAR count register	YEAR	00H	8
FFF99H	Watch error correction register	SUBCUD	00H	8
FFF9AH	Alarm MIN register	ALARMWM	00H	8
FFF9BH	Alarm HOUR register	ALARMWH	12H	8
FFF9CH	Alarm WEEK register	ALARMWW	00H	8
FFF9DH	Real-time clock control register 0	RTCC0	00H	1, 8
FFF9EH	Real-time clock control register 1	RTCC1	00H	1, 8

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: F02C0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of input clock supply to real-time clock (RTC)
0	Stops input clock supply. • SFR used by the real-time clock (RTC) cannot be written. • The real-time clock (RTC) is in the reset status.
1	Enables input clock supply. • SFR used by the real-time clock (RTC) can be read/written.

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.**
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 bit 6 to 0.**

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. (See **Table 23-1, Table 23-2** and **Table 23-3** 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	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.

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 RL78/F25 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.

3. RL78/F22 products do not have this register.

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: F02C5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCCL	RTCCL_7	RTCCL_6	0	0	0	0	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}).

Caution 24-, 32-pin products do not have a subsystem clock (f_{SUB}), so it should not be selected.

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. • **Table 9-3** 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 RCLOE1 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.	

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”. 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)

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. 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. Be sure to write “1” to it to read or write the counter value. As the internal counter (16 bits) is continuing to run, complete reading or writing within one second and turn back to 0. 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** When reading or writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt. When the internal counter (16 bits) overflowed while RWAIT = 1, it keeps the event of overflow until RWAIT = 0, then counts up. However, when it wrote a value to second count register, it will not keep the overflow event.	

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”.

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 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 9-3 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 9-3. Displayed Time Digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 0)	
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. • When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H • 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. Range of correction value: (when F12 = 0) 2, 4, 6, 8, … , 120, 122, 124 (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	0	0	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. • When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H • 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. Range of correction value: (when F12 = 0) 2, 4, 6, 8, … , 8184, 8186, 8188 (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

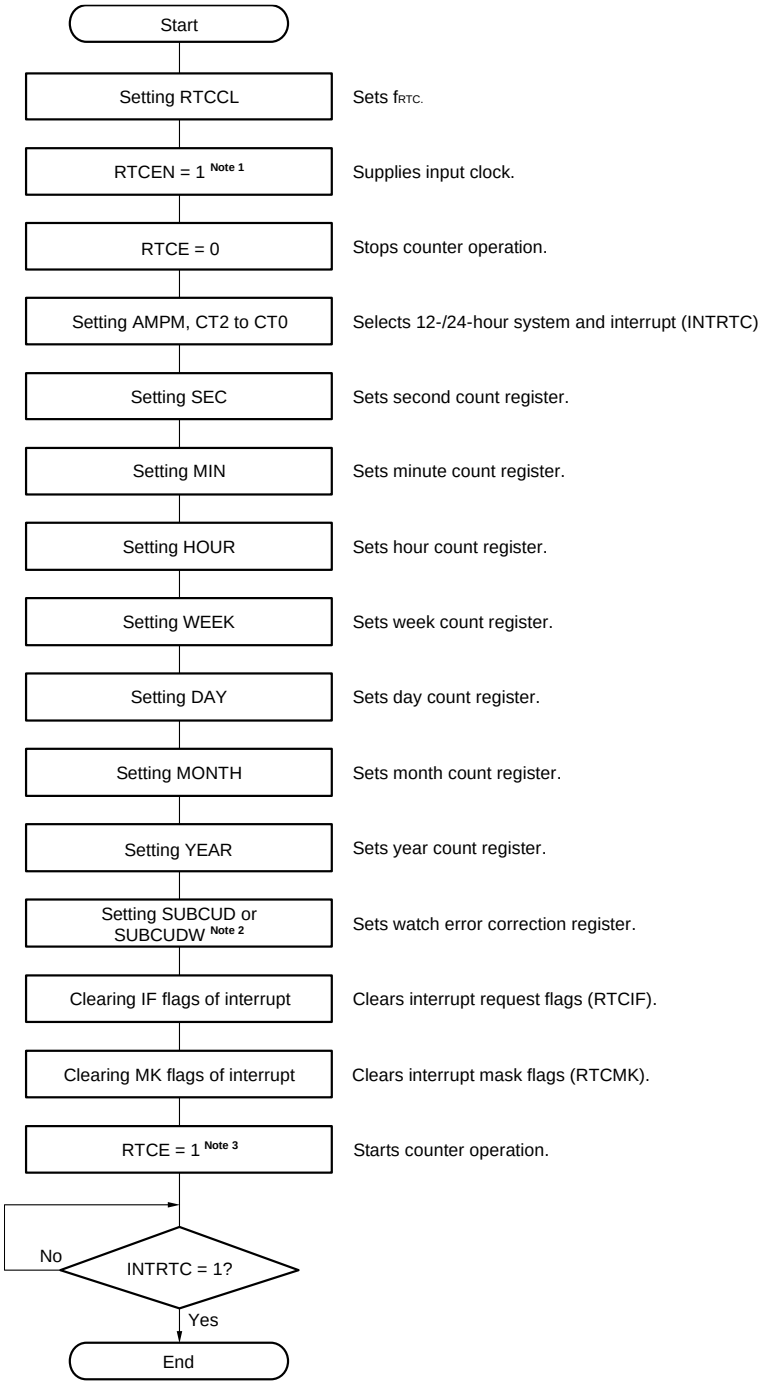
Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday 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.

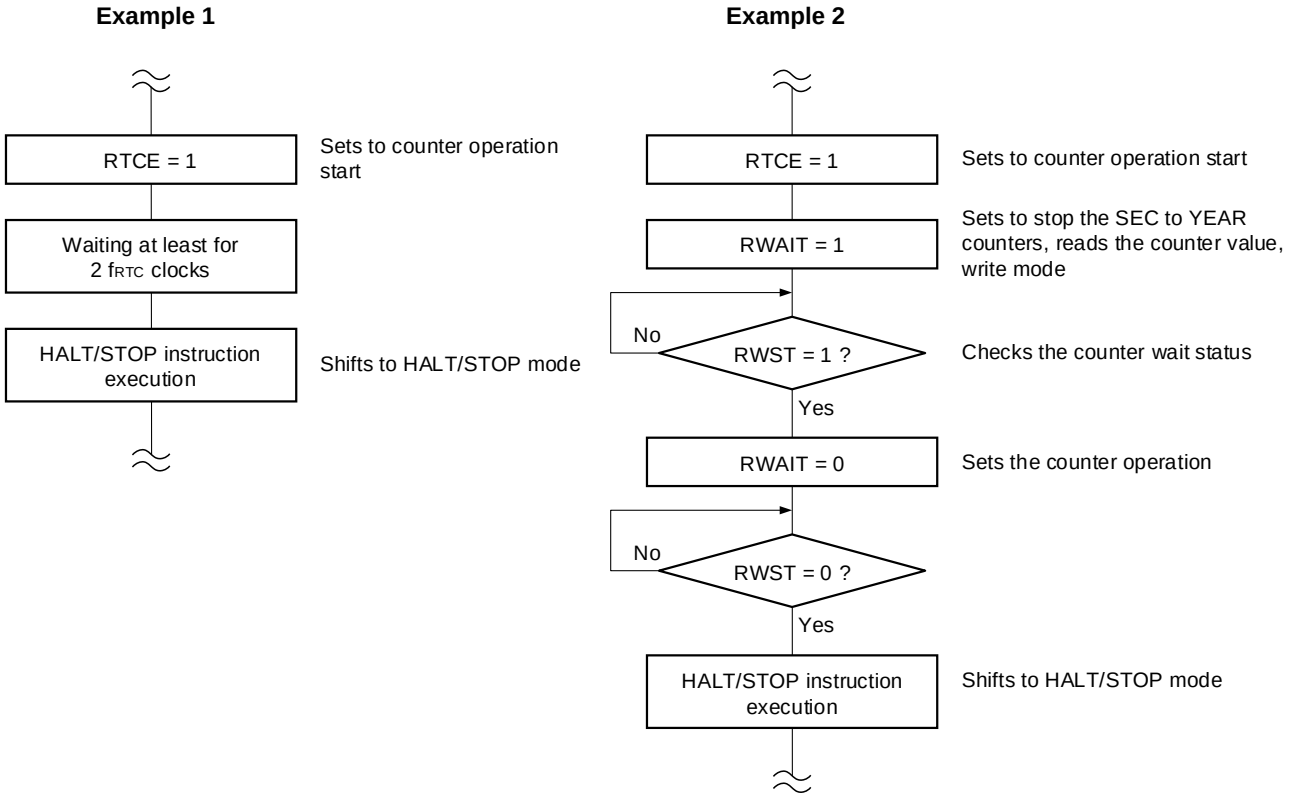
9.4.2 Shifting to HALT/STOP Mode After Starting Operation

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

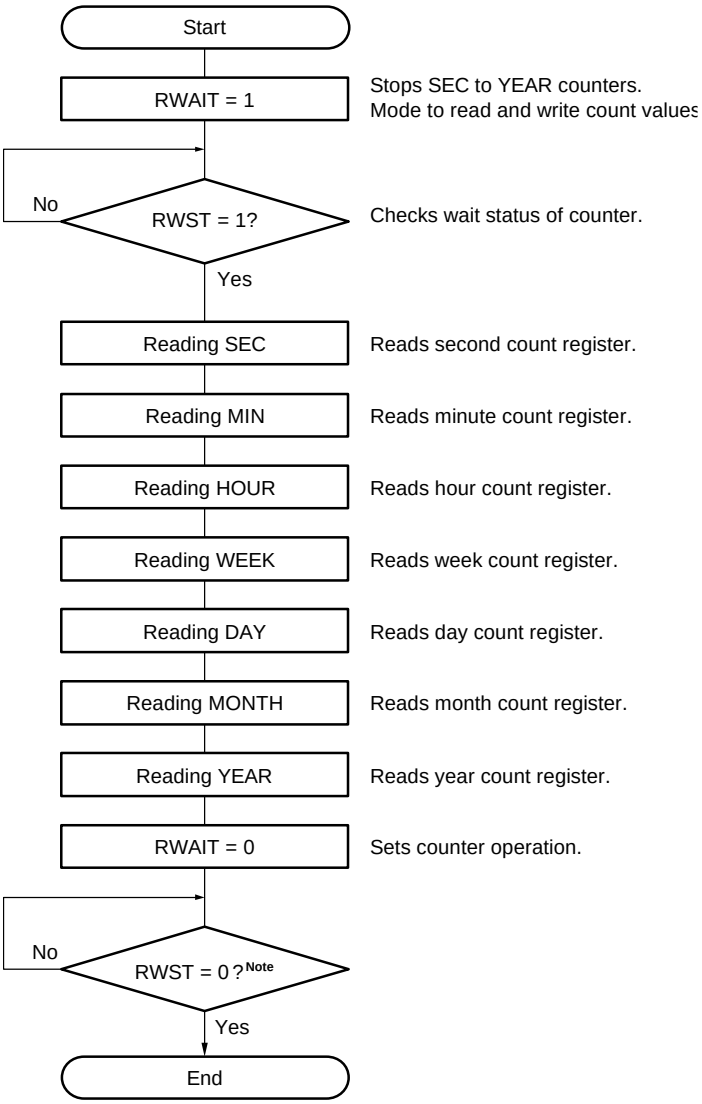


9.4.3 Reading/Writing Real-time Clock

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



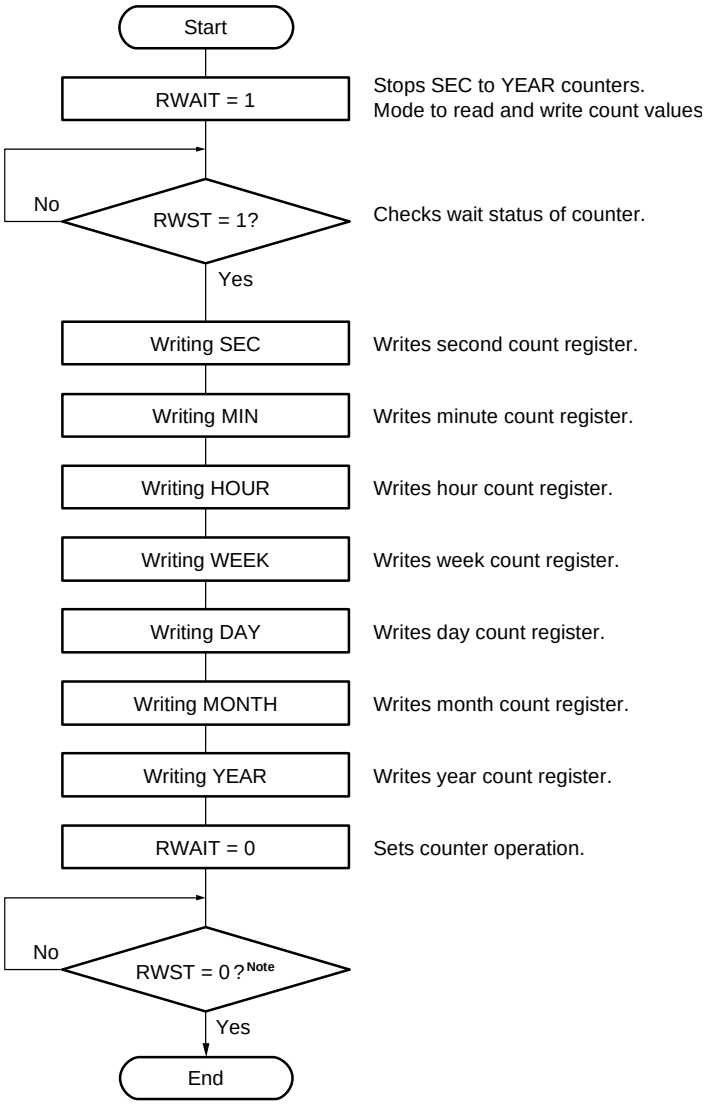
Note Be sure to confirm that RWST = 0 before setting HALT/STOP mode.

Caution **Complete the series of process of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second. When reading to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.**

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.

Also, it is not necessary to read all the registers, and only some of the 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. When writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.

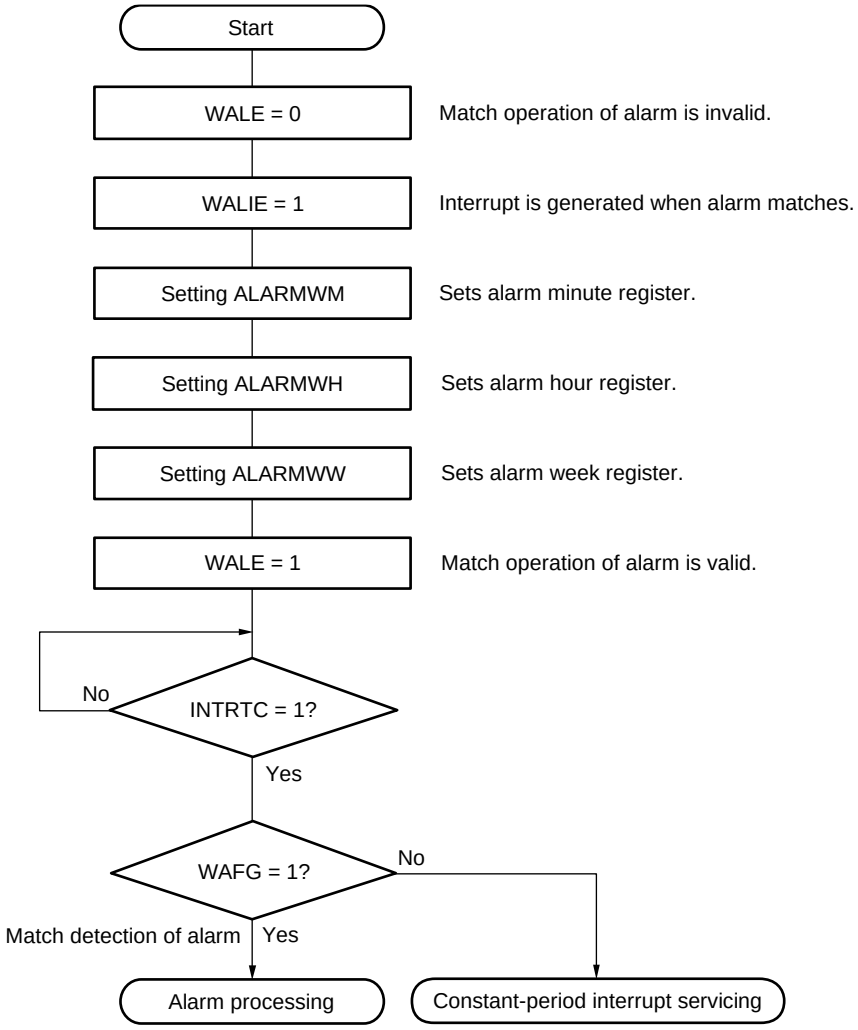
2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counter operates (RTCE = 1), rewrite the values of the MIN register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags 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

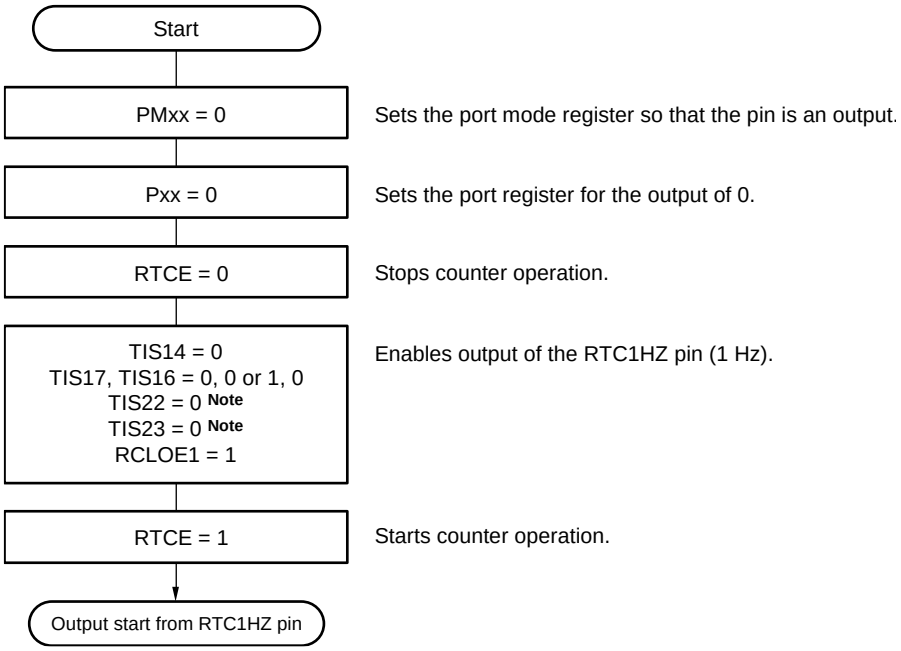


Remarks 1. The alarm minute register (ALARMWM), alarm hour register (ALARMWH), and alarm week register (ALARMWW) may be written in any sequence.

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/F25 products.

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.

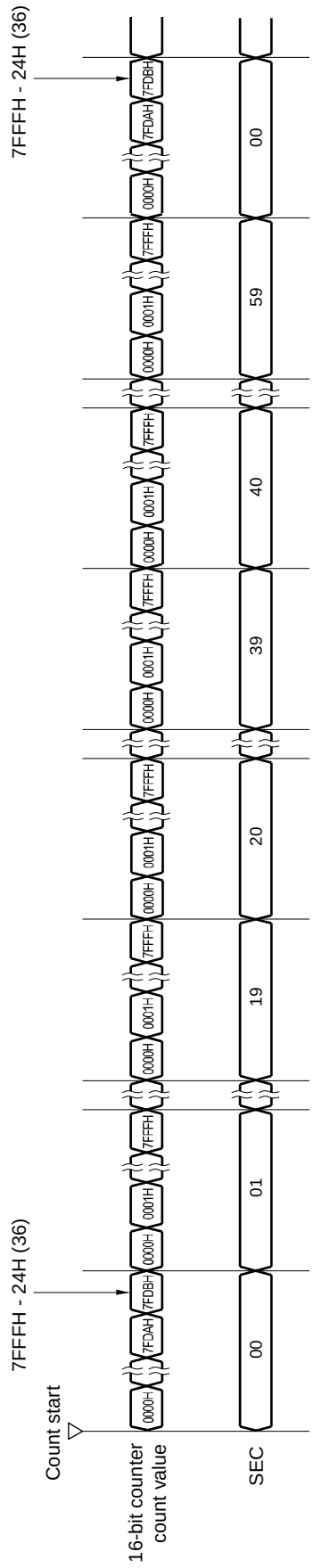
Calculate (F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) from the correction value.

− {(/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 9-27 shows the operation example 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, 0, 1, 1, 1, 0).

Figure 9-27. Operation Example of Watch Error Correction


7FFFH - 24H (36)
Count start
7FFFH - 24H (36)
16-bit counter count value
SEC
0000H
7FDAH
7FDBH
0000H
0001H
7FFFH
0000H
0001H
7FFFH
0000H
0001H
7FFF

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	RL78/F22, F25 48, 64, 80, and 100-pin products	RL78/F22 24-pin and 32-pin products
PCLBUZ0	√	–

Caution Most of the following descriptions in this chapter use the RL78/F25 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.

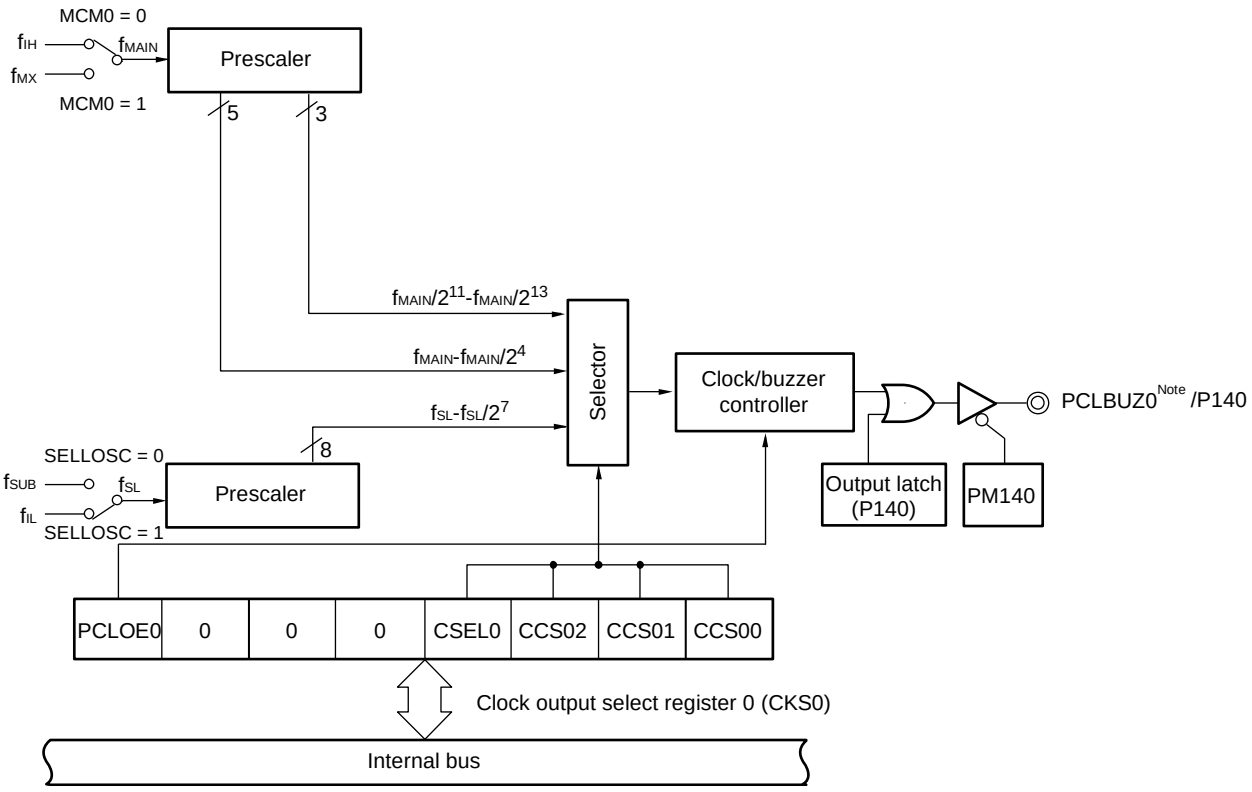
One pin can be used to output a clock or buzzer sound.

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 37** and **CHAPTER 38 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) • Port mode register 14 (PM14) • Port register 14 (P14)

10.3 Registers Controlling Clock Output/Buzzer Output Controller

Table 10-2. Clock Output/Buzzer Output Controller Register Configuration

Address	Register Name	Symbol	After Reset	Access Size
FFFA5H	Clock output select register 0	CKS0	00H	1, 8
F02CAH	Clock select register	CKSEL	00H	1, 8

Remark See **10.3.3 Port mode register 14** for the port mode register 14 (PM14) and port register 14 (P14).

10.3.1 Clock Output Select Register 0 (CKS0)

This register sets output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZ0), and sets 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) 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} = 40 MHz	f_{MAIN} = 48 MHz	f_{MAIN} = 64 MHz	f_{MAIN} = 80 MHz
0	0	0	0	f_{MAIN}	5 MHz	10 MHz Note	Setting prohibited Note	Setting prohibited 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	Setting prohibited Note	Setting prohibited Note
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz	8 MHz	10 MHz Note	12 MHz Note	16 MHz Note	Setting prohibited Note
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz	4 MHz	5 MHz	6 MHz	8 MHz	10 MHz Note
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz	2 MHz	2.5 MHz	3 MHz	4 MHz	5 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.77 kHz	15.63 kHz	19.53 kHz	23.44 kHz	31.25 kHz	39.06 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz	9.77 kHz	11.72 kHz	15.63 kHz	19.53 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz	4.88 kHz	5.86 kHz	7.81 kHz	9.77 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 1.875 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 234.38 Hz (f_{IL})							
1	1	1	1	$f_{SL}/2^7$	256 Hz (f_{SUB}) or 117.19 Hz (f_{IL})							

Note Use the output clock within a range of 16 MHz. See the AC characteristics in **CHAPTER 37** and **CHAPTER 38 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 RDe, 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: F02CAH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	<2>	1	<0>
CKSEL	FPLLDIV	FMAINDIV1	FMAINDIV0	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 and stopping the low speed on-chip oscillator
1	Selects f_{IL} Note 2 and running the low speed on-chip oscillator

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 RL78/F22 24-pin and 32-pin products, set the SELLOSC bit to 1.

10.3.3 Port Mode Register 14 (PM14)

This register sets 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 this register 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.

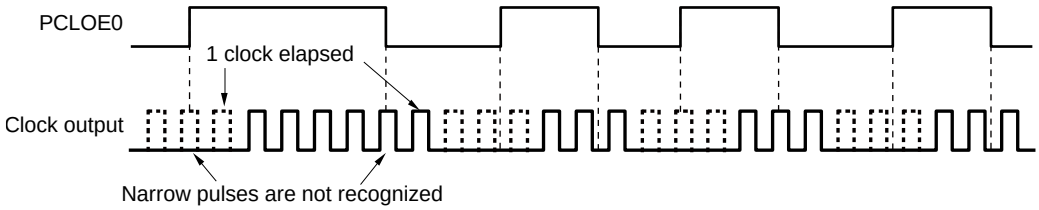
<1> Set the bit in the port mode register 14 (PM14) and the bit in the port register 14 (P14) corresponding to the port used as a PCLBUZ0 pin.

<2> 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).

<3> 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 25 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.

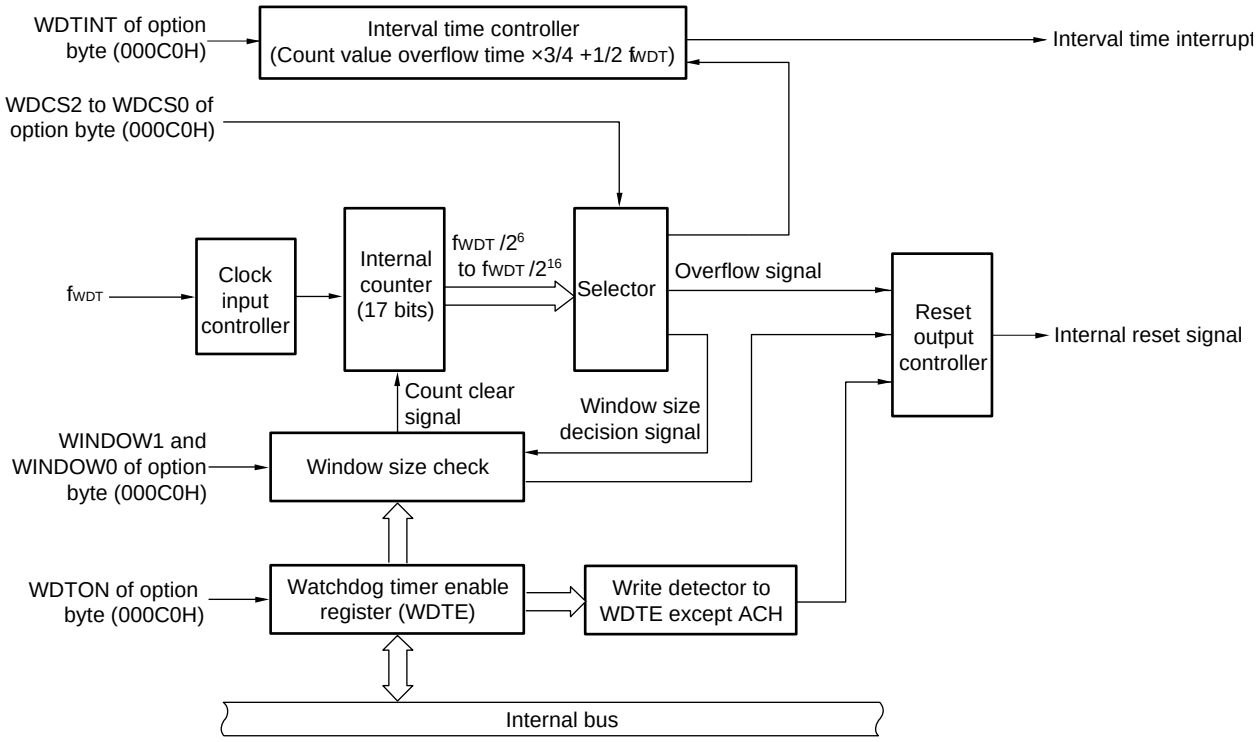
Caution Set the same value as 000C0H to 040C0H when the boot swap operation is used because 000C0H is replaced by 040C0H.

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 32 OPTION BYTE**.

Figure 11-1. Block Diagram of Watchdog Timer



11.3 Register to Controlling Watchdog Timer

The following register is used to control the watchdog timer.

Table 11-3. Watchdog Timer Register Configuration

Address	Register Name	Symbol	After Reset	Access Size
FFFABH	Watchdog timer enable register	WDTE	1AH or 9AH Note	8

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.

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.

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 32 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 32 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 32 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 $2/f_{WDT}$ 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 11-4. 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	$2^{13}/f_{WDT}$ (474.89 ms)
1	1	0	$2^{14}/f_{WDT}$ (949.79 ms)
1	1	1	$2^{16}/f_{WDT}$ (3799.18 ms)

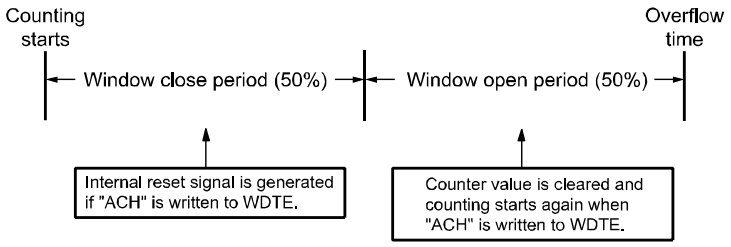
Remark f_{WDT}: WDT-dedicated low-speed on-chip oscillator clock frequency

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 11-5. 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% Note
1	1	100%

Note When the window open period is set to 75%, clearing the counter of the watchdog timer (writing ACH to WDTE) must proceed outside the corresponding period from among those listed below, over which clearing of the counter is prohibited (for example, confirming that the interval timer interrupt request flag (WDTIIF) of the watchdog timer is set).

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{WDT} = 17.25 kHz (MAX.))	Period over which clearing the counter is prohibited when the window open period is set to 75%
0	0	0	$2^6/f_{WDT}$ (3.71 ms)	1.85 ms to 2.51 ms
0	0	1	$2^7/f_{WDT}$ (7.42 ms)	3.71 ms to 5.02 ms
0	1	0	$2^8/f_{WDT}$ (14.84 ms)	7.42 ms to 10.04 ms
0	1	1	$2^9/f_{WDT}$ (29.68 ms)	14.84 ms to 20.08 ms
1	0	0	$2^{11}/f_{WDT}$ (118.72 ms)	59.36 ms to 80.32 ms
1	0	1	$2^{13}/f_{WDT}$ (474.89 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{WDT}$ (949.79 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{WDT}$ (3799.18 ms)	1899.59 ms to 2570.04 ms

(**Remark** and **Caution** are listed in the next page.)

Remark f_{WDT} (WDT-dedicated low-speed on-chip oscillator frequency)
= (MIN.) 12.75 kHz / (TYP.) 15 kHz / (MAX.) 17.25 kHz

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 + 1/2 f_{WDT} is reached.

Table 11-6. 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 12-BIT A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

RL78/F25			RL78/F25, F22	RL78/F22	
100 pin	80 pin	64 pin	48 pin	32 pin	24 pin
29 ch ANI0 to ANI28 Note	23 ch ANI0 to ANI15, ANI22 to ANI28 Note	22 ch ANI0 to ANI14, ANI22 to ANI28 Note	17 ch ANI0 to ANI10, ANI22 to ANI27 Note	8 ch ANI0 to ANI5, ANI22 to ANI23 Note	4 ch ANI0 to ANI2, ANI23 Note

Note ANI0 to ANI15: High speed pin, ANI16 to ANI28: Normal speed pin

12.1 Overview

This MCU incorporates one unit of a 12-bit successive approximation A/D converter. And this A/D converter supports dual sampling and self-diagnosis function.

Table 12-1 lists the specifications of the 12-bit A/D converter and **Table 12-2** lists the functions of the 12-bit A/D converter. **Figure 12-1** shows a block diagram of the 12-bit A/D converter.

Table 12-1. Specifications of 12-Bit A/D Converter

Item	Description
Number of units	One unit
Input channels	Up to 29 channels (depending on the product)
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time Note 4	1.125 μs per channel (fastest conversion time) (when A/D conversion clock ADCLK = 40 MHz)
A/D conversion clock	Peripheral hardware clock f_{CLK} Note 1 and A/D conversion clock ADCLK Note 1 can be set so that the frequency division ration should be one of the following. f_{CLK} to ADCLK frequency division ration = 1:1, 2:1, 4:1, 8:1
Data registers	• 30 registers for each analog input (including V_{TSCAP} level input) The value obtained by adding up A/D-converted results is stored as a value in the number of bit for conversion accuracy + 2 bits/4 bits Note 2 in the A/D data registers in A/D-converted value addition mode. • One register for internal reference voltage (V_{BGR}) • Oner register for self-diagnosis
Operating modes Note 3	• Single scan mode: A/D conversion is performed only once for the analog inputs or the internal reference voltage (V_{BGR}). • Continuous scan mode: A/D conversion is performed repeatedly for the analog inputs in ascending channel order. • Group scan mode: Analog inputs are divided into groups A and B, and A/D conversion with synchronous trigger is performed only once for all analog inputs selected in each group. When the group priority operation is enabled and high priority group trigger is generated during the scan of the low priority group, the scan of the low priority group is suspended and the scan of the high priority group is started. The priority order is Group A (high) > Group B (low). Setting is possible to specify whether the scan of the low priority group is restarted after completion of the scan of the priority group. Setting is also possible to specify whether to restart the scan from the first channel of the selected channels or from a channel for which A/D conversion has not been made.
Conditions for A/D conversion start	• Software trigger • Synchronous trigger Triggered by the event link controller (ELC) Note 5 and timer function (TAU0, RTC, Timer RDe, Timer RJ)
Functions	• Simultaneous dual sampling function (built-in channel-dedicated sample-and-hold circuit on ANI1 and ANI2) • Variable sampling state count (selectable per channel) • Self-diagnosis of 12-bit A/D converter • Disconnection detection assist function (precharge or discharge) • Selectable A/D-converted value addition mode or average mode • INTAD can activate the data transfer controller (DTC).
Interrupt sources	• A/D scan end interrupt request (INTAD) can be generated on completion of single scan. • A/D conversion completion interrupt for Scan Group-B (INTADGB) can be generated on completion of group B scan.
Reference voltage	• Reference voltage on the high-potential side: AV_{REFP} • Reference voltage on the low-potential side: AV_{REFM}

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

Notes 1. The frequency divider for the peripheral hardware clock f_{CLK} and A/D conversion clock ADCLK is set in the ADCKS register. Note that the ADCLK cannot be set to a frequency less than 2 MHz.

2. The number of extended bits during addition differs depending on the A/D conversion accuracy and the addition count.
2-bit extension: A/D conversion accuracy = 1-time to 4-time conversion (addition zero to three times)
4-bit extension: A/D conversion accuracy = 16-time conversion (addition 15 times)

3. When the internal reference voltage (V_{BGR}) is selected, do not use the continuous scan mode and the group scan mode.

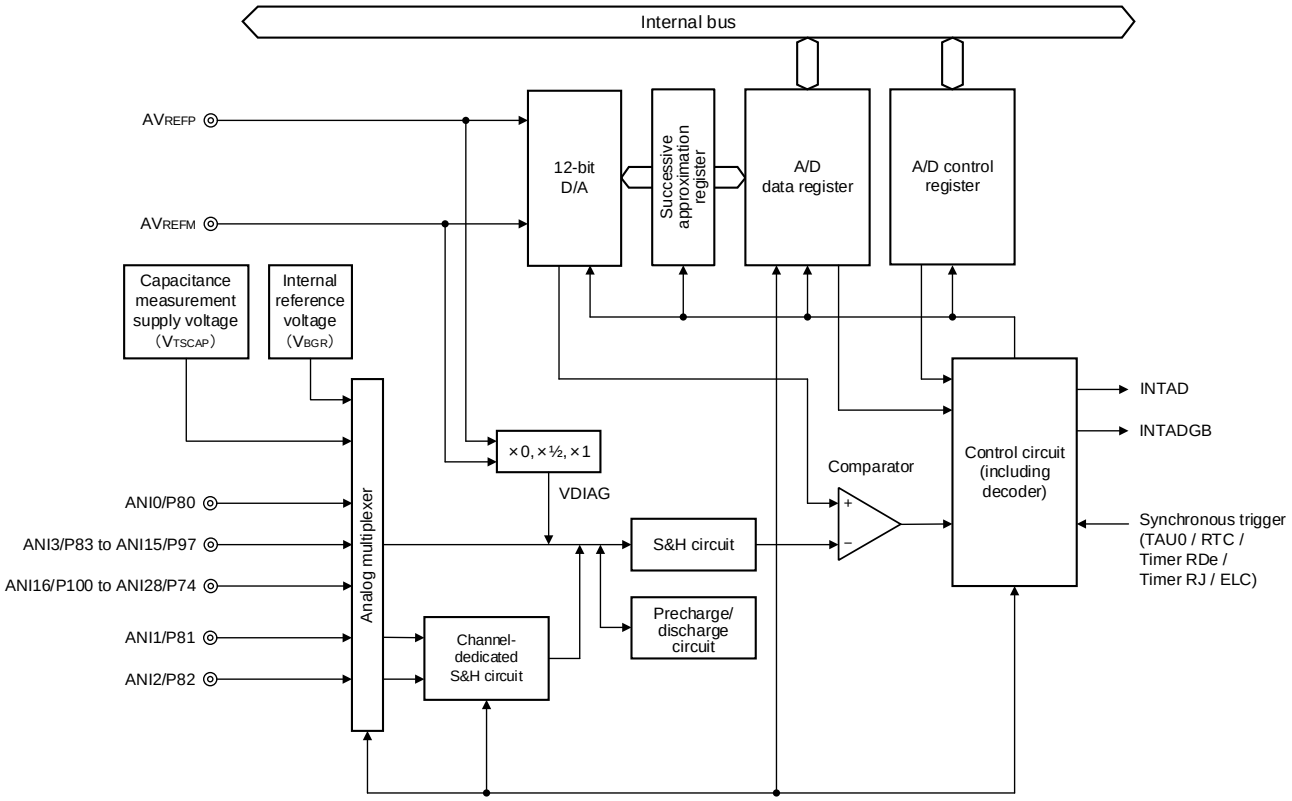
4. For the A/D conversion processing time, refer to **Table 12-21 Times for Conversion during Scanning (in Numbers of Cycles of ADCLK and f_{CLK})**.

5. In this chapter, the description about ELC is applicable for RL78/F25 only.

Table 12-2. Functions of 12-Bit A/D Converter

Item			Pin Name, Function
Analog input channels			ANI0 to ANI28, internal reference voltage (V_{BGR}), Capacitance measurement supply voltage (V_{TSCAP}). ANI1 and ANI2 have a channel-dedicated sample-and-hold circuit.
Conditions for A/D conversion start	Software	Software trigger	Available
	Synchronous trigger	Triggers from TAU0, RTC, Timer RDe, Timer RJ, ELC	Available
Interrupt			INTAD, INTADGB interrupt
Setting of clock supply stop function			PER0.ADCEN bit
A/D conversion clock control			A/D conversion clock (ADCLK) can be selected by the ADCK[1:0] bit in the ADCKS register.

Figure 12-1. Block Diagram of 12-Bit A/D Converter



Remarks S&H: Sample-and-Hold
VDIAG: Self-diagnosis voltage

Caution The V_{TSCAP} level input of this circuit is used as the ANI29 input (stores the A/D conversion result in the ADDR29 register).

Table 12-3 shows the pin function used for the 12-bit A/D converter.

Table 12-3. Pins Used for the 12-bit A/D Converter

Pin Function	I/O	Function	Channel-Dedicated Sample-and-Hold Circuits
AV_{DD}	–	Positive power supply	–
AV_{SS}	–	Ground potential	–
AV_{REFP}	Input	A/D converter reference voltage (+ side) input	–
AV_{REFM}	Input	A/D converter reference voltage (− side) input	–
ANI0, ANI3 to ANI15	Input	Analog input that supports high-speed conversion	–
ANI1, ANI2	Input	Analog input that supports high-speed conversion	○
ANI16 to ANI28	Input	Analog input that supports normal speed conversion	–

12.2 Register Descriptions

Table 12-4 shows the list of registers for the 12-bit A/D converter.

Table 12-4. A/D Converter Register Configuration

Address	Register Name	Symbol	After Reset	Access Size
F02C0H	Peripheral enable register 0	PER0	00H	1, 8
F02C7H	A/D conversion clock control register	ADCKS	00H	8
F06A0H	A/D data register 0 mirror area	ADDR0M	0000H	16
F06A2H	A/D data register 1 mirror area	ADDR1M	0000H	16
F06A4H	A/D data register 2 mirror area	ADDR2M	0000H	16
F06A6H	A/D data register 3 mirror area	ADDR3M	0000H	16
F06A8H	A/D data register 4 mirror area	ADDR4M	0000H	16
F06AAH	A/D data register 5 mirror area	ADDR5M	0000H	16
F06ACH	A/D data register 6 mirror area	ADDR6M	0000H	16
F06AEH	A/D data register 7 mirror area	ADDR7M	0000H	16
F06B0H : F06BFH	A/D converter window registers For details, see **Table 12-5** to **Table 12-15**.	–	–	–
FFF30H	A/D converter access window register	ADWINR	00H	8

Remark See **12.2.18 Port Mode Control Registers (PMC7 to PMC10, PMC12)** and **12.2.19 Port Mode Registers (PM7 to PM10, PM12)** for the port mode control registers and port mode registers.

Table 12-5. A/D Converter Window Register (Page.0 Note)

Address	Register Name	Symbol	After Reset	Access Size
F06B0H	A/D control register	ADCSR	0000H	16
F06B4H	A/D channel select register A0	ADANSA0	0000H	16
F06B6H	A/D channel select register A1	ADANSA1	0000H	16
F06B8H	Addition/average function channel select register 0	ADADS0	0000H	16
F06BAH	Addition/average function channel select register 1	ADADS1	0000H	16
F06BCH	Addition/average counter select register	ADADC	00H	1, 8
F06BEH	A/D control expansion register	ADCER	0000H	16

Note ADWINR.ADPAGE[3:0] bits are the condition of 0000B (page.0 access selection).

Table 12-6. A/D Converter Window Register (Page.1 Note)

Address	Register Name	Symbol	After Reset	Access Size
F06B0H	A/D start trigger select register	ADSTRGR	0000H	16
F06B2H	A/D expansion input control register	ADEXICR	0000H	16
F06B4H	A/D channel select register B0	ADANSB0	0000H	16
F06B6H	A/D channel select register B1	ADANSB1	0000H	16
F06BCH	A/D internal reference voltage data register	ADOCDR	0000H	16
F06BEH	A/D self-test mode data register	ADRD	0000H	16

Note ADWINR.ADPAGE[3:0] bits are the condition of 0001B (page.1 access selection).

Table 12-7. A/D Converter Window Register (Page.2 Note)

Address	Register Name	Symbol	After Reset	Access Size
F06B0H	A/D data register 0	ADDR0	0000H	16
F06B2H	A/D data register 1	ADDR1	0000H	16
F06B4H	A/D data register 2	ADDR2	0000H	16
F06B6H	A/D data register 3	ADDR3	0000H	16
F06B8H	A/D data register 4	ADDR4	0000H	16
F06BAH	A/D data register 5	ADDR5	0000H	16
F06BCH	A/D data register 6	ADDR6	0000H	16
F06BEH	A/D data register 7	ADDR7	0000H	16

Note ADWINR.ADPAGE[3:0] bits are the condition of 0010B (page.2 access selection).

Table 12-8. A/D Converter Window Register (Page.3 Note)

Address	Register Name	Symbol	After Reset	Access Size
F06B0H	A/D data register 8	ADDR8	0000H	16
F06B2H	A/D data register 9	ADDR9	0000H	16
F06B4H	A/D data register 10	ADDR10	0000H	16
F06B6H	A/D data register 11	ADDR11	0000H	16
F06B8H	A/D data register 12	ADDR12	0000H	16
F06BAH	A/D data register 13	ADDR13	0000H	16
F06BCH	A/D data register 14	ADDR14	0000H	16
F06BEH	A/D data register 15	ADDR15	0000H	16

Note ADWINR.ADPAGE[3:0] bits are the condition of 0011B (page.3 access selection).

Table 12-9. A/D Converter Window Register (Page.4 Note)

Address	Register Name	Symbol	After Reset	Access Size
F06B0H	A/D data register 16	ADDR16	0000H	16
F06B2H	A/D data register 17	ADDR17	0000H	16
F06B4H	A/D data register 18	ADDR18	0000H	16
F06B6H	A/D data register 19	ADDR19	0000H	16
F06B8H	A/D data register 20	ADDR20	0000H	16
F06BAH	A/D data register 21	ADDR21	0000H	16
F06BCH	A/D data register 22	ADDR22	0000H	16
F06BEH	A/D data register 23	ADDR23	0000H	16

Note ADWINR.ADPAGE[3:0] bits are the condition of 0100B (page.4 access selection).

Table 12-10. A/D Converter Window Register (Page.5 Note)

Address	Register Name	Symbol	After Reset	Access Size
F06B0H	A/D data register 24	ADDR24	0000H	16
F06B2H	A/D data register 25	ADDR25	0000H	16
F06B4H	A/D data register 26	ADDR26	0000H	16
F06B6H	A/D data register 27	ADDR27	0000H	16
F06B8H	A/D data register 28	ADDR28	0000H	16
F06BAH	A/D data register 29	ADDR29	0000H	16

Note ADWINR.ADPAGE[3:0] bits are the condition of 0101B (page.5 access selection).

Table 12-11. A/D Converter Window Register (Page.6 Note)

Address	Register Name	Symbol	After Reset	Access Size
F06B6H	A/D sample-and-hold circuit control register	ADSHCR	001AH	16

Note ADWINR.ADPAGE[3:0] bits are the condition of 0110B (page.6 access selection).

Table 12-12. A/D Converter Window Register (Page.7 Note)

Address	Register Name	Symbol	After Reset	Access Size
F06BAH	A/D disconnection detection control register	ADDISCR	00H	8

Note ADWINR.ADPAGE[3:0] bits are the condition of 0111B (page.7 access selection).

Table 12-13. A/D Converter Window Register (Page.8 Note)

Address	Register Name	Symbol	After Reset	Access Size
F06B0H	A/D group scan priority control register	ADGSPCR	0000H	16
F06BAH	A/D high-/low-potential reference voltage control register	ADHVREFCNT	91H	1, 8

Note ADWINR.ADPAGE[3:0] bits are the condition of 1000B (page.8 access selection).

Table 12-14. A/D Converter Window Register (Page.13 Note)

Address	Register Name	Symbol	After Reset	Access Size
F06BDH	A/D sampling state register L (ANI16 to ANI29)	ADSSTRL	0DH	8
F06BFH	A/D sampling state register O (V_{BGR})	ADSSTRO	0DH	8

Note ADWINR.ADPAGE[3:0] bits are the condition of 1101B (page.13 access selection).

Table 12-15. A/D Converter Window Register (Page.14 Note)

Address	Register Name	Symbol	After Reset	Access Size
F06B0H	A/D sampling state register 0	ADSSTR0	0DH	8
F06B1H	A/D sampling state register 1	ADSSTR1	0DH	8
F06B2H	A/D sampling state register 2	ADSSTR2	0DH	8
F06B3H	A/D sampling state register 3	ADSSTR3	0DH	8
F06B4H	A/D sampling state register 4	ADSSTR4	0DH	8
F06B5H	A/D sampling state register 5	ADSSTR5	0DH	8
F06B6H	A/D sampling state register 6	ADSSTR6	0DH	8
F06B7H	A/D sampling state register 7	ADSSTR7	0DH	8
F06B8H	A/D sampling state register 8	ADSSTR8	0DH	8
F06B9H	A/D sampling state register 9	ADSSTR9	0DH	8
F06BAH	A/D sampling state register 10	ADSSTR10	0DH	8
F06BBH	A/D sampling state register 11	ADSSTR11	0DH	8
F06BCH	A/D sampling state register 12	ADSSTR12	0DH	8
F06BDH	A/D sampling state register 13	ADSSTR13	0DH	8
F06BEH	A/D sampling state register 14	ADSSTR14	0DH	8
F06BFH	A/D sampling state register 15	ADSSTR15	0DH	8

Note ADWINR.ADPAGE[3:0] bits are the condition of 1110B (page.14 access selection).

12.2.1 Peripheral Enable Register 0 (PER0)

The PER0 register is used to enable or disable supply of the clock signal to peripheral hardware. Clock supply to a hardware that is not in use is stopped in order to reduce power consumption and noise. When the 12-bit A/D converter is to be used, be sure to set bit 5 (ADCEN) to 1.

The PER0 register can be set by a 1-bit or an 8-bit memory manipulation instruction. Reset signal generation clears this register to 00H.

Figure 12-2. Format of Peripheral Enable Register 0 (PER0)

Address: F02C0H 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 over supply of the input clock for the 12-bit A/D converter
0	Stops input clock supply. • SFR used by the A/D converter cannot be written. • The A/D converter is in the reset status.
1	Enables input clock supply. • SFR used by the A/D converter can be read and written.

Caution 1. When setting the 12-bit A/D converter, be sure to set the registers first while the ADCEN bit is set to 1. If ADCEN = 0, the values of the registers which control the 12-bit A/D converter are cleared to their initial values and writing to them is ignored (except for port mode registers 7 to 10, 12 (PM7 to PM10, PM12) and port mode control registers 7 to 10, 12 (PMC7 to PMC10, PMC12).

2. Be sure to clear the bit 6.

12.2.2 A/D Data Registers (ADDRy, ADOCDR, ADDRxM) [x = 0 to 7, y = 0 to 29]

ADDRy (y = 0 to 29) are 16-bit read-only registers which store the A/D conversion results.

The each ADDR0 to ADDR28 registers corresponds to the ANI0 to ANI28 pin input voltage.

The ADDR29 register is a 16-bit read-only register that stores the A/D conversion results of the capacitance measurement supply voltage (V_{TSCAP}).

ADOCDR is a 16-bit read-only register that stores the A/D conversion results of the internal reference voltage (V_{BGR}).

ADDRxM are 16-bit read-only registers. These registers are mirror registers of ADDR0 to ADDR7.

The format of each register differs depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (right aligned or left aligned)
- Settings of the addition count select bits (ADADC.ADC[2:0]) (addition once, twice, three, or 15 times)
- Settings of the average mode enable bit (ADADC.AVEE) (addition or average)

Figure 12-3 shows the format of each A/D data registers.

Figure 12-3. Format of A/D Data Registers (ADDRy, ADOCDR, ADDRxM)

- **A/D Data Registers (ADDRy) [y = 0 to 29]**

Address: ADDR0 to ADDR7: F06B0H to F06BFH (ADWINR = 02H)
ADDR8 to ADDR15: F06B0H to F06BFH (ADWINR = 03H)
ADDR16 to ADDR23: F06B0H to F06BFH (ADWINR = 04H)
ADDR24 to ADDR29: F06B0H to F06BBH (ADWINR = 05H)
After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRy																

- **A/D Internal Reference Voltage Data Register (ADOCDR)**

Address: ADOCDR: F06BCH (ADWINR = 01H) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADOCDR																

- **A/D Data Registers Mirror Area (ADDRxM) [x = 0 to 7]**

Address: ADDR0M to ADDR7M: F06A0H to F06AFH After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRxM																

- Format of A/D data registers
  - ADCER.ADRFMT = 0 (Right aligned)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	AD[11:0]											

  - ADCER.ADRFMT = 1 (Left aligned)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AD[11:0]												0	0	0	0

- Format of A/D data registers for A/D-converted value addition mode (Additional counter = 1 to 4 times)
  - ADCER.ADRFMT = 0 (Right aligned)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	AD[13:0]													

  - ADCER.ADRFMT = 1 (Left aligned)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AD[13:0]														0	0

- Format of A/D data registers for A/D-converted value addition mode (Additional counter = 16 times)
  - ADCER.ADRFMT = 0 or 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AD[15:0]															

Remark See **12.3.7 A/D-converted Value Addition/Average Mode** for A/D-converted value addition mode.

12.2.3 A/D Self-diagnosis Data Register (ADRD)

ADRD is a 16-bit read-only register that stores the A/D conversion results based on the 12-bit A/D converter's self-diagnosis. In addition to the A/D-converted value, the self-diagnosis status is included in. The data format of the ADRD register is set by the A/D data register format select bit (ADCER.ADRFMT) (right aligned or left aligned).

Figure 12-4. Format of A/D Self-diagnosis Data Register (ADRD)

Address: F06BEH (ADWINR = 01H) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADRD																

- ADCER.ADRFMT = 0 (Right aligned)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIAGST[1:0]		0	0	AD[11:0]											

- ADCER.ADRFMT = 1 (Left aligned)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AD[11:0]												0	0	DIAGST[1:0]	

AD[11:0]	A/D conversion result
• Right aligned: The A/D-converted value is stored in bits 11 to 0 from the most significant bit. • Left aligned: The A/D-converted value is stored in bits 15 to 4 from the most significant bit.	

DIAGST[1:0]	Self-diagnosis status
00B	Self-diagnosis has never been executed since power-on.
01B	Self-diagnosis using the 0 V (low-potential reference voltage) has been executed.
10B	Self-diagnosis using the high-potential reference voltage × 1/2 has been executed.
11B	Self-diagnosis using the high-potential reference voltage has been executed.

Caution The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function.

Remarks 1. See **12.3.7 A/D-converted Value Addition/Average Mode** for A/D-converted value addition mode.

2. When the self-diagnosis is complete, the DIAGST[1:0] bit value is updated according to the self-diagnosis voltage (VDIAG).

12.2.4 A/D Control Register (ADCSR)

The ADCSR register sets A/D conversion start trigger, enables/disables scan end interrupt, selects the scan mode, and starts or stops A/D conversion.

This register can be set by a 16-bit memory manipulation instruction.

Figure 12-5. Format of A/D Control Register (ADCSR)

Address: F06B0H (ADWINR = 00H) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCSR	ADST	ADCS[1:0]		ADIE	0	AD HSC	TRGE	0	0	GBA DIE	0	0	0	0	0	0

ADST	Control of A/D conversion start
0	Stops A/D conversion process.
1	Starts A/D conversion process.

Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input.

[Conditions to go 1]

- When 1 was written by software.
- When the ADCSR.TRGE bit set to 1 and the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits was detected.
- In case of group scan mode and when the ADCSR.TRGE bit being set to 1 and the synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits was detected.
- When the group priority operation mode enabled (ADCSR.ADCS[1:0] bits = 01B and ADGSPCR.PGS bit = 1) and the ADGSPCR.GBRP bit is set to 1 and A/D conversion of the lowest-priority group was started.

[Conditions to go 0]

- When 0 was written by software.
- When A/D conversion for all selected channels or the internal reference voltage (V_{BGR}) was completed in single scan mode.
- When Group A scan was completed in group scan mode.
- When Group B scan was completed in group scan mode.
- When the group priority operation mode enabled (ADCSR.ADCS[1:0] bits = 01B and ADGSPCR.PGS bit = 1) and the ADGSPCR.GBRSCN bit is set to 1 and scanning was finished by a trigger of the lowest priority group.

Cautions 1. When the group priority operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01B and ADGSPCR.PGS bit = 1), do not set the ADST bit to 1.

2. When the group priority operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01B and ADGSPCR.PGS bit = 1), do not set the ADST bit to 0. The procedure for forcing A/D conversion to stop is shown in 12.7.3 Procedure for Stopping A/D Conversion.

3. If the single scan continuous function is used (ADGSPCR.GBRP = 1) when the group priority operation mode is enabled (ADCSR.ADCS[1:0] bits = 01B and ADGSPCR.PGS bit = 1), the ADST bit retains 1.

ADCS[1:0]	Scan mode selection
00B	Single scan mode
01B	Group scan mode
10B	Continuous scan mode
11B	Setting prohibited

In single scan mode, A/D conversion is performed for the analog inputs of a maximum of 30 channels selected with the ADANSA0 and ADANSA1 registers by the ascending order of the channel number. And when A/D conversion is completed for all the selected channels, the scan conversion is stopped.

In continuous scan mode, while the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs of a maximum of 30 channels selected with the ADANSA0 and ADANSA1 registers by the ascending order of the channel number. And when A/D conversion is completed for all the selected channels, A/D conversion is repeated from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan operation, A/D conversion is stopped even if scanning is in progress.

In group scan mode, A/D conversion is performed for the analog inputs (group A) of a maximum of 30 channels selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number after scanning is started by the synchronous trigger (ELC and timer function) selected by the TRSA[5:0] bits in ADSTRGR, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. A/D conversion is also performed for the analog inputs (group B) of a maximum of 30 channels selected with the ADANSB0 and ADANSB1 registers in the ascending order of the channel number after scanning is started by the synchronous trigger (ELC and timer function) selected by ADSTRGR.TRSB[5:0] and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. In the group scan mode, different triggers should be selected for group A, group B.

When selecting the internal reference voltage (VBGR), select single scan mode, and deselect all the channels selected with the ADANSA0 and ADANSA1 registers before performing A/D conversion. When A/D conversion of the internal reference voltage (VBGR) is completed, A/D conversion is stopped.
The ADCS[1:0] bits should be set while the ADST bit is 0. They should not be set simultaneously when 1 is written to the ADST bit.

ADIE	Control of scan end interrupt enable
0	Disables INTAD interrupt generation upon scan completion.
1	Enables INTAD interrupt generation upon scan completion.

ADHSC	Control of low current conversion mode
0	High-speed conversion mode
1	Low current conversion mode

ADHSC bit must be set to 1 when used with 1.8 V ≤ V_{DD} < 2.7 V.

To change the ADHSC bit (from 0 to 1 or from 1 to 0), follow the steps below.

<1> Set ADSLP bit in the ADHVREFCNT register to 1 (A/D analog block standby state)
<2> Wait at least 0.2 μs
<3> Set ADHSC bit
<4> Wait at least 4.8 μs
<5> Set ADSLP bit to 0 (A/D analog block normal operation)
<6> Start A/D conversion after 1 μs or more has passed

TRGE	Control of trigger start enable
0	Disables A/D conversion to be started by trigger.
1	Enables A/D conversion to be started by trigger.

Caution When using the group scan mode, set TRGE bit to 1.

GBADIE	Control of group B scan end interrupt enable
0	Disables INTADGB interrupt generation upon group B scan completion.
1	Enables INTADGB interrupt generation upon group B scan completion.

12.2.5 A/D Channel Select Register A (ADANSA0, ADANSA1)

ADANSA0 and ADANSA1 registers select analog input channels for A/D conversion among ANI0 to ANI29.
These registers can be set by a 16-bit memory manipulation instruction.
In group scan mode, these registers select group A channels.

Figure 12-6. Format of A/D Channel Select Register A (ADANSA0, ADANSA1)

Address: F06B6H (ADWINR = 00H) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADANSA1	–	–	ANSA0[29:16]													

Address: F06B4H (ADWINR = 00H) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADANSA0	ANSA0[15:0]															

ANSA0[n]	A/D conversion channel select
0	ANIn is not subjected to conversion.
1	ANIn is subjected to conversion.
The ANSA0[n] bit select analog input channels for A/D conversion among ANI0 to ANI29. The channels to be selected and the number of channels can be arbitrarily set. The ANSA0[0] bit in the ADANSA0 register corresponds to ANI0 and the ANSA0[28] bit in the ADANSA1 register corresponds to ANI28. Also, the ANSA0[29] bit in the ADANSA1 register corresponds to V_{TSCAP} level input.	

Cautions 1. When performing A/D conversion of the internal reference voltage (V_{BGR}), do not select among input channels. (The setting value of these registers should be 0000H.)
2. The ANSA0[n] bits should be set while the ADCSR.ADST bit is 0.
3. Do not select the same channel in groups A and B in the ADANSA0/ADANSA1 and ADANSB0/ ADANSB1 registers.

Remark n: 0 to 29

12.2.6 A/D Channel Select Register B (ADANSB0, ADANSB1)

ADANSB0 and ADANSB1 registers select analog input channels for A/D conversion among ANI0 to ANI29.

These registers can be set by a 16-bit memory manipulation instruction.

In group scan mode, these registers select group B channels. These registers are not used in other scan modes.

Figure 12-7. Format of A/D Channel Select Register B (ADANSB0, ADANSB1)

Address: F06B6H (ADWINR = 01H) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADANSB1	–	–	ANSB0[29:16]													

Address: F06B4H (ADWINR = 01H) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADANSB0	ANSB0[15:0]															

ANSB0[n]	A/D conversion channel select
0	ANIn is not subjected to conversion.
1	ANIn is subjected to conversion.
The ANSB0[n] bit select analog input channels for A/D conversion among ANI0 to ANI29. The channels to be selected and the number of channels can be arbitrarily set. The ANSB0[0] bit in the ADANSB0 register corresponds to ANI0 and the ANSB0[28] bit in the ADANSB1 register corresponds to ANI28. Also, the ANSA0[29] bit in the ADANSA1 register corresponds to V_{TSCAP} level input.	

Cautions 1. When performing A/D conversion of the internal reference voltage (V_{BGR}), do not select among input channels. (The setting value of these registers should be 0000H.)

2. The ANSB0[n] bits should be set while the ADCSR.ADST bit is 0.

3. Do not select the same channel in groups A and B in the ADANSA0/ADANSA1 and ADANSB0/ ADANSB1 registers.

Remark n: 0 to 29

12.2.7 A/D-converted Value Addition/Average Function Channel Select Register (ADADS0, ADADS1)

ADADS0 and ADADS1 registers select the channels to ANI0 to ANI29 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

These registers can be set by a 16-bit memory manipulation instruction.

Figure 12-8. Format of A/D-converted Value Addition/Average Function Channel Select Register (ADADS0, ADADS1)

Address: F06BAH (ADWINR = 00H) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADADS1	–	–	ADS0[29:16]													

Address: F06B8H (ADWINR = 00H) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADADS0	ADS0[15:0]															

ADS0[n]	A/D-converted value addition/average channel select
0	A/D-converted value addition/average mode for ANIn is not selected.
1	A/D-converted value addition/average mode for ANIn is selected.

When the ADS0[n] bit of the number that is the same as that of A/D-converted channel selected by the ANSA0[n] bits in ADANSA0/ADANSA1 or the ANSB0[n] bits in ADANSB0/ADANSB1 is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits.

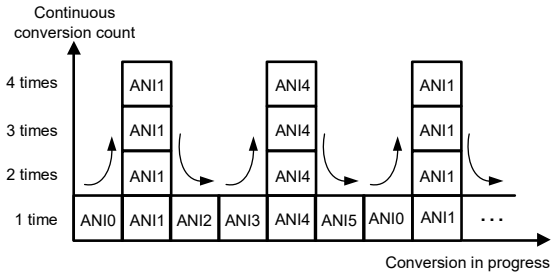
When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register.

Caution The setting of ADS0[1] and ADS0[2] should be 0 when the channel-dedicated sample-and-hold circuits is enable for ANI1 and ANI2.

Remark n: 0 to 29

Figure 12-9 shows a scanning operation sequence in which both the ADS0[1] and ADS0[4] bits are set to 1. It is assumed that addition mode is selected (ADADC.AVEE = 0), the addition count is set to three times (ADADC.ADC[2:0] = 011B), and the channels ANI0 to ANI5 are selected (ADANSA0.ANSA0[15:0] = 003FH) in continuous scan mode (ADCSR.ADCS[1:0] = 10B). The channel-dedicated sample-and-hold circuits of ANI1 is set to disabled (ADSHCR.SHANS[1] = 0). The conversion process begins with ANI0. The ANI1 conversion is performed successively four times (addition three times), and the added (integrated) value is returned to A/D data register 1. After that the ANI2 conversion is started. The ANI4 conversion is performed successively 4 times and the added (integrated) value is returned to A/D data register 4. After conversion of ANI5, the conversion operation is repeatedly performed in the same sequence from ANI0.

Figure 12-9. Scan Conversion Sequence
with ADADC.ADC[2:0] = 011B, ADADC.AVEE = 0, ADS0[1] = 1, ADS0[4] = 1, and ADSHCR.SHANS[1] = 0



Cautions 1. Do not enable this function if channel-dedicated sample-and-hold circuit on ANI1 and ANI2 is enabled.
2. The ANSA0[n] bits should be set while the ADCSR.ADST bit is 0.

Remark n: 0 to 29

12.2.8 A/D-converted Value Addition/Average Count Select Register (ADADC)

The ADADC register sets the addition count for A/D conversion of the channel and selects either addition or average mode.

This register can be set by a 1-bit or an 8-bit memory manipulation instruction.

Figure 12-10. Format of A/D-converted Value Addition/Average Count Select Register (ADADC)

Address: F06BCH (ADWINR = 00H) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
ADADC	AVEE	0	0	0	0	ADC[2:0]		

AVEE	Control of average mode enable
0	Addition mode is selected.
1	Average mode is selected.
The AVEE bit selects addition or average mode for A/D conversion of the channel for which A/D-converted value addition/ average mode is selected. The average value of 1-time, 3-time, and 16-time conversion cannot be obtained. The AVEE bit should be set while the ADCSR.ADST bit is 0.	

ADC2	ADC1	ADC0	Setting of addition count selection
0	0	0	1-time conversion (no addition; same as normal conversion) Note
0	0	1	2-time conversion (addition once)
0	1	0	3-time conversion (addition twice) Note
0	1	1	4-time conversion (addition three times)
1	0	1	16-time conversion (addition 15 times) Note
Other than above			Setting prohibited
The ADC[2:0] bits set the addition count to the channels for which A/D-converted value addition/average mode is selected, and to A/D conversion. The ADC[2:0] bits should be set while the ADCSR.ADST bit is 0.			

Note Only 2-time or 4-time conversion is selectable when the AVEE bit is 1 (average mode). When average mode is selected, do not set 1-time conversion (ADADC.ADC[2:0] bits = 000B), 3-time conversion (ADADC.ADC[2:0] bits = 010B), or 16-time conversion (ADADC.ADC[2:0] bits = 101B).

12.2.9 A/D Control Extended Register (ADCER)

The ADCER register sets self-diagnosis mode, format of A/D data registers (ADDRy, ADDRxM), and automatic clearing of A/D data registers.

This register can be set by a 16-bit memory manipulation instruction.

Figure 12-11. Format of A/D Control Extended Register (ADCER)

Address: F06BEH (ADWINR = 00H) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCER	ADR FMT	0	0	0	DIAG M	DIAG LD	DIAG VAL[1:0]		0	0	ACE	0	0	0	0	0

ADRFMT	A/D data register format select
0	Right aligned is selected for the A/D data register format.
1	Left aligned is selected for the A/D data register format.
The ADRFMT bit specifies right aligned or left aligned for the data to be stored in the ADDRy, ADDRxM, ADRD, or ADOCDR register. The ADRFMT bit should be set while the ADST bit is 0.	

DIAGM	Setting of self-diagnosis enable
0	Disables self-diagnosis of 12-bit A/D converter.
1	Enabled self-diagnosis of 12-bit A/D converter.
Self-diagnosis is used to detect a failure of the 12-bit A/D converter. Specifically, the self-diagnosis voltage (VDIAG) selected by the DIAGVAL[1:0] bits is converted. When conversion is completed, information on the converted voltage and the conversion result is stored into the self-diagnosis data register (ADRD). ADRD register can then be read out by software to determine whether the conversion result falls within the normal range (normal) or not (abnormal). Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. The DIAGM bit should be set while the ADST bit is 0.	

DIAGLD	Setting of self-diagnosis mode select
0	Setting prohibited when self-diagnosis is enabled
1	Fixed mode for self-diagnosis voltage
When self-diagnosis voltage fixed mode is selected, the fixed voltage specified by the DIAGVAL[1:0] bits is converted. The DIAGLD bit should be set while the ADST bit is 0.	

DIAGVAL[1:0]	Setting of self-diagnosis conversion voltage
00B	Setting prohibited in self-diagnosis voltage fixed mode.
01B	Uses the voltage of 0 V (low-potential reference voltage) for self-diagnosis.
10B	Uses the voltage of high-potential reference voltage × 1/2 for self-diagnosis.
11B	Uses the voltage of high-potential reference voltage for self-diagnosis.
For details, refer to the descriptions of the DIAGLD bit. Self-diagnosis should not be executed by setting the DIAGLD bit to 1 when the DIAGVAL[1:0] bits are set to 00B.	

ACE	Setting of A/D data register automatic clearing enable
0	Disables automatic clearing.
1	Enables automatic clearing.
The ACE bit enables or disables automatic clearing (all 0) of ADDRy, ADDRxM, ADRD, or ADOCDR after any of these registers have been read by the CPU or DTC. Automatic clearing of the A/D data register can detect a failure which has not been updated in the A/D data register.	

12.2.10 A/D Conversion Start Trigger Select Register (ADSTRGR)

The ADSTRGR register selects the A/D conversion start trigger.

This register can be set by a 16-bit memory manipulation instruction.

Figure 12-12. Format of A/D Conversion Start Trigger Select Register (ADSTRGR)

Address: F06B0H (ADWINR = 01H) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADSTRGR	0	0	TRSA[5:0]						0	0	TRSB[5:0]					

TRSA[5:0], TRSB[5:0]	A/D conversion start synchronous trigger select Note 1
000001B	INTTM01 (TAU0 ch1 interrupt)
000010B	INTRTC (RTC interrupt)
000011B	INTTRJ0 (Timer RJ0 interrupt)
000100B	INTTRD0_IFA (Timer RDe0 interrupt)
000101B	INTTRD0_IFB (Timer RDe0 interrupt)
000110B	INTTRD1_IFA (Timer RDe1 interrupt)
000111B	INTTRD1_IFB (Timer RDe1 interrupt)
001000B	INTTRD1_UDF (Timer RDe1 under flow)
001011B	INTTRD_ADTRG (Timer RDe A/D trigger request)
110000B	Event output signal from event link controller (ELCTRG0)
111111B	Disables synchronous trigger
Other than above	Setting prohibited

TRSA[5:0] Notes 2 ,3

The TRSA[5:0] bits select the synchronous trigger to start A/D conversion in single scan mode and continuous scan mode, and this setting is invalid when ADCSR.TRGE bit is 0. In group scan mode, the trigger to start scanning of the analog input selected in group A is selected. When scanning is executed in group scan mode, can't use a software trigger.

TRSB[5:0]

The TRSB[5:0] bits select the synchronous trigger to start scanning of the analog input selected in group B. These bits require to be set only in group scan mode and are not used in any other scan mode.

A setting of software trigger is prohibited for the scan conversion start trigger of group B. Therefore, TRSB[5:0] bits should be set to the value other than 000000B and the ADCSR.TRGE bit should be set to 1 in group scan mode.

The setting of 110000B (ELCTRG0) is prohibited for RL78/F22 because of no ELC function.

Notes 1. The issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by a trigger may have no effect. See **12.3.5 Analog Input Sampling Time and Scan Conversion Time** for details.

2. When using the A/D conversion startup source of a synchronous trigger, set the ADCSR.TRGE bit to 1.

3. Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit and the TRSA[5:0] bits.

12.2.11 A/D Conversion Extended Input Control Register (ADEXICR)

The ADEXICR register specifies the settings of A/D conversion of the internal reference voltage (V_{BGR}).
This register can be set by a 16-bit memory manipulation instruction.

Figure 12-13. Format of A/D Conversion Extended Input Control Register (ADEXICR)

Address: F06B2H (ADWINR = 01H) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADEXICR	0	0	0	0	0	0	OCSA	0	0	0	0	0	0	0	0	0

OCSA	Internal reference voltage (V_{BGR}) conversion select
0	A/D conversion of internal reference voltage (V_{BGR}) is not performed.
1	A/D conversion of internal reference voltage (V_{BGR}) is performed.

This bit selects A/D conversion of the internal reference voltage (V_{BGR}) in single scan mode. When A/D conversion of the internal reference voltage (V_{BGR}) is to be performed, all the bits in the ADANSA0, ADANSA1, ADANSB0, and ADANSB1 registers should be set to all 0 in single scan mode.

The OCSA bit should be set while the ADCSR.ADST bit is 0. For A/D conversion of the internal reference voltage (V_{BGR}), discharge the A/D converter before sampling. The sampling time should be 5 μs or longer.
Sampling starts after discharging is completed during A/D conversion of the internal reference voltage (V_{BGR}), so an auto-discharging period of 15 ADCLK cycles is inserted before sampling.

12.2.12 A/D Sampling State Register n (ADSSTRn) [n = 0 to 15, L, O]

The ADSSTRn register sets the sampling time for analog input.

1 state means 1 ADCLK (A/D conversion clock) cycle. So if the ADCLK clock is 40 MHz, one state is 25 ns.

The initial value is 13.5 states. The sampling time can be adjustable by this register according to the impedance of analog input signal source. The ADSSTRn register should be set while the ADCSR.ADST bit is 0. Set a value between 05H (5 states) and FFH (255 states) in this register.

This register can be set by an 8-bit memory manipulation instruction.

Figure 12-14. Format of A/D Sampling State Register n (ADSSTRn) (n = 0 to 15, L, O)

Address: ADSSTRL: F06BDH, ADSSTRO: F06BFH, ADSSTR0: F06B0H, ADSSTR1: F06B1H, ADSSTR2: F06B2H,
ADSSTR3: F06B3H, ADSSTR4: F06B4H, ADSSTR5: F06B5H, ADSSTR6: F06B6H, ADSSTR7: F06B7H,
ADSSTR8: F06B8H, ADSSTR9: F06B9H, ADSSTR10: F06BAH, ADSSTR11: F06BBH, ADSSTR12: F06BCH,
ADSSTR13: F06BDH, ADSSTR14: F06BEH, ADSSTR15: F06BFH
(ADSSTRL: ADWINR = 0DH, ADSSTRO: ADWINR = 0DH, ADSSTR0 to ADSSTR15: ADWINR = 0EH)

After reset: 0DH R/W

Symbol	7	6	5	4	3	2	1	0
ADSSTRn								

Table 12-16 shows the relationship between the A/D sampling state register and the relevant channels. For details, refer to **12.3.5 Analog Input Sampling Time and Scan Conversion Time**.

Table 12-16. Relationship between A/D Sampling State Register and Relevant Channels

Register Name	Channels
ADSSTR0	ANI0 Note 1
ADSSTR1	ANI1 Note 1
ADSSTR2	ANI2 Note 1
ADSSTR3	ANI3 Note 1
ADSSTR4	ANI4 Note 1
ADSSTR5	ANI5 Note 1
ADSSTR6	ANI6 Note 1
ADSSTR7	ANI7 Note 1
ADSSTR8	ANI8 Note 1
ADSSTR9	ANI9 Note 1
ADSSTR10	ANI10 Note 1
ADSSTR11	ANI11 Note 1
ADSSTR12	ANI12 Note 1
ADSSTR13	ANI13 Note 1
ADSSTR14	ANI14 Note 1
ADSSTR15	ANI15 Note 1
ADSSTRL	ANI16 to ANI29 Note 2
ADSSTRO	Internal reference voltage (V_{BGR}) Note 3

Notes 1. To perform the A/D conversion of the high-speed analog input channel, the sampling time must be set to 0.337 µs or longer. When using low current conversion mode, the sampling time must be 1.3125 µs or longer.

2. To perform the A/D conversion of the analog input channel, the sampling time must be set to 1.012 µs or longer. When using low current conversion mode, the sampling time must be 2.5625 µs or longer.

3. When performing A/D conversion of the internal reference voltage (V_{BGR}), the sampling time should be 5 µs or longer.

12.2.13 A/D Sample-and-Hold Circuit Control Register (ADSHCR)

The ADSHCR register is used to control the channel-dedicated sample-and-hold circuits.

This register can be set by a 16-bit memory manipulation instruction.

Figure 12-15. Format of A/D Sample-and-Hold Circuit Control Register (ADSHCR)

Address: F06B6H (ADWINR = 06H) After reset: 001AH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADSHCR	0	0	0	0	0	SHANS[2:1]		0	SSTSH[7:0]							

SHANS[2]	Control of ANI2 channel-dedicated sample-and-hold circuit bypass
0	Bypass the channel-dedicated sample-and-hold circuit.
1	Use the channel-dedicated sample-and-hold circuit.
This bit should be set while the ADCSR.ADST bit is 0. In the group priority operation, make the setting of bypassing the channel-dedicated sample-and-hold circuit when ANI2 is selected for group B.	

SHANS[1]	Control of ANI1 channel-dedicated sample-and-hold circuit bypass
0	Bypass the channel-dedicated sample-and-hold circuit.
1	Use the channel-dedicated sample-and-hold circuit.
This bit should be set while the ADCSR.ADST bit is 0. In the group priority operation, make the setting of bypassing the channel-dedicated sample-and-hold circuit when ANI1 is selected for group B.	

SSTSH[7:0]	Setting of channel-dedicated sample-and-hold circuit sampling time
04H to FFH	Set the sampling time (4 to 255 states).
Other than above	Setting prohibited.
If the impedance of analog input signal source is too high to secure sufficient sampling time or the ADCLK clock is slow, the sampling time can be adjusted. The SSTSH[7:0] bits should be set while the ADCSR.ADST bit is 0. The sampling time must be set to a value that is 4 states or more and is 255 states or less. Also, the sampling time needs to be 0.4 μs or more.	

Caution When using the low current conversion mode, set SHANS[2:1] bits to 00B (channel-dedicated sample-and-hold circuit is not used).

12.2.14 A/D Disconnection Detection Control Register (ADDISCR)

The ADDISCR register is used to control the disconnection detection assist function.

This register can be set by an 8-bit memory manipulation instruction.

Figure 12-16. Format of A/D Disconnection Detection Control Register (ADDISCR)

Address: F06BAH (ADWINR = 07H) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADDISCR	0	0	0	ADNDIS[4:0]				

ADNDIS4	Setting of precharge/discharge selection Note 1
0	Discharge select
1	Precharge select
The ADNDIS4 bit selects either precharge or discharge for the A/D disconnection detection assist function. The ADNDIS4 bit should be set while the ADCSR.ADST bit is 0.	

ADNDIS[3:0]	Period of precharge/discharge Note 2
0000B	Disables the disconnection detection assist function.
0001B	Setting prohibited.
0010B	2 cycles (2/ADCLK)
0011B	3 cycles (3/ADCLK)
0100B	4 cycles (4/ADCLK)
0101B	5 cycles (5/ADCLK)
0110B	6 cycles (6/ADCLK)
0111B	7 cycles (7/ADCLK)
1000B	8 cycles (8/ADCLK)
1001B	9 cycles (9/ADCLK)
1010B	10 cycles (10/ADCLK)
1011B	11 cycles (11/ADCLK)
1100B	12 cycles (12/ADCLK)
1101B	13 cycles (13/ADCLK)
1110B	14 cycles (14/ADCLK)
1111B	15 cycles (15/ADCLK)
When the self-diagnosis is used, the disconnection detection assistance cannot be used. In that case, the ADNDIS[3:0] bits should be set to 0000B. When the ADNDIS[3:0] bits are set to any values other than 0000B or 0001B, and the disconnection detection assistance is enabled, the disconnection detection assistance for the channel-dedicated sample-and-hold circuit are also enabled. ADNDIS[3:0] bits should be set while the ADCSR.ADST bit is 0.	

Notes 1. When the internal reference voltage is converted, A/D converter executes discharge automatically.

2. This operation is achieved by setting ADNDIS[4:0] to 0FH (15 cycles) automatically in setting ADEXICR.OCSA to 1. After executing discharge, the sampling will start.

12.2.15 A/D Group Scan Priority Control Register (ADGSPCR)

The ADGSPCR register is used to control the group scan priority which suspend the scan of lower priority group in group scan mode and perform the higher priority group scan.

This register can be set by a 16-bit memory manipulation instruction.

Figure 12-17. Format of A/D Group Scan Priority Control Register (ADGSPCR)

Address: F06B0H (ADWINR = 08H) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADEXICR	GBRP	LGRRS	0	0	0	0	0	0	0	0	0	0	0	0	GBRSCN	PGS

GBRP	Control of single scan continuous start
0	Single scan is not continuously activated.
1	Single scan for the group with the lower priority is continuously activated.

When the GBRP bit has been set to 1, single scan is performed continuously for the lowest-priority group regardless of the setting of the GBRSCN bit. It's enabled only when PGS bit is 1 and reserved when PGS bit is 0.

Setting the GBRP bit to 1 starts a single scan on the group of the lower priority: Group B. On completion of the scan, another single scan on the group B is automatically started. If the scan has been suspended due to the group priority operation, single scan on the group B is automatically restarted on completion of the A/D conversion of the group A.

Disable the trigger input of the group B before setting the GBRP bit to 1. When the GBRP bit is set to 1 re-scan is done only on the group B even if the GBRSCN bit is 0.

The GBRP bit should be set while the ADCSR.ADST bit is 0.

LGRRS	Setting of restart channel selection
0	Start rescanning from the first channel for scanning.
1	Start rescanning from the channel for which A/D conversion is not completed.

This bit specifies the starting channel of re-scan in group priority operation. The setting of the LGRRS bit is effective when the PGS bit and the GBRSCN bit are 1.

When the LGRRS bit is 0, the scan of the low-priority group that has been suspended by the group priority operation restarts from the first channel, when the scan of the priority group finishes.

When the LGRRS bit is 1, the scan of the low-priority group that has been suspended by the group priority operation restarts from a channel for which A/D conversion has not been done, when the scan of the priority group finishes [Note].

The ADCSR.ADST bit must be 0 when the LGRRS bit is to be set.

Note If A/D conversion had not been done on an addition setting channel for the specified number of times, A/D conversion is done for that number of times on the addition setting channel when the scan restarts.

GBRSCN	Setting of lower-priority group restart
0	Disable rescanning of the group that was stopped in group priority operation.
1	Enable rescanning of the group that was stopped in group priority operation.

The GBRSCN bit controls the restart of scan for group priority operation. It's enabled only when PGS bit is 1, and reserved when PGS bit is 0.

When the GBRSCN bit is set to 1, the scan of a low-priority group restarts after the completion of the scan of the priority group that has been started by a trigger that has suspended the scan of the low-priority group. If there has been a trigger of a low-priority group during the scan of a priority group, the scan of the low-priority group starts after the completion of the scan of the priority group.

If the GBRSCN bit has been set to 0, triggers that are input during the scan are ignored. Also, the ADCSR.ADST bit must be 0 when the GBRSCN bit is to be set.

PGS	Setting of group priority operation
0	Operate without group priority control
1	Operate with group priority control

This bit controls the priority operation in the group scan mode. Set this bit to 1 to enable the group priority operation. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01B (group scan mode). If the bits are set to any other values, proper operation is not guaranteed.

The group priority operation accepts the start of scan for a priority group during the scan of a low-priority group, suspends the scan of the low-priority group, and starts the scan of the priority group.

The priority order is group A > group B. when this operation accepts the start of scan for group A during the scan of group B, it suspends the scan of the group B and starts the scan of group A.

When the PGS bit has been set to 0, clear operation must be performed by software according to section **12.7.3 Procedure for Stopping A/D Conversion**.

When the PGS bit has been set to 1, make settings according to section **12.3.4.2 Group Priority Operation (when ADGSPCR.PGS = 1)**.

12.2.16 A/D High-/Low-potential Reference Voltage Control Register (ADHVREFCNT)

The ADHVREFCNT register specifies the high-potential and low-potential reference voltages. Set this register before performing A/D conversion.

This register can be set by a 1-bit or an 8-bit memory manipulation instruction.

Figure 12-18. Format of A/D High-/Low-potential Reference Voltage Control Register (ADHVREFCNT)

Address: F06BAH (ADWINR=08H) After reset: 91H R/W

Symbol	<7>	6	5	4	3	2	1	0
ADHVREFCNT	ADSLP	0	0	LVSEL	0	0	0	HVSEL

ADSLP	Selects A/D analog block standby state
0	Normal operation
1	A/D analog block standby state
This bit is used to transition the A/D analog block to the standby state. The ADSLP bit should be set after the ADST bit has been set to 0. After the ADSLP bit is set to 1, wait at least 5 μs before clearing this bit to 0. Furthermore, after the ADSLP bit is cleared to 0, wait at least 1 μs and then start the A/D conversion. However in STOP mode, A/D analog block is standby state regardless of this bit setting.	

LVSEL	Selects low-potential reference voltage
0	Setting is prohibited.
1	AV_{REFM} is selected.
This bit is used to set the low-potential reference voltage.	

HVSEL	Selects high-potential reference voltage
0	Setting is prohibited.
1	AV_{REFP} is selected.
This bit is used to set the high-potential reference voltage.	

Caution The ADSLP bit becomes 1 (A/D analog block standby mode) when reset is released. Refer to 12.7.1 for A/D converter initial settings. In STOP mode, the analog block is in standby state. When STOP mode is released, refer to 12.7.9 Notes on Canceling STOP Mode.

12.2.17 A/D Conversion Clock Control Register (ADCKS)

The ADCKS register is the register that sets the division ratio of the peripheral hardware clock frequency (f_{CLK}), which is the clock source of ADCLK. Set this register before starting A/D conversion.

This register can be set by an 8-bit memory manipulation instruction.

Writing to the ADCKS register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Figure 12-19. Format of A/D Conversion Clock Control Register (ADCKS)

Address: F02C7H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADCKS	0	0	0	0	0	0	ADCK[1:0]	

ADCK[1:0]	Selects A/D conversion clock
00B	Non divided (f_{CLK})
01B	Divided by 2 (f_{CLK}/2)
10B	Divided by 4 (f_{CLK}/4)
11B	Divided by 8 (f_{CLK}/8)

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Caution Settings such that the frequency of the A/D conversion clock (ADCLK) is less than 2 MHz are prohibited. When using the low-current conversion mode, set the A/D clock frequency in the range of 2 MHz to 8 MHz.

12.2.18 Port Mode Control Registers (PMC7 to PMC10, PMC12)

These registers are used to switch the ANI0 to ANI28 pins between the analog input of the A/D converter and the digital I/O of the port.

The PMC7 to PMC10 and PMC12 registers can be set by a 1-bit or an 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 12-20. Formats of Port Mode Control Registers (100-pin products)

Address: PMC7: F0067H PMC8: F0068H PMC9: F0069H PMC10: F006AH PMC12: F006CH

After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PMC7	1	1	1	PMC74	PMC73	PMC72	PMC71	PMC70

Symbol	7	6	5	4	3	2	1	0
PMC8	PMC87	PMC86	PMC85	PMC84	PMC83	PMC82	PMC81	PMC80

Symbol	7	6	5	4	3	2	1	0
PMC9	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

Symbol	7	6	5	4	3	2	1	0
PMC10	1	1	PMC105	PMC104	PMC103	PMC102	PMC101	PMC100

Symbol	7	6	5	4	3	2	1	0
PMC12	1	1	PMC125	1	1	1	1	PMC120

PMCmn	Pmn pin digital I/O or analog input selection (m = 7 to 10, 12, n = 0 to 7)
0	Digital I/O (alternate function other than analog input)
1	Analog I/O

Cautions 1. Set port pins specified as analog input pins to input mode by using port mode register x (PMx).

2. Be sure to set bits for pins that are not present to their initial values.

12.2.19 Port Mode Registers (PM7 to PM10, PM12)

When using the ANI0/P80 to ANI21/P105 and ANI22/P125 to ANI28/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 an 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 12-21. Formats of Port Mode Registers (100-pin products)

Address: PM7: FFF27H PM8: FFF28H PM9: FFF29H PM10: FFF2AH PM12: FFF2CH

After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

Symbol	7	6	5	4	3	2	1	0
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80

Symbol	7	6	5	4	3	2	1	0
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

Symbol	7	6	5	4	3	2	1	0
PM10	PM107	PM106	PM105	PM104	PM103	PM102	PM101	PM100

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
0	Output mode (output buffer on)
1	Input mode (output buffer off)

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 = 7 to 10, 12, n = 0 to 7

The ANI0 to ANI15 pins are as shown below depending on port mode control registers 8 and 9 (PMC8, PMC9), A/D channel select register A0/B0 (ADANSA0, ADANSB0), PM8, and PM9 registers.

Table 12-17. Setting Functions of ANI0 to ANI15 Pins

PMC8 and PMC9	PM8 and PM9	ADANSA0, ADANSB0	ANI0 to ANI15 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 ANI16 to ANI28 pins are as shown below depending on the settings of port mode control registers 7, 10 and 12 (PMC7, PMC10, PMC12), A/D channel select register A1/B1 (ADANSA1, ADANSB1), PM7, PM10 and PM12 registers.

Table 12-18. Setting Functions of ANI16 to ANI28 Pins

PMC7, PMC10, and PMC12	PM7, PM10, and PM12	ADANSA1, ADANSB1	ANI16 to ANI28 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.	

12.2.20 A/D Converter Access Window Register (ADWINR)

The ADWINR register selects a window for the switching of registers that are allocated to addresses from F06B0H to F06BFH.

This register can be set by an 8-bit memory manipulation instruction.

Figure 12-22. Format of A/D Converter Access Window Register (ADWINR)

Address: FFF30H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADWINR	0	0	0	0	ADPAGE[3:0]			

ADPAGE[3:0]	A/D converter access window select bit
0000B	Read or write access is enabled for the registers of ADCSR, ADANSA0, ADANSA1, ADADS0, ADADS1, ADADC and ADCER.
0001B	Read or write access is enabled for the registers of ADSTRGR, ADEXICR, ADANSB0, ADANSB1, ADOCDR and ADRD.
0010B	Read access is enabled for the registers of ADDRy (y:0 to 7).
0011B	Read access is enabled for the registers of ADDRy (y:8 to 15).
0100B	Read access is enabled for the registers of ADDRy (y:16 to 23).
0101B	Read access is enabled for the registers of ADDRy (y:24 to 29).
0110B	Read or write access is enabled for the register of ADSHCR.
0111B	Read or write access is enabled for the register of ADDISCR.
1000B	Read or write access is enabled for the registers of ADGSPCR and ADHVREFCNT.
1101B	Read or write access is enabled for the registers of ADSSTRL and ADSSTRO.
1110B	Read or write access is enabled for the registers of ADSSTRn (n:0 to 15).
Other than above	Setting prohibited.
The ADPAGE bits are used to select the A/D converter access window, and enable the register to be read or written. Set this register before accessing each A/D related register.	

12.3 Operation

12.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

A scan conversion is performed in three operating modes: single scan mode, continuous scan mode and group scan mode.

In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until the ADCSR.ADST bit is cleared to 0 from 1 by software. In group scan mode, the scan of group A and group B is started by the synchronous trigger selected for each group, and the scanning ends when all channels selected for each group are scanned once.

In single scan mode and continuous scan mode, A/D conversion is performed for ANIn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for ANIn channels of group A selected by the ADANSA0 and ADANSA1 registers, for ANIn channels of group B selected by the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.

When self-diagnosis is selected, it is executed once at the beginning of each scan and one of the three voltages internally generated in the 12-bit A/D converter is converted.

When any of ANI1 to ANI2 channels is set as a channel-dedicated sample-and-hold circuit by the ADSHCR.SHANS[2:1] bits, the target analog input is sampled and held before the first A/D conversion of each scan.

For A/D conversion of the internal reference voltage (V_{BGR}), perform the conversion by using the single scan mode and without selecting the other channels.

Caution While the ADCSR.ADST bit is 1 (scanning is in progress), the software trigger and synchronization trigger input that are the start conditions for A/D conversion are ineffective regardless of the scan mode excluding group priority operation.

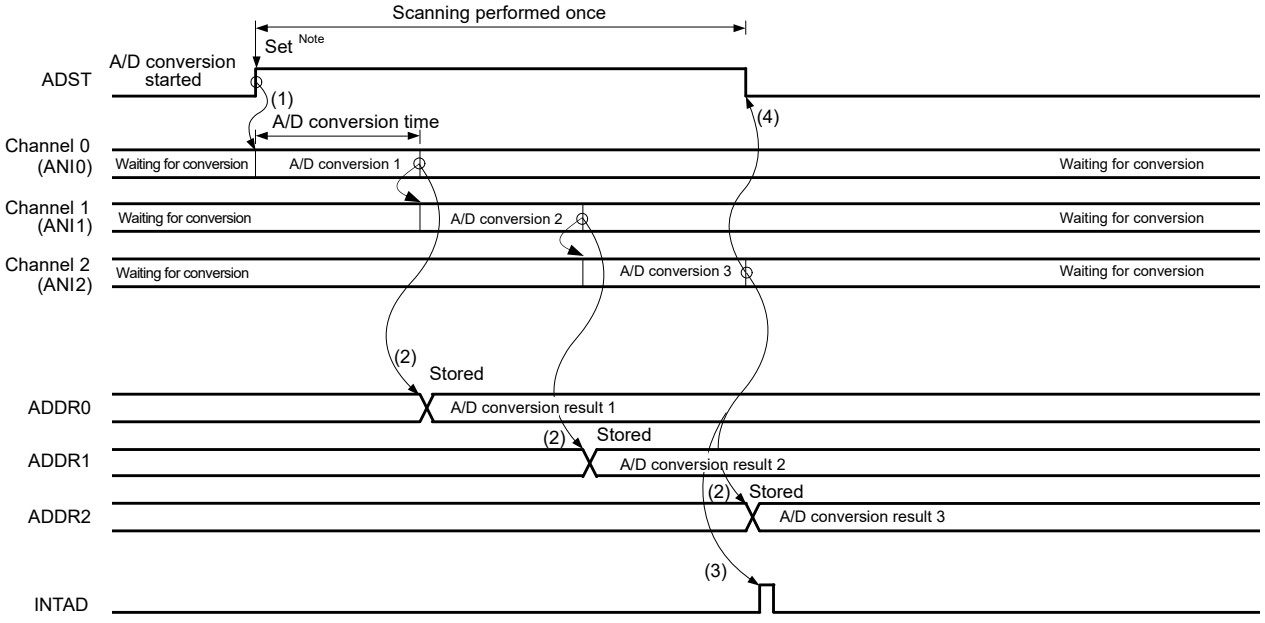
12.3.2 Single Scan Mode

12.3.2.1 Basic Operation (without Channel-Dedicated Sample-and-Hold Circuits)

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input, A/D conversion is performed for ANIn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy, ADDRxM).
(3) When A/D conversion of all the selected channels is completed, an INTAD interrupt request is generated if the ADCSR.ADIE bit is 1 (INTAD interrupt upon scanning completion enabled).
(4) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

Figure 12-23. Example of Operation in Single Scan Mode (Basic Operation: ANI0, ANI1, and ANI2 Selected and without channel-dedicated sample-and-hold circuits)



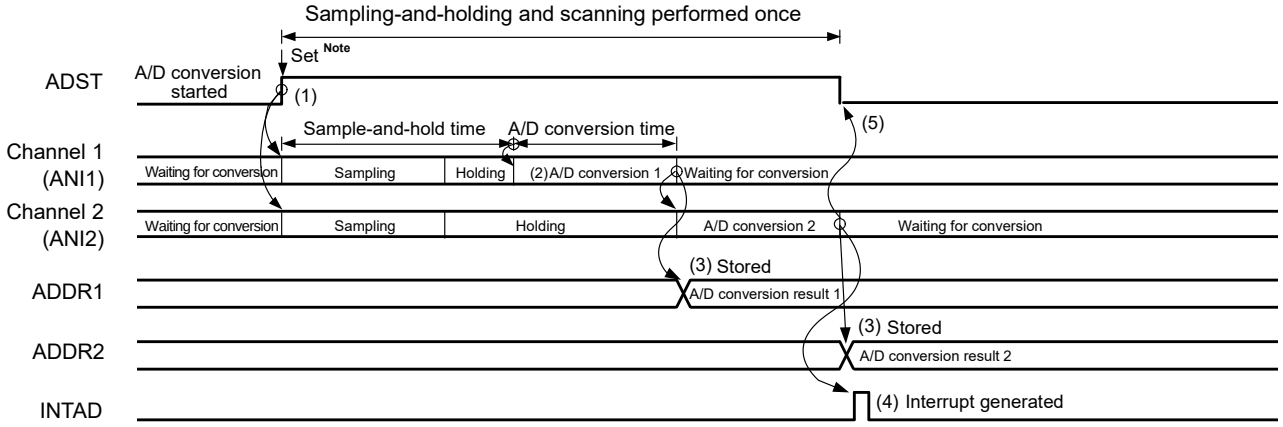
Note ↓indicates the instruction is executed by software.

12.3.2.2 Basic Operation (with Channel-Dedicated Sample-and-Hold circuits)

When a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and this is followed by A/D conversion once of the analog inputs on all selected channels. The ADSHCR.SHANS[2:1] bits are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input.
(2) After sample-and-hold operation, A/D conversion is performed for ANIn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy, ADDRxM).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(5) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a waiting state.

Figure 12-24. Example of Operation in Single Scan Mode
(Channel-Dedicated Sample-and-Hold Circuits Used, ANI1 and ANI2 Selected)



Note ↓ indicates the instruction is executed by software.

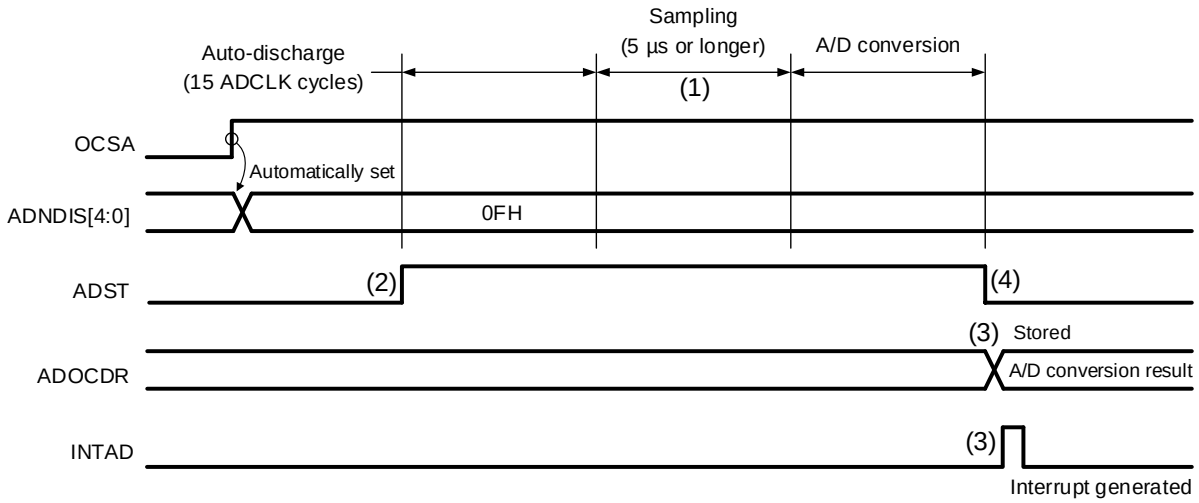
Cautions 1. The maximum total of holding time for the channel dedicated sample-and-hold circuit and sampling time in A/D converter (t_{SPL}) is 10 μs. Set the sampling time of the ADCLK (A/D clock) and the target channel so that the hold time is within 10 μs.
2. In low current conversion mode, do not use the channel-dedicated sample-and-hold circuit.

12.3.2.3 A/D Conversion of Internal Reference Voltage (V_{BGR})

When internal reference voltage is selected, single scan mode should be used.
All analog input channels should be set to disable (All bits of ADANSA0 and ADANSA1 register are set to 0).
A/D Conversion of Internal reference voltage (V_{BGR}) is performed in single scan mode as below.

(1) Sampling time should be set more than or equal to 5 µs.
(2) After the internal reference voltage is selected for A/D conversion, when the ADST bit in ADCSR is set to 1, A/D conversion of internal reference voltage is started.
(3) When A/D conversion is completed, the A/D conversion result is stored into the A/D Internal reference voltage data register (ADOCDR) and an INTAD interrupt request is generated if the ADCSR.ADIE bit is 1 (INTAD interrupt upon scanning completion is enabled).
(4) The ADCSR.ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically cleared to 0 upon completion of A/D conversion. Then the 12-bit A/D converter enters a waiting state.

Figure 12-25. Example of Operation in Single Scan Mode (Internal Reference Voltage (V_{BGR}) Selected)



12.3.3 Continuous Scan Mode

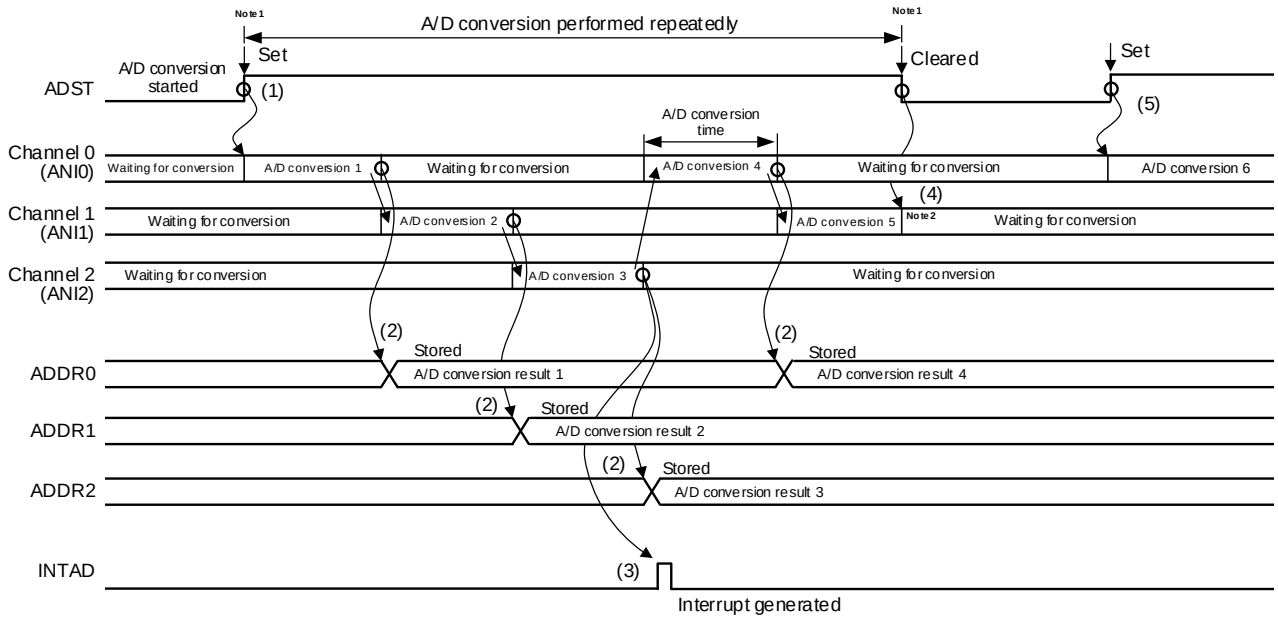
12.3.3.1 Basic Operation (without Channel-Dedicated Sample-and-Hold Circuits)

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as below.

In continuous scan mode, the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input, A/D conversion is performed for ANIn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy, ADDRxM).
(3) When A/D conversion of all the selected channels is completed, an INTAD interrupt request is generated if the ADCSR.ADIE bit is 1 (INTAD interrupt upon scanning completion enabled).
(4) The 12-bit A/D converter sequentially starts A/D conversion for ANIn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(5) The ADCSR.ADST bit is not automatically cleared to 0 and steps 2 and 3 are repeated as long as the bit remains 1 (A/D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state. When A/D conversion is stopped, the A/D conversion result of the channel that has not been converted does not remain.
(6) When the ADST bit is later set to 1 (A/D conversion start), A/D conversion is started again for ANIn channels
(7) selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.

Figure 12-26. Example of Operation in Continuous Scan Mode
(Basic Operation: ANI0, ANI1, and ANI2 Selected)



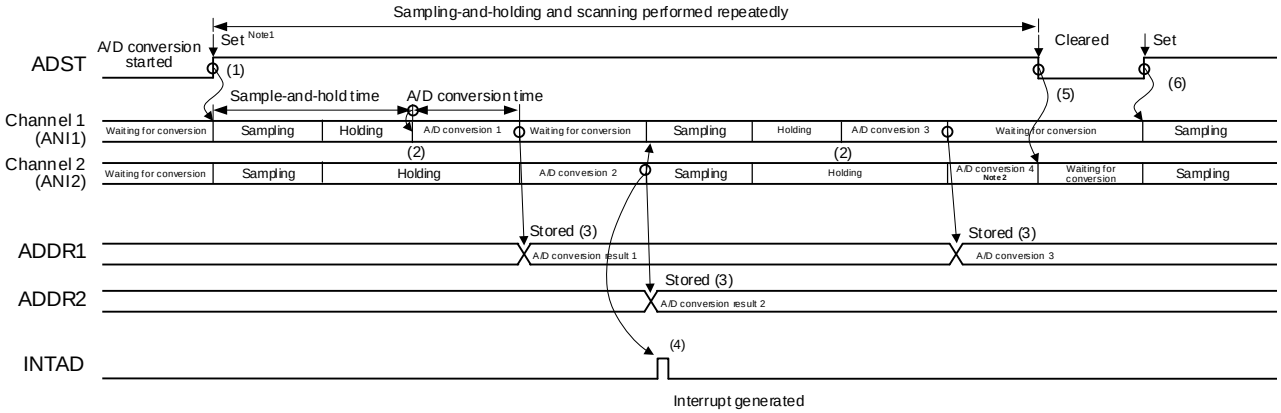
Notes 1. ↓ indicates the instruction is executed by software.
2. The converted data of A/D conversion 5 (in the processing of conversion) is ignored.

12.3.3.2 Basic Operation (with Channel-Dedicated Sample-and-Hold Circuits)

When the channel-dedicated sample-and-hold circuit is used, sample-and-hold operation is first performed, and then A/D conversion is performed repeatedly on the analog input of all the selected channels as below. The channels for which the channel-dedicated sample-and-hold circuits are to be used can be selected by the ADSHCR.SHANS[2:1] bits. In continuous scan mode, the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should be set to 0 (deselected).

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software or synchronous trigger input.
(2) After sample-and-hold operation, A/D conversion is performed for ANIn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy, ADDRxM).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled). At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.
(5) The ADCSR.ADST bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state. When A/D conversion is stopped, the A/D conversion result of the channel that has not been converted does not remain.
(6) When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input sampling is started again for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.

Figure 12-27. Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used)



Notes 1. ↓ indicates the instruction is executed by software.
2. The converted data of A/D conversion 4 (in the processing of conversion) is ignored.

Cautions 1. The maximum total of holding time for the channel dedicated sample-and-hold circuit and sampling time in A/D converter (t_{SPL}) is 10 µs. Set the sampling time of the ADCLK (A/D clock) and the target channel so that the hold time is within 10 µs.
2. In low current conversion mode, do not use the channel-dedicated sample-and-hold circuit.

12.3.4 Group Scan Mode

12.3.4.1 Basic Operation

In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in group A and B after scanning is started by a synchronous trigger as shown below. Scan operation of group A and B is the same as the scan operation in single scan mode.

In group scan mode, a synchronous trigger is selected for group A by the ADSTRGR.TRSA[5:0] bits, and for group B by the ADSTRGR.TRSB[5:0] bits. Different triggers should be used for group A and group B to prevent simultaneous scan of group A and B. Software trigger should not be used.

The scan target channels are selected by the ADANSA0 and ADANSA1 registers for group A; by the ADANSB0 and ADANSB1 registers for group B. Do not select the same channel in groups A and B in the ADANSA0/ADANSA1 and ADANSB0/ADANSB1 registers.

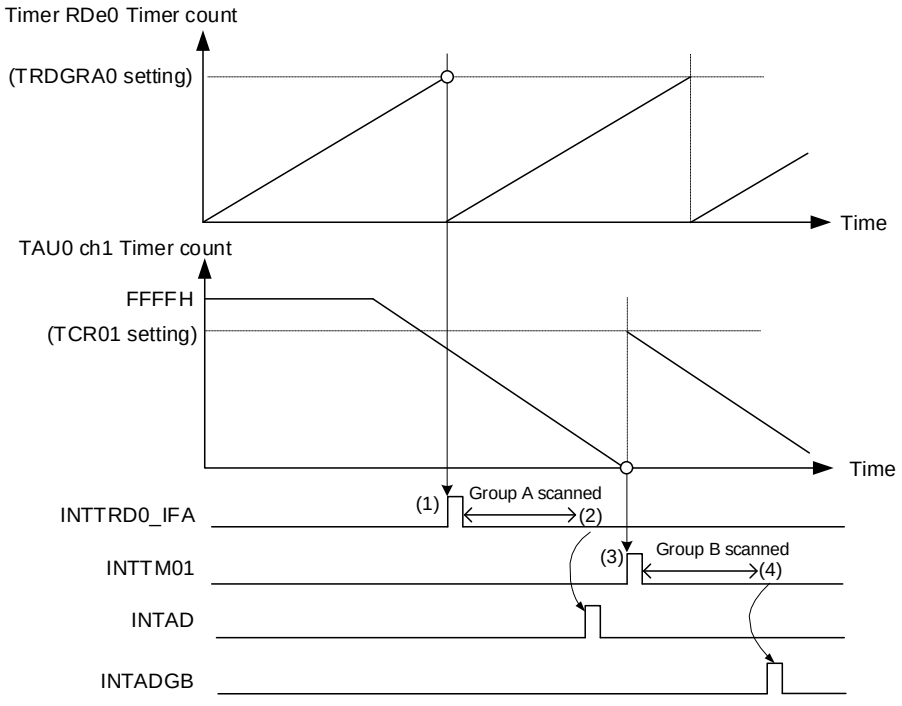
When self-diagnosis is selected in group scan mode, self-diagnosis is performed by individually scanning group A and group B.

The following describes the group scan mode operation started by a synchronous trigger from the timer function. In this setting, group A starts the conversion by the INTTRD0_IFA trigger from the Timer RDe0, group B starts the conversion by the INTTM01 trigger from the TAU0 ch1.

The operation is as follows:

(1) Scanning of group A is started by INTTRD0_IFA.
(2) When group A scanning is completed, an INTAD interrupt is output if the ADIE bit in ADCSR is 1 (INTAD interrupt upon scanning completion is enabled).
(3) Scanning of group B is started by INTTM01.
(4) When group B scanning is completed, an INTADGB interrupt is output if the GBADIE bit in ADCSR is 1 (INTADGB interrupt upon scanning completion is enabled).

Figure 12-28. Example of Operation in Group Scan Mode when Synchronous Triggers are Used



12.3.4.2 Group Priority Operation (when ADGSPCR.PGS = 1)

Group priority operation is performed by setting the ADGSPCR.PGS bit to 1 in group-scan mode. The priority order of the groups is group A > group B.

When setting the PGS bit in the ADGSPCR register to 1, follow the procedure described in **Figure 12-29**. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.

As the basic operation in group-scan mode, a trigger input generated during A/D conversion of group A, B is ignored, and the A/D conversion operation of each group is similar to the operation in single-scan mode.

In group priority operation, if a trigger for a priority group is input during scanning of a lower-priority group, A/D conversion for the lower-priority group is stopped and A/D conversion for the priority group is performed.

If the setting of the ADGSPCR.GBRSCN bit is 0, the lower-priority group enters a wait state when A/D conversion for the priority group completes. A trigger input of the lower-priority group generated during A/D conversion is ignored.

If the setting of the ADGSPCR.GBRSCN bit is 1, A/D conversion for the lower-priority group automatically restarts upon completion of A/D conversion for the priority group. A trigger input of the lower-priority group generated during A/D conversion on the priority group takes effect, and A/D conversion for the lower-priority group is automatically performed upon completion of A/D conversion on the priority group.

If the ADGSPCR.GBRSCN bit is 1 and the ADGSPCR.LGRRS bit is 0, A/D conversion for the lower-priority group is restarted from the first channel. If the setting of the ADGSPCR.LGRRS bit is 1, A/D conversion for the lower priority group is restarted from the channel for which the conversion stopped. However, if the self-diagnosis function is used, the A/D conversion is restarted from the channel for which the conversion stopped after self-diagnosis completed.

Table 12-19 summarizes operations in response to the input of a trigger during A/D conversion with the settings of the ADGSPCR.GBRSCN bit.

If the setting of the ADGSPCR.GBRP bit is 1, A/D conversion operation for the lowest-priority group is to continuously perform single scans.

For the trigger settings in group-scan mode, select a synchronous trigger for group A by using the ADSTRGR.TRSA[5:0] bits, a synchronous trigger for group B by using the ADSTRGR.TRSB[5:0] bits. Each trigger must be different from each other. Set the ADSTRGR.TRSB[5:0] bits to 3FH when setting the ADGSPCR.GBRP bit to 1.

Figure 12-29. Flowchart for ADGSPCR.PGS Bit Setting

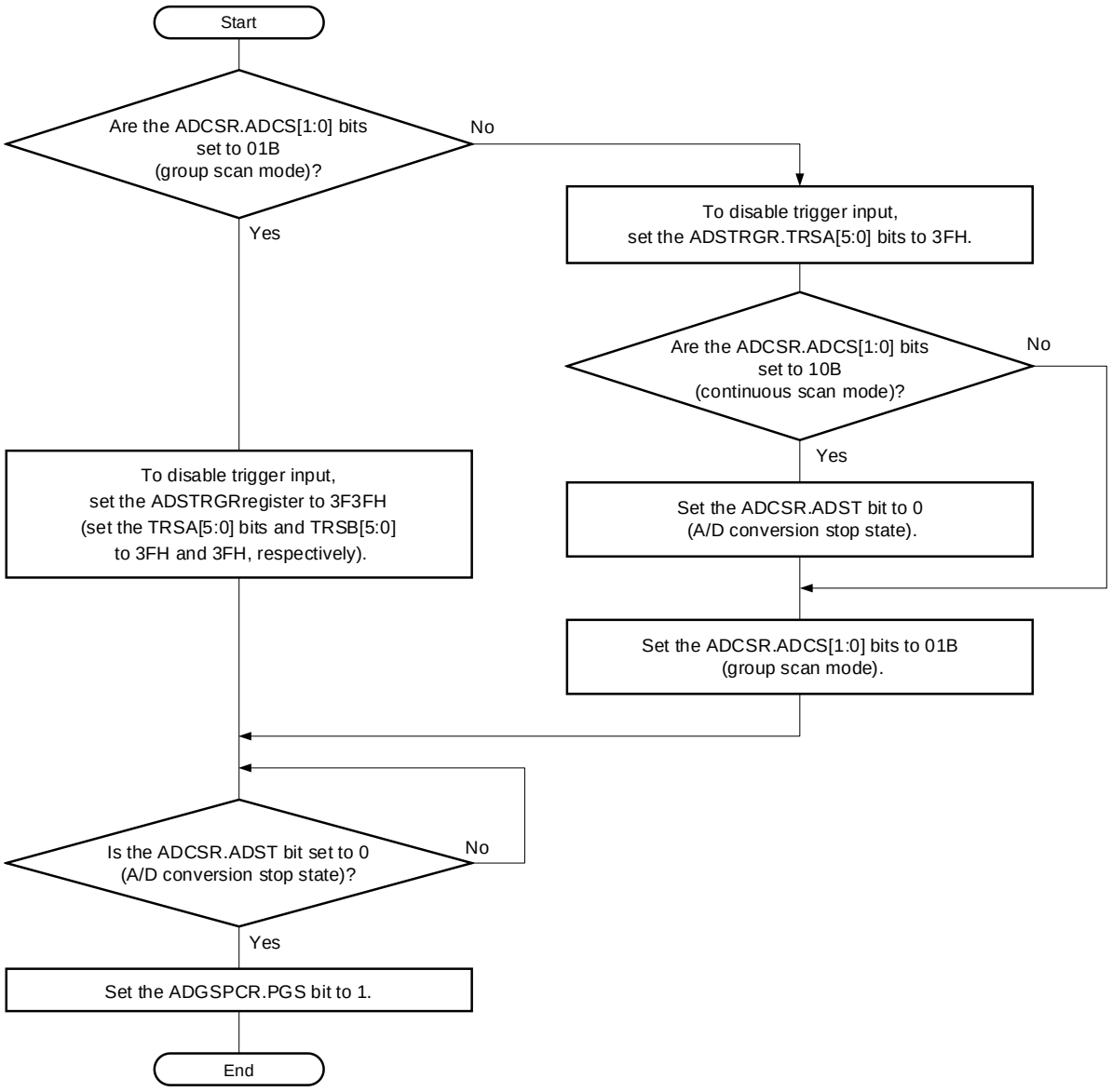


Table 12-19. Control of A/D Conversion Operation According to ADGSPCR.GBRSCN Bit Setting

A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is invalid.	Trigger input is invalid.
	Input of trigger for group B	Trigger input is invalid.	A/D conversion for group B is performed after A/D conversion for group A completes.
When A/D conversion for group B is in progress	Input of trigger for group A	A/D conversion for group B is discontinued and A/D conversion for group A starts.	- A/D conversion for group B is discontinued and A/D conversion for group A starts. - A/D conversion for group B starts after A/D conversion for group A completes.
	Input of trigger for group B	Trigger input is invalid.	Trigger input is invalid.

To use group priority operation mode, select the operation mode to be implemented and set the registers according to the following table.

Table 12-20. Group Priority Operation Setting and Operation Mode for Two Groups (ADGSPCR.PGS = 1)

ADGSPCR Register			Operation Category
GBRSCN	LGRRS	GBRP	
0	x Note 2	0	When a trigger of group A is input, A/D conversion for group B is terminated (and will not be restarted).
1	0	0	After A/D conversion for group B stopped, when A/D conversion for group A completes, A/D conversion for the group B channels selected in the ADANSB0 and ADANSB1 registers restart according to the conversion order of smaller channel number.
1	1	0	After A/D conversion for group B stopped, when A/D conversion for group A completes, A/D conversion for the group B channels selected in the ADANSB0 and ADANSB1 registers restart according to the conversion order of smaller channel number, beginning from the channel for which A/D conversion stopped. Note 1
x Note 2	0	1	Single scanning for group B is continuously performed without a start trigger input. After A/D conversion for group B stopped, when A/D conversion for group A completes, single scanning for the channels selected in the ADANSB0 and ADANSB1 registers restart according to the conversion order of smaller channel number.
1	1	1	Single scanning for group B is continuously performed without a start trigger input. After A/D conversion for group B stopped, when A/D conversion for group A completes, single scanning for the channels selected in the ADANSB0 and ADANSB1 registers restart according to the conversion order of smaller channel number, beginning from the channel for which A/D conversion stopped. Note 1

Notes 1. When the self-diagnosis function is enabled (ADCER.DIAGM = 1), A/D conversion for the channel that has been stopped is started after self-diagnosis is performed.

2. x: don't care.

Operation examples 1 to 3 (**12.3.4.2.1** to **12.3.4.2.3**) show group priority operations in group-scan mode (when ADGSPCR.GBRSCN = 1, ADGSPCR.LGRRS = 0, and ADGSPCR.GBRP = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

Operation examples 4 (**12.3.4.2.4**) shows group priority operations in group-scan mode (when ADGSPCR.GBRSCN = 0, ADGSPCR.LGRRS = 0, and ADGSPCR.GBRP = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

Operation examples 5 (**12.3.4.2.5**) shows group priority operations in group-scan mode (when ADGSPCR.LGRRS = 0, and ADGSPCR.GBRP = 1) when channel 0 is selected for group A and channels 1 to 2 are selected for group B.

12.3.4.2.1 Operation Example 1 (“Group A trigger input during group B scan” when rescanning is enabled)

(1) When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order from the channel with the smallest number n.

(2) On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D data registers (ADDRy, ADDRxM).

(3) When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1. Then A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order from the channel with the smallest number n. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D data registers (ADDRy, ADDRxM).

(4) On completion of A/D conversion on the channels, the result is stored in the corresponding A/D data registers (ADDRy, ADDRxM).

(5) If the setting of the ADCSR.ADIE bit is 1 (enabling interrupt generation upon completion of scanning), a scan end interrupt request is generated.

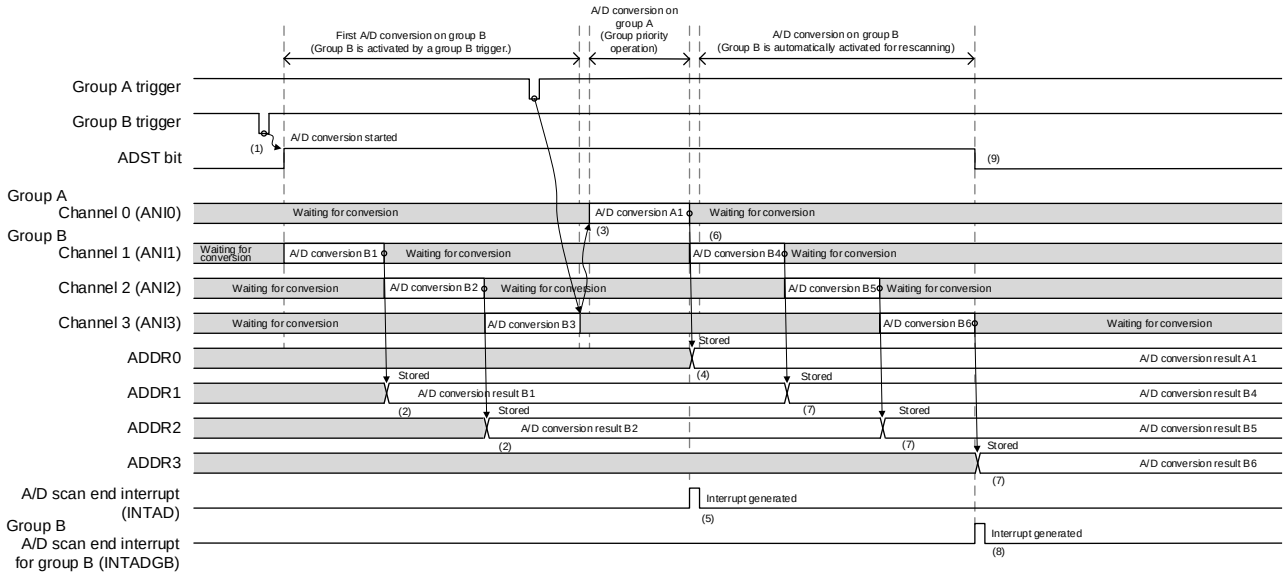
(6) If the setting of the ADGSPCR.GBRSCN bit is 1 (enabling scanning of the group that was stopped in group priority operation), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order from the channel with the smallest number n while the ADCSR.ADST remains 1.

(7) On completion of A/D conversion on the channels, the result is stored in the corresponding A/D data registers (ADDRy, ADDRxM).

(8) If the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan), a group B scan end interrupt request is generated.

(9) When A/D conversion for all the channels completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state.

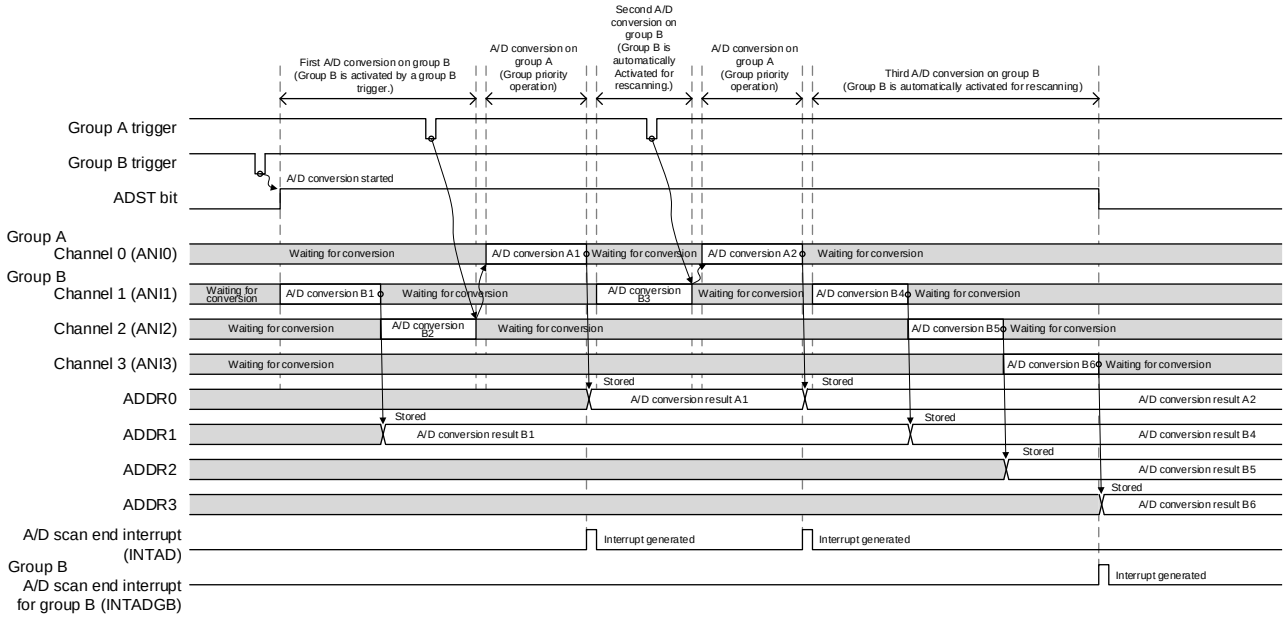
Figure 12-30. Example of Group Priority Operation 1: Group A Trigger Input During Group B Scanning when Rescanning is Enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)



12.3.4.2.2 Operation Example 2 (“Group A trigger input during rescanning of group B” when rescanning is enabled)

Figure 12-31 shows the operation when a group A trigger is input during rescanning operation for group B. Even during rescanning operation, when a trigger for group A is input, A/D conversion on group B stops and A/D conversion for group A starts. A/D conversion for group B starts after A/D conversion for group A completes. Operations for setting the ADCSR.ADST bit, storing the A/D conversion result in the corresponding A/D data registers (ADDRy, ADDRxM), and generating interrupt requests are the same as those in operation example 1.

Figure 12-31. Example of Group Priority Operation 2: Group A Trigger Input During Rescanning of Group B when Rescanning is Enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)



12.3.4.2.3 Operation Example 3 (“Group B trigger input during group A scan” when rescanning is enabled)

(1) When input of a trigger for group A sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order from the channel with the smallest number n.

(2) When a trigger for group B is input during A/D conversion for group A, group B is ready for A/D conversion.

(3) On completion of A/D conversion for each channel in group A, the result is stored in the corresponding A/D data registers (ADDRy, ADDRxM).

(4) If the setting of the ADCSR.ADIE bit is 1 (enabling interrupt generation upon completion of scanning), a scan end interrupt request is generated.

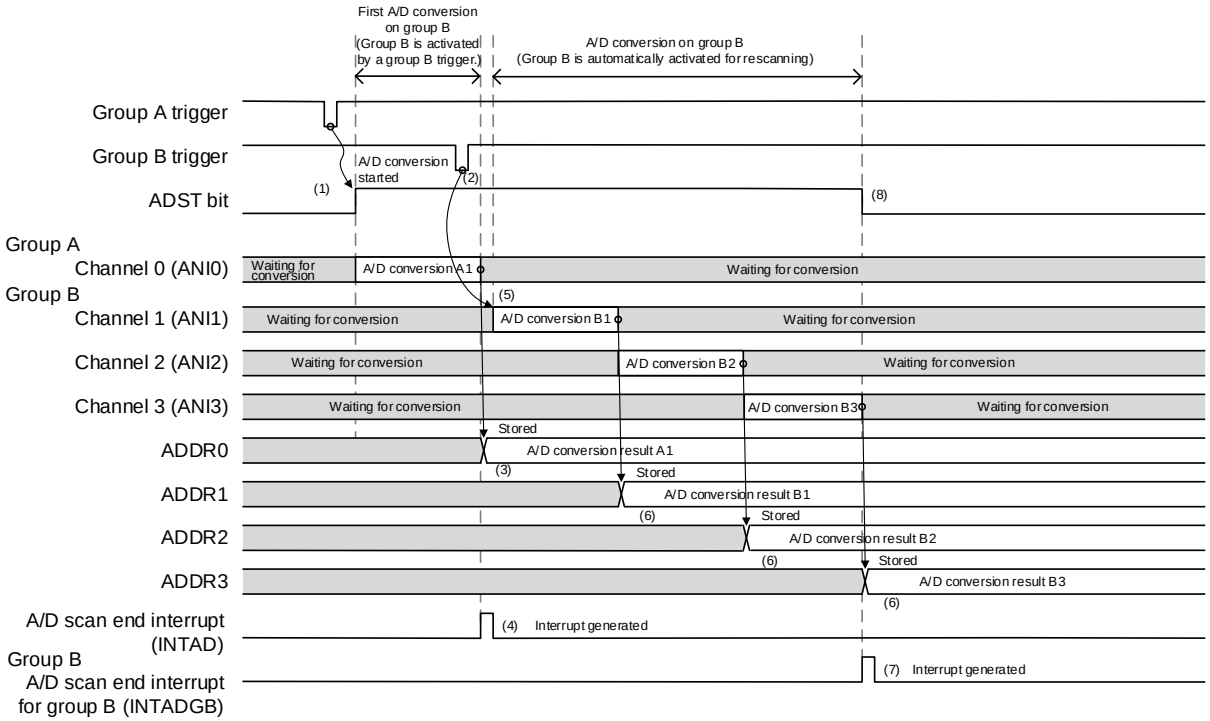
(5) When A/D conversion for group A completes, while the ADCSR.ADST bit remains 1, A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order from the channel with the smallest number n. (As with the case of operation example 1-1, if a trigger for group A is input during A/D conversion for group B, A/D conversion for group A starts. Then A/D conversion for group B starts upon completion of A/D conversion for group A.)

(6) On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D data registers (ADDRy, ADDRxM).

(7) Upon completion of A/D conversion for group B, a group B scan end interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan).

(8) When A/D conversion for all the channels completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state.

Figure 12-32. Example of Group Priority Operation 3: Group B Trigger Input During Group A Scan when Rescanning is Enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)

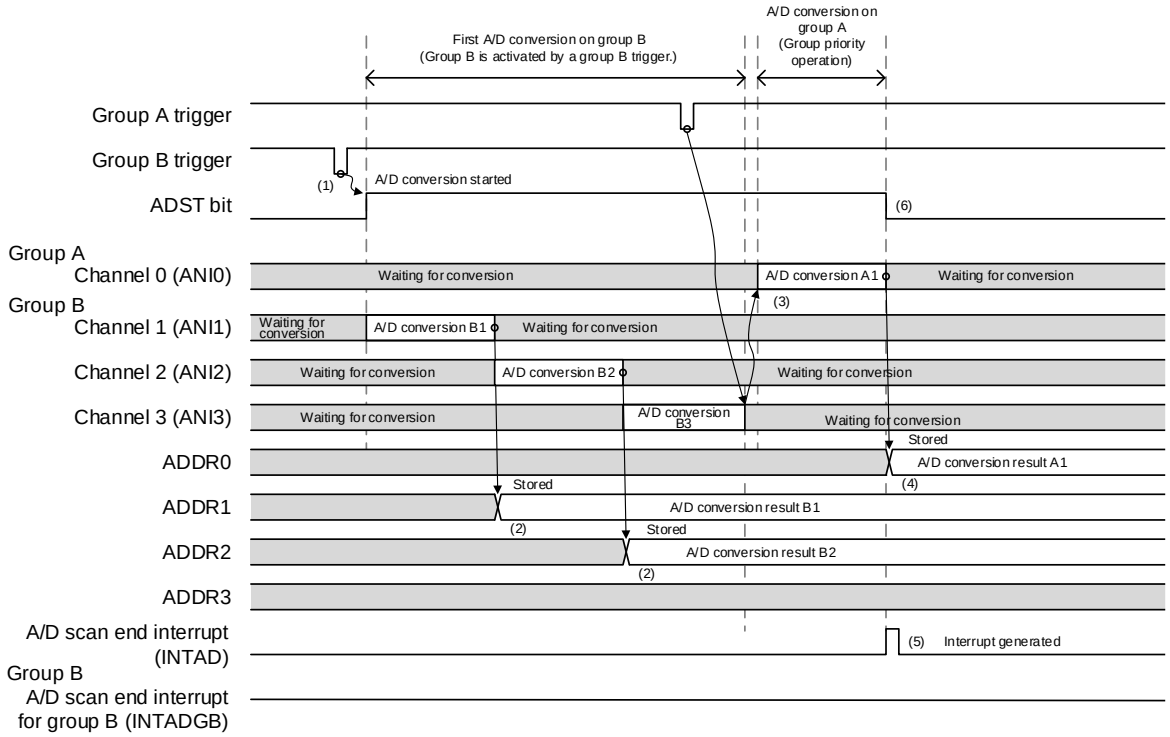


12.3.4.2.4 Operation Example 4 (“Group A trigger input during group B scan” when rescanning is disabled)

Operation example 4 shows the group priority operation in group-scan mode (when ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

(1) When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order from the channel with the smallest number n.
(2) On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D data registers (ADDRy, ADDRxM).
(3) When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order from the channel with the smallest number n. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D data registers (ADDRy, ADDRxM).
(4) On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D data registers (ADDRy, ADDRxM).
(5) On completion of A/D conversion for group A, a scan end interrupt request is generated if the setting of the ADCSR.ADIE bit is 1(enabling interrupt generation upon completion of scanning).
(6) When A/D conversion for group A completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state. A/D conversion for group B is not performed until a trigger for group B is input the next time.

Figure 12-33. Group Priority Operation Example 4: Group A Trigger is Input During Group B Scan when Rescanning is Disabled (when ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)



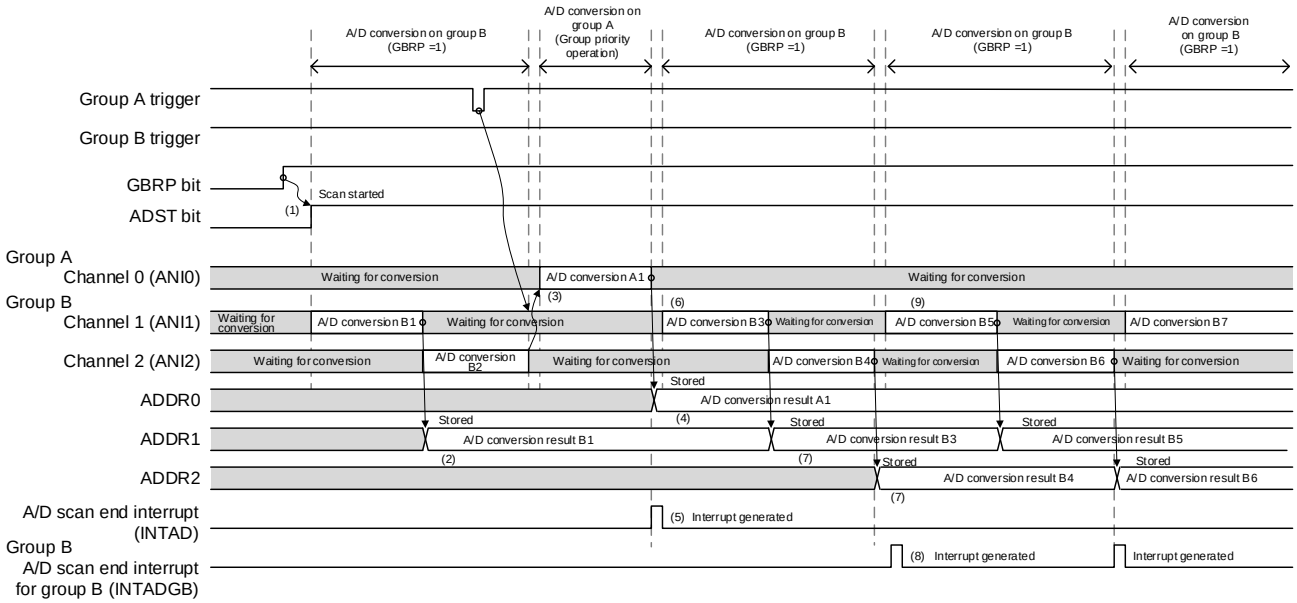
12.3.4.2.5 Operation Example 5 (Continuously activating single-scan operation for group B)

Operation example 5 shows the group priority operation in group-scan mode (when ADGSPCR.GBRP = 1, and ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 and 2 are selected for group B.

(1) When ADGSPCR.GBRP = 1 is set, the ADCSR.ADST bit is set to 1 (starting A/D conversion) and A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order from the channel with the smallest number n.
(2) On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D data registers (ADDRy, ADDRxM).
(3) When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order from the channel with the smallest number n. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D data registers (ADDRy, ADDRxM).
(4) On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D data registers (ADDRy, ADDRxM).
(5) On completion of A/D conversion for group A, a scan end interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (enabling interrupt generation upon completion of scanning).
(6) If ADGSPCR.GBRP = 1 is set (performing single scan continuously), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order from the channel with the smallest number n while the ADCSR.ADST remains 1 (starting A/D conversion).
(7) On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D data registers (ADDRy, ADDRxM).
(8) If the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan), a group B scan end interrupt request is generated.
(9) If ADGSPCR.GBRP = 1 is set (performing single scan continuously), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order from the channel with the smallest number n while the ADCSR.ADST remains 1 (starting A/D conversion).

Steps 6 to 9 are repeated as long as the ADGSPCR.GBRP bit remains 1. Do not clear the ADCSR.ADST bit as long as the ADGSPCR.GBRP bit is 1. To forcibly stop A/D conversion while ADGSPCR.GBRP = 1, follow the procedure shown in **12.7.3 Procedure for Stopping A/D Conversion**.

Figure 12-34. Group Priority Operation Example 5: Continuously Activating Single Scan for Group B [Note] (when ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 0)



Note To continuously activate single-scan operation for group B, disable group B trigger input.

12.3.5 Analog Input Sampling Time and Scan Conversion Time

Scan conversion can be activated either by software or synchronous trigger input. After the start-of-scanning-delay time (t_D) has elapsed, processing by the channel-dedicated sample-and-hold circuits, processing for disconnection detection assistance, and processing of conversion for self-diagnosis proceed, and this is followed by processing for A/D conversion.

Figure 12-35 shows the timing of scan conversion in response to a software trigger or synchronous trigger.

The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay time (t_D), channel-dedicated sample-and-hold circuit processing time (t_{SPLSH}) **Note 1**, disconnection detection assistance processing time (t_{DIS}) **Note 2**, self-diagnosis A/D conversion processing time (t_{DIAG}) **Note 3**, wait time between last channel conversion end and self-diagnosis sampling start in continuous scan mode (t_{DSD}), A/D conversion processing time (t_{CONV}), channel-dedicated sample-and-hold circuit end time (t_{SHED}) **Note 4**, and end-of-scanning-delay time (t_{ED}).

The A/D conversion processing time (t_{CONV}) consists of sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTRn register.

The time for conversion by successive approximation (t_{SAM}) is at 32 ADCLK states during high-speed conversion operation. **Table 12-21** shows the scan conversion time.

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n)^{\text{Note 5}} + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED} plus t_{SHED}.

The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed to

$$t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + t_{DSD} + (t_{CONV} \times n)^{\text{Note 5}} + t_{SHED}$$

Notes 1. When no channel-dedicated sample-and-hold circuits are used, $t_{SPLSH} = 0$.

2. When disconnection detection assistance is not selected, $t_{DIS} = 0$

3. When the self-diagnosis function is not used, $t_{DIAG} = 0$, $t_{DSD} = 0$.

4. When no channel-dedicated sample-and-hold circuits are used, $t_{SHED} = 0$. Here, continuous scan mode is assumed. In single scan mode, t_{SHED} is included in the end-of-scanning-delay time (t_{ED}).

5. When input sampling time (t_{SPL}) of all selected channels are the same, this element equals $t_{CONV} \times n$, if each channel has a different sampling time, this element equals sum of t_{SPL} and t_{SAM} which are set to each selected channel respectively.

Table 12-21. Times for Conversion during Scanning (in Numbers of Cycles of ADCLK and f_{CLK})

Item			Symbol		Type/Conditions		Unit
					Synchronous Trigger Note 4	Software Trigger	
Scan start processing time Notes 1, 2	A/D conversion of a priority group by the group priority operation	With the suspension of low-priority group (A trigger of A/D conversion on a priority group stops the low-priority group and starts the priority group.)	t_D		2 f_{CLK} + 6 × ADCLK	–	Cycle
		Without suspension of low-priority group (activation by a trigger of A/D conversion on a priority group)			2 f_{CLK} + 4 × ADCLK	–	
	A/D conversion when self-diagnosis is enabled	A/D conversion for self-diagnosis is to be started.			2 f_{CLK} + 6 × ADCLK	6 × ADCLK	
	Other than above				2 f_{CLK} + 4 × ADCLK	4 × ADCLK	
Channel-dedicated sample-and- hold processing time Note 1	Sampling time		t_{SPLSH}	t_{SH}	The setting of ADSHCR.SSTSH[7:0] (initial value = 1AH) × ADCLK		
	Wait time between sampling and A/D conversion			t_W	13 × ADCLK		
Disconnection detection assistance processing time			t_{DIS}		ADDISCR.ADNDIS[3:0] set value (initial value 07H) × ADCLK		
Self-diagnosis conversion processing time Note 1	Sampling time		t_{DIAG}	t_{SPL}	The setting of ADSSTR0 (initial value = 0DH) × ADCLK + 0.5 × ADCLK Note 3		
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	31.5 × ADCLK		
	Normal A/D conversion is to be started after completion of self-diagnosis conversion.			t_{DED}	2 × ADCLK		
	A/D conversion for self-diagnosis is to be started after completion of conversion for continuous scan on the last channel specified.		t_{DSD}		2 × ADCLK		
A/D conversion processing time Note 1	Sampling time		t_{CONV}	t_{SPL}	The setting of ADSSTRn (n = 0 to 15, L, O) (initial value = 0DH) × ADCLK + 0.5 × ADCLK Note 3		
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	31.5 × ADCLK		
Channel-dedicated sample-and-hold end processing time			t_{SHED}		3 × ADCLK		
Scan end processing time Note 1			t_{ED}		1 f_{CLK} + 3 × ADCLK		

Notes 1. For t_D, t_{SPLSH}, t_{DIAG}, t_{CONV}, and t_{ED}, see **Figure 12-35**.

2. This is the maximum time required from software writing or trigger input to A/D conversion start.

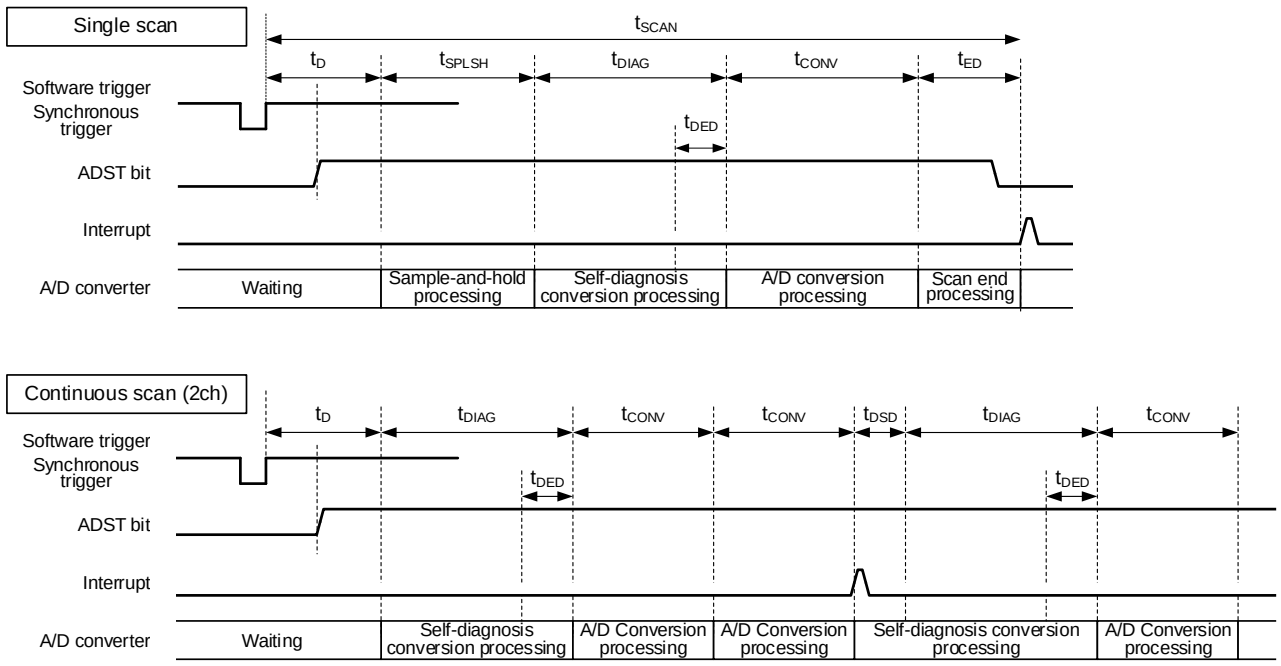
3. The sampling time is specified for ANI channels in the ADSSTRn register.
The setting range of the sampling time differs between the high-speed analog input channel and the analog input channel.
The sampling time also differs between the high-speed conversion mode and the low current conversion mode.
For details, see **12.2.12 A/D Sampling State Register n (ADSSTRn)**.

4. The timing delay between output of timer and trigger input is not included.

5. In low current conversion mode, the successive conversion time (t_{SAM}) is 40.5 × ADCLK.

Caution In high-speed conversion mode (ADCSR.ADHSC bit is 0), the A/D clock (ADCLK) must be 2 MHz or higher. In low current conversion mode (ADHSC bit is 1), the A/D clock must be set to a frequency between 2 MHz and 8 MHz.

Figure 12-35. Scan Conversion Timing (Activated by Software or Synchronous Trigger)



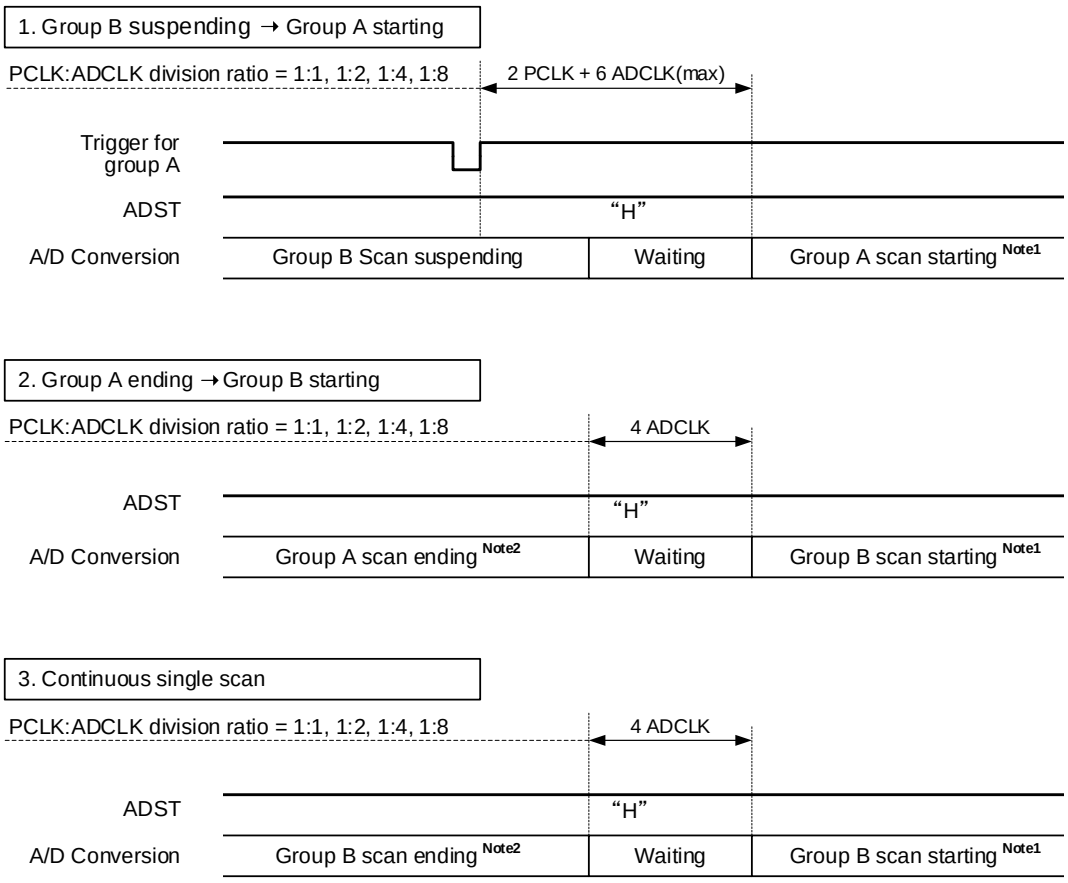
12.3.5.1 Scan Suspending/Starting Timings in Group Priority Operation

There are the following timings of suspending/starting the scan in the group priority operation:

1. Timing of suspending the scan of a low-priority group and starting the scan of a priority group
2. Timing of restarting the suspended scan of a low-priority group; and timing of starting, upon completion of the scan of the high-priority group, the scan of a low-priority group by a trigger of the low-priority group received during the scan of the high-priority group
3. Timing in which a low-priority group performs single scan continuously

Figure 12-36 shows these timings.

Figure 12-36. Timing of Suspending/Starting Scan in Group Priority



Notes 1. Without Scan start delay time

2. With Scan end delay time

Remark PCLK: CPU/peripheral hardware clock (f_{CLK})

12.3.6 Usage Example of A/D Data Register Automatic Clearing Function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADDRxM, ADRD, ADOCDR) to 0000H when the A/D data registers are read by the CPU or DTC.

This function enables detection of update failures of the A/D data registers (ADDRy, ADDRxM, ADRD, ADOCDR). The following describes the examples in which the function to automatically clear the ADDRy register is enabled and disabled.

In a case where the ADCER.ACE bit is 0 (automatic clearing disabled), if the A/D conversion result (0222H) is not written to the ADDRy register for some reason, the old data (0111H) will be the ADDRy value. Furthermore, if this ADDRy value is read into a general register using an A/D scan end interrupt, the old data (0111H) can be saved in the general register. When checking whether there is an update failure, it is necessary to frequently save the old data in the RAM or a general register.

In a case where the ADCER.ACE bit is 1 (automatic clearing enabled), when ADDRy = 0111H is read by the CPU or DTC, ADDRy is automatically cleared to 0000H. After that, if the A/D conversion result 0222H cannot be transferred to ADDRy for some reason, the cleared data (0000H) remains as the ADDRy value. If this ADDRy value is read into a general register using an A/D scan end interrupt at this point, 0000H will be saved in the general register.

Occurrence of an ADDRy update failure can be determined by simply checking that the read data value is 0000H.

12.3.7 A/D-converted Value Addition/Average Mode

In A/D-converted value addition mode, the same channel is A/D-converted 2, 3, 4, or 16 consecutive times and the sum of the converted values is stored in the data register. In A/D-converted value average mode, the same channel is A/D-converted two or four consecutive times and the mean of the converted values is stored in the data register. Using an average can improve the accuracy of A/D conversion, depending on the types of noise components that are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.

The A/D-converted value addition/average mode can be specified when A/D conversion of the channel select analog input or internal reference voltage is selected.

12.3.8 Disconnection Detection Assist Function

This A/D converter incorporates the function to fix the charge for sampling capacitance to the specified state before start of A/D conversion. This function enables disconnection detection in wiring of analog inputs.

Figure 12-37 shows the A/D conversion operation when the disconnection detection assist function is used.

Figure 12-38 shows an example of disconnection detection when precharge is selected. **Figure 12-39** shows an example of disconnection detection when discharge is selected.

It's necessary for stable operation of the disconnection detection assist function to place the external pullup or pulldown via high resistance.

Figure 12-37. Operation of A/D Conversion when the Disconnection Detection Assist Function is Used

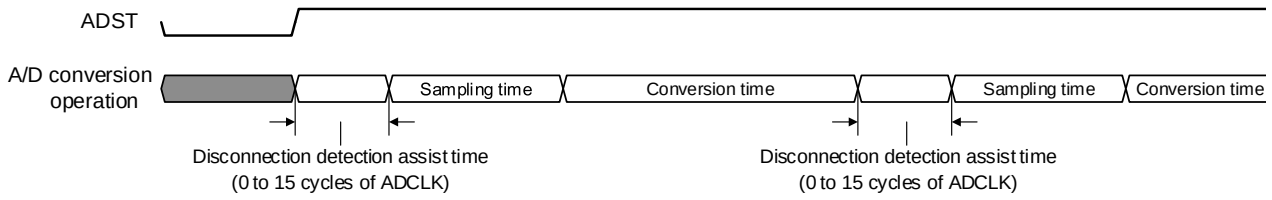
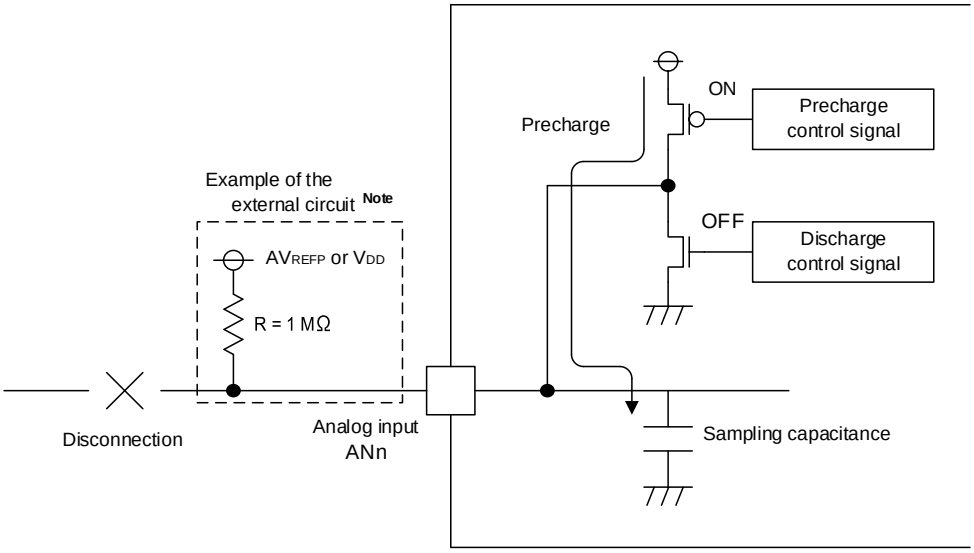
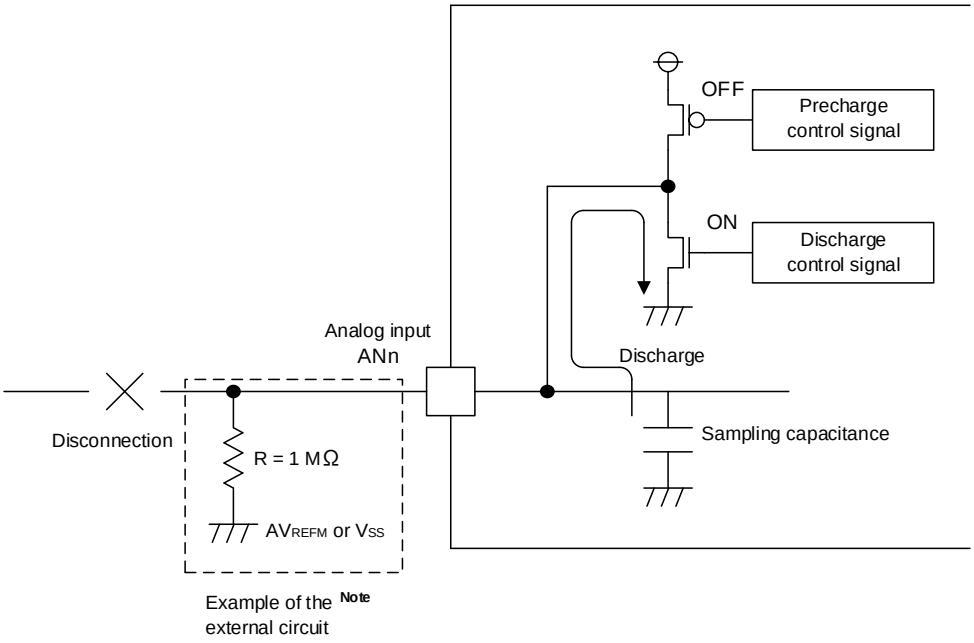


Figure 12-38. Example of Disconnection Detection when Precharge is Selected



Note The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 12-39. Example of Disconnection Detection when Discharge is Selected



Note The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

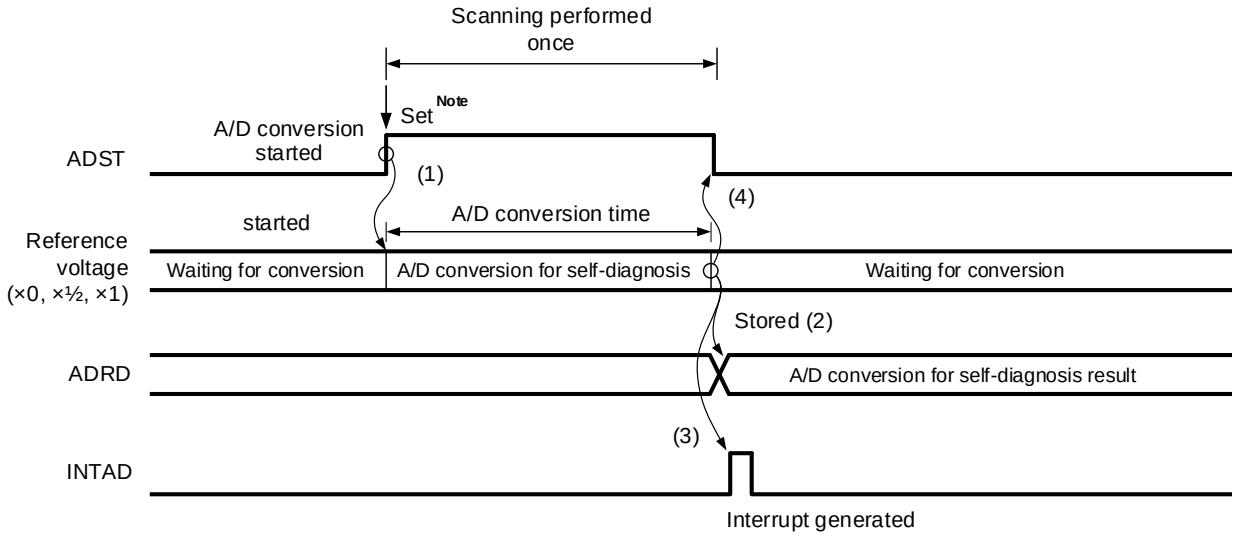
12.3.9 Self-diagnosis Function

This A/D converter incorporates the self-diagnosis function.

To set both of DIAGM and DIAGLD bits of ADCER register to 1, and with the voltage setting of ADCER.DIAGVAL bits, this function is enabled. When self-diagnosis is selected, A/D conversion is performed for the reference voltage supplied to the 12-bit A/D converter as below operation flow.

(1) A/D conversion for self-diagnosis is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software trigger input.
(2) When A/D conversion for self-diagnosis is completed, the result of A/D conversion is stored in the A/D self- diagnosis data register (ADRD).
(3) When A/D conversion of self-diagnosis is completed, an INTAD interrupt request is generated if the ADCSR.ADIE bit is 1 (Enabling INTAD interrupt upon scanning completion).
(4) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of self-diagnosis is completed. Then the 12-bit A/D converter enters a wait state.

Figure 12-40. Example of Self-diagnosis when Single Scan is Selected



Note ↓ indicates the instruction is executed by software.

12.3.10 Starting A/D Conversion with Synchronous Trigger from Peripheral Function

The A/D conversion can be started by a synchronous trigger from the event link controller and timer functions. To start the A/D conversion by a synchronous trigger, the ADCSR.TRGE bit should be set to 1, and the relevant sources should be selected by the ADSTRGR.TRSA[5:0] and TRSB[5:0] bits.

12.4 Interrupt Sources and DTC Transfer Requests

12.4.1 Interrupt Requests

The 12-bit A/D converter can send scan end interrupt requests INTAD and INTADGB to the CPU.

Setting the ADCSR.ADIE bit to 1 and 0 enables and disables an INTAD interrupt, respectively. Similarly, the ADCSR. GBADIE bit can be used for enabling/disabling INTADGB.

In addition, the DTC can be activated when an INTAD interrupt is generated. Using an INTAD interrupt to allow the DTC to read the converted data enables continuous conversion without burden on software. For details on the DTC settings, see **CHAPTER 20 DATA TRANSFER CONTROLLER (DTC)**.

12.5 Event Link Function (For RL78/F25 only)

12.5.1 12-bit A/D converter operation by event from the ELC

The 12-bit A/D converter can be started by the predetermined event by setting ELSELRn of the ELC.

12.5.2 Note on 12-bit A/D Converter When an Events Input from the ELC

If an event occurs during A/D conversion, the event is disabled.

12.6 Reference Voltage

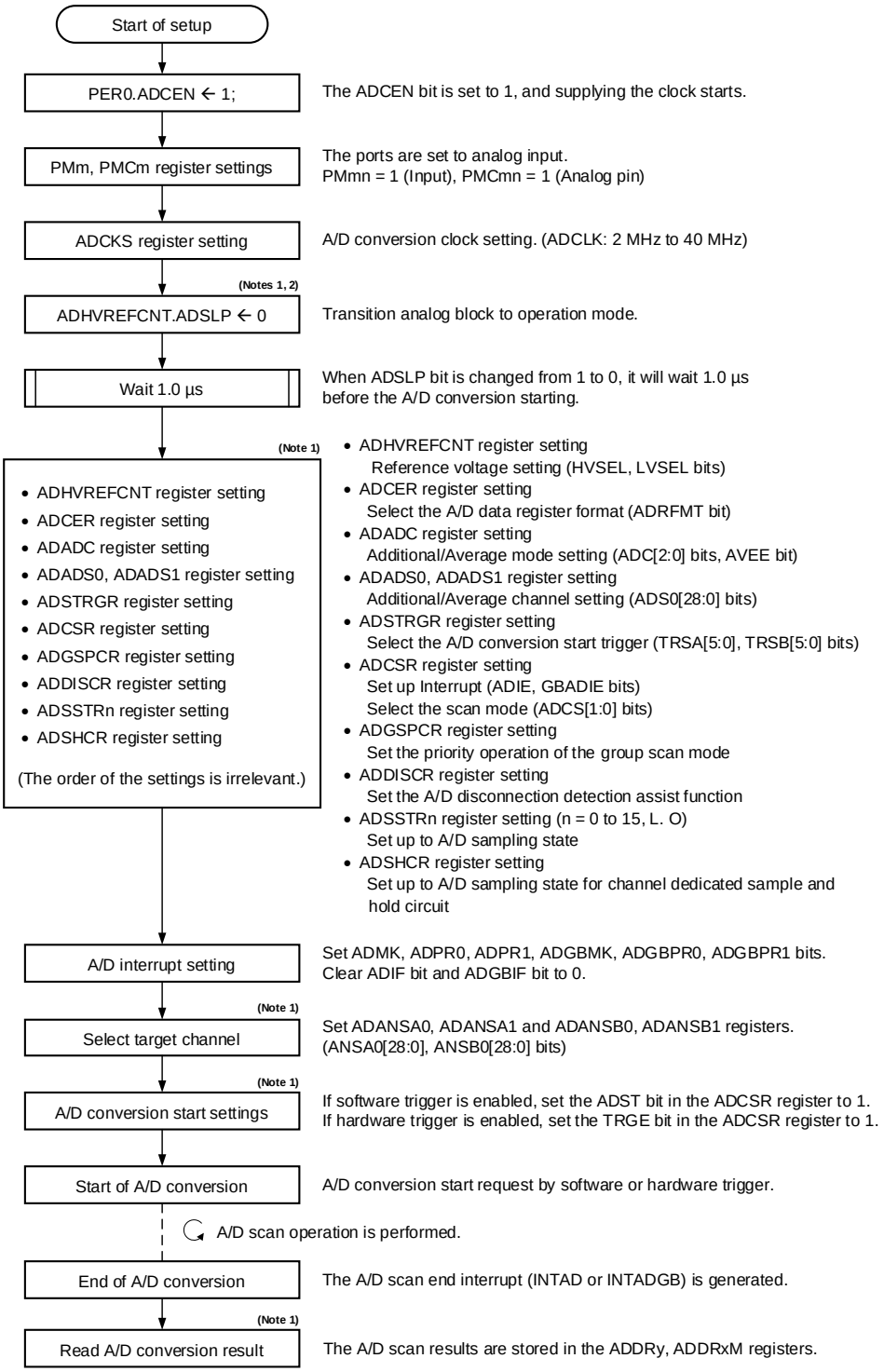
The high-potential reference voltage of the A/D converter is the external pin input (AV_{REFP}). Also, the low-potential reference voltage is the external pin input (AV_{REFM}). See **12.2.16 A/D High-/Low-potential Reference Voltage Control Register (ADHVREFCNT)**.

12.7 Usage Notes

12.7.1 Procedure for A/D Converter Initialization

Figure 12-41 on the next page shows the A/D converter initialization procedure.

Figure 12-41. A/D Converter Initialization Procedure



Notes 1. Before accessing the target register, set the window number of the register to be accessed in ADWINR register.

2. ADHSC bit in the ADCSR register must be set to 1 (low current conversion mode) when used at 1.8 V ≤ V_{DD} < 2.7 V. To change the ADHSC bit from 0 to 1, follow the steps below.

(1) Set ADSLP bit to 1 (A/D analog block standby state).
(2) Wait at least 0.2 µs.
(3) Set ADHSC bit to 1 (Low current conversion mode).
(4) Wait at least 4.8 µs.
(5) Set ADSLP bit to 0 (A/D analog block operation mode).

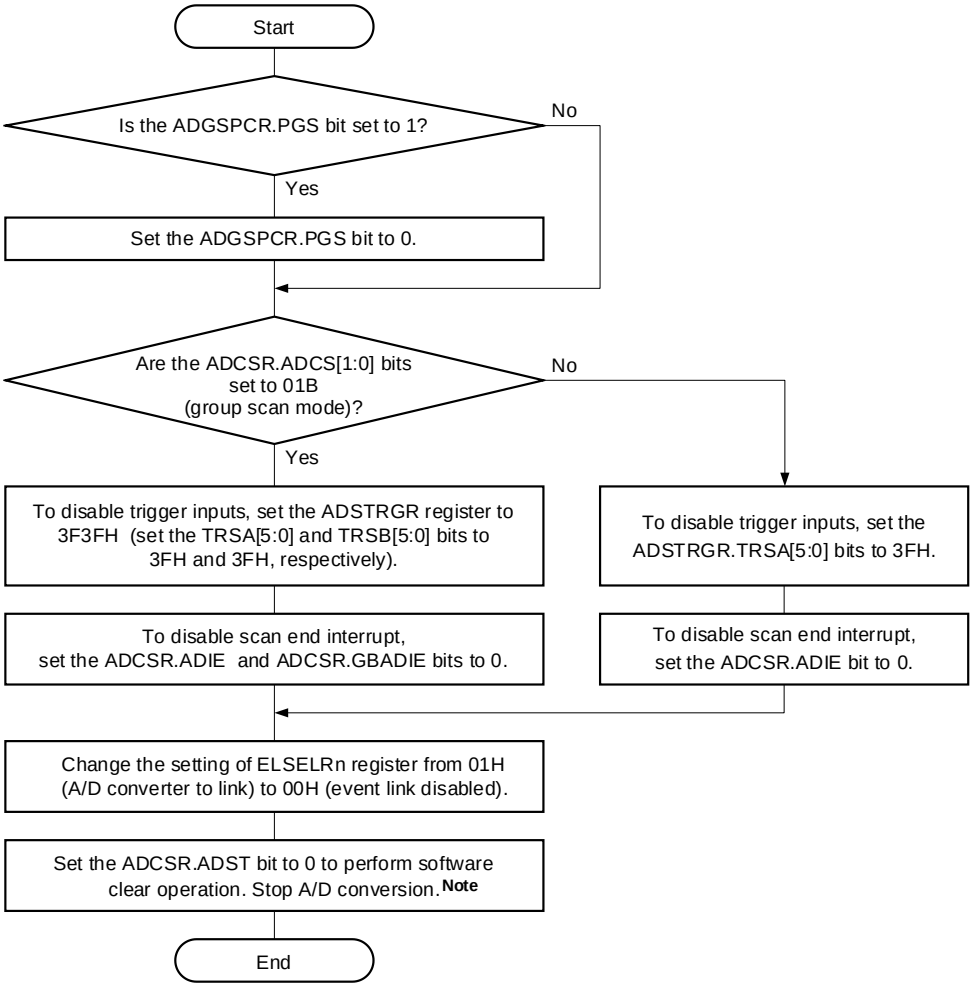
12.7.2 Notes on Reading Data Registers

The A/D data registers, A/D internal reference voltage data register, and A/D self- diagnosis data register should be read in 16-bit memory manipulation instruction.

12.7.3 Procedure for Stopping A/D Conversion

To stop A/D conversion when a synchronous trigger has been selected as the condition for starting A/D conversion, follow the procedure in **Figure 12-42**.

Figure 12-42. Procedure for Clear Operation by Software through the ADCSR.ADST Bit



Note From software clearing to stopping of scanning takes two clock cycles of ADCLK. Accordingly, wait for at least two clock cycles of ADCLK before making any of the following settings after software clearing.

- Enabling the scan end interrupt
- Selecting operation of peripheral function 1 (A/D converter) to link
- Using software to starting A/D conversion
- Enabling trigger input

Remark n = 0 to 25

12.7.4 Notes on Self-diagnosis Function

Initialize or re-set the voltage state for self-diagnosis as necessary.

Set ADCER.DIAGLD to 1 and select the voltage state for self-diagnosis by using ADCER.DIAGVAL[1:0] to re-set the voltage state for self-diagnosis.

12.7.5 A/D Conversion Restarting Timing and Termination Timing

It takes a maximum of six ADCLK cycles for the idle analog unit of the 12-bit A/D converter to be restarted by setting the ADCSR.ADST bit to 1. It takes a maximum of three ADCLK cycles for the operating analog unit of the 12-bit A/D converter to be terminated by setting the ADCSR.ADST bit to 0.

12.7.6 Notes on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data in the case that the CPU does not complete reading the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

12.7.7 Clock Supply Stop Setting

Operation of the 12-bit A/D converter can be disabled or enabled by setting peripheral enable register 0 (PER0).

The initial setting is for operation of the 12-bit A/D converter to be halted. Register access is enabled by releasing the clock supply stop state.

After the clock supply stop state is released, set the analog block to operation mode (ADSLP = 0) and wait for 1 μs or more to start A/D conversion.

12.7.8 Notes on Entering Low Power Consumption States

Be sure to stop A/D conversion before stopping supply of the clock signals or placing the chip in STOP mode. Here, set the ADCSR.ADST bit to 0, and secure certain period of time until the analog unit of the 12-bit A/D converter is stopped.

Follow the procedure for clear operation by software through the ADCSR.ADST bit, shown in **Figure 12-42**.

After that, wait for three clock cycles of ADCLK before stopping supply of the clock signals or placing the chip in STOP mode.

If the clock state is used in the “Clock state after transition” shown in the table below, put the analog block of A/D converter on standby state before the clock transition.

Clock state before transition	CPU operation on high-speed system clock (f_{MX}), high-speed on-chip oscillator clock (f_{IH}), or PLL clock (f_{PLL})
Clock state after transition	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})

Change to analog block standby state using one of the following:

(a) Clear the ADCEN bit in the PER0 register to 0.

(b) Set the ADSLP bit in the ADHVREFCNT register to 1 while the ADCEN bit is set to 1.

12.7.9 Notes on Canceling STOP Mode

After release from STOP mode, set the analog block to operation mode (ADSLP = 0) after the oscillation stabilization time has elapsed, and wait for 1 μs or more before starting A/D conversion. For details on release from STOP mode, see **24.3.2 STOP mode, (2) Release from STOP mode.**

12.7.10 Error in Absolute Accuracy When Disconnection Detection Assistance Is in Use

Using disconnection detection assistance leads to an error in absolute accuracy of the A/D converter. This is because an error voltage is input to the analog input pins due to the resistive voltage division between the pull-up or pull-down resistor (Rp) and the resistance of the signal source (Rs). This error in absolute accuracy is calculated from the following formula. Only use disconnection detection assistance after thorough evaluation.

Maximum error in absolute accuracy (LSB) = 4095 × Rs/Rp

12.7.11 Voltage Range of Analog Power Supply Pins

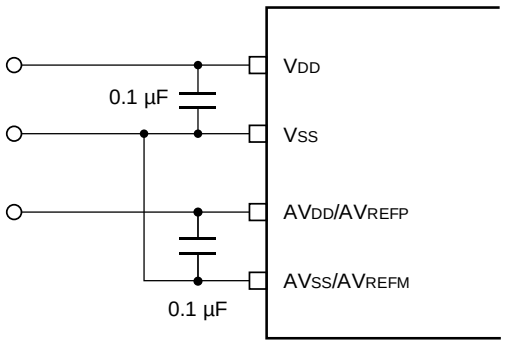
When using the 12-bit A/D converter, observe the following conditions.

- Analog input voltage range
  Voltage applied to analog input pins ANIn (n = 0 to 28) is $AV_{SS}/AV_{REFM} \leq V_{AIN} \leq AV_{DD}/AV_{REFP}$ according to the setting of reference voltage.

- Relationship between power supply pin pairs (AV_{DD}/AV_{REFP} - AV_{SS}/AV_{REFM})
  Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply. When performing A/D conversion of analog input pin ANIn (n = 0 to 28), a capacitor should be connected between each pair of power supply pins to create a closed loop with the shortest route possible as shown in **Figure 12-43**, and connection should be made so that the following conditions are satisfied at the supply side.
  $AV_{DD}/AV_{REFP} \leq V_{DD}$ and $AV_{SS}/AV_{REFM} = V_{SS}$

  When the 12-bit A/D converter is not used, the following conditions should be satisfied.
  $AV_{DD}/AV_{REFP} = V_{DD}$ and $AV_{SS}/AV_{REFM} = V_{SS}$

Figure 12-43. Power Supply Pin Connection Example



12.7.12 Notes on Board Design

The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (ANI0 to ANI28), analog power supply/reference power supply pin (AV_{DD}/AV_{REFP}) should be separated from digital circuits using the analog ground/reference ground pin (AV_{SS}/AV_{REFM}).

12.7.13 Notes on Noise Prevention

(i) To prevent the analog input pins (ANI0 to ANI28) from being destroyed by abnormal voltage such as excessive surge, a capacitor should be inserted between V_{DD} and V_{SS} and between AV_{DD}/AV_{REFP} and AV_{SS}/AV_{REFM}, and a protection circuit should be connected to protect the analog input pins (ANI0 to ANI28) as shown in **Figure 12-44**.

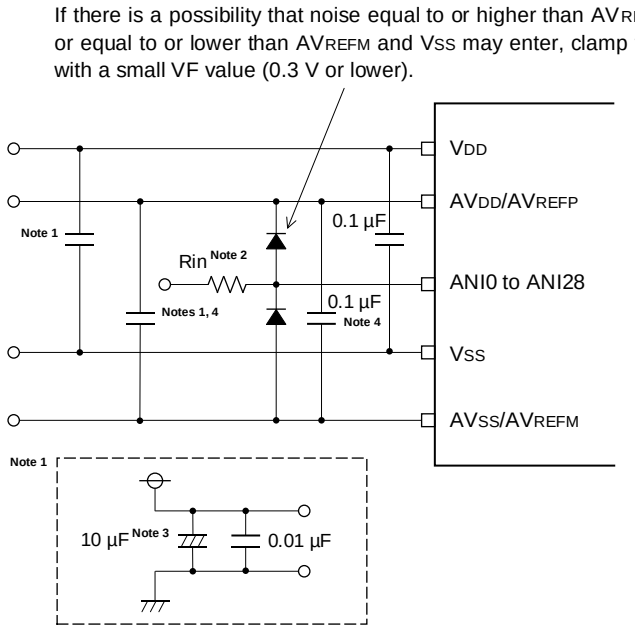
(ii) The A/D conversion result may vary due to fluctuations of the power supply voltage which is caused by changing of a port output current consumption change by erasing or writing operation of the flash memory, or the effects of noise. Also, if there is noise on the analog input pin (ANIx), power supply voltage/reference voltage input pin (AV_{DD}/AV_{REFP}, AV_{SS}/AV_{REFM}), the conversion result may vary.
Apply software processing to avoid negative influence by variations of the results of A/D conversion to the system. Examples of software processing are described below.
- Use averaged values from several rounds of A/D conversion.
- Execute A/D conversion for several time and rule out extreme results.

(iii) The accuracy is improved if the HALT mode is set immediately after the start of conversion.

(iv) When selecting any channel (ANI0-ANI29) to start A/D conversion, do not change the levels output on P70 to P74, P80 to P87, P90 to P97, P100 to P105, P120, and P125 during conversion, since doing so may decrease the accuracy.

(v) When a pin adjacent to a pin on which A/D conversion is in progress is used as a digital I/O port pin, the result of A/D conversion may differ from the accurate value due to noise coupling.
Take care to avoid pulses which change dramatically, like digital signals, being input or output through adjacent pins during A/D conversion.

Figure 12-44. Sample Protection Circuit for Analog Inputs



Notes 1. The values indicated in the figure are reference values.

2. Rin: Signal source impedance

3. Connect a 10 µF capacitor between the AV_{REFP} and AV_{REFM} pins.
It is recommended that the capacitor be placed adjacent to the reference pins of the A/D converter.

4. Connect a capacitor between the AV_{REFP} and AV_{REFM} pins.

12.7.14 Allowable Impedance of Signal Source

The A/D converter charges an internal sampling capacitor during the sampling period. Therefore, only leakage current flows except during sampling, but current such as capacitor charging current flows during sampling. So the input impedance differs between the sampling state and other states.

The analog input pins of this MCU are designed so that the conversion accuracy is guaranteed if the impedance of the input signal source is 0.5 kΩ or less. If the impedance of the input signal source cannot be reduced to 0.5 kΩ or less, it is recommended to set a long sampling time or attach a capacitor of about 0.1 μF to the analog input pins. However, since the load of the input pins creates an equivalent low-pass filter, the tracking of analog signals with large differential coefficients may become impossible.

Insert a low-impedance buffer in cases of the conversion of high-speed analog signals or of multiple signals in scan mode and so on.

Furthermore, if conversion of the signal on a single pin is to proceed in single-scan mode, even in the case that an external capacitor with a large value is provided, switching of the analog multiplexer from input signal to input signal will affect the current.

Figure 12-45 shows an equivalent circuit of an analog input pin and an external sensor.

To perform A/D conversion accurately, charging of the internal capacitor shown in **Figure 12-45** must be completed within the specified period of time. This specified period is referred to as sampling time.

Figure 12-45. Equivalent Circuit of ANIn Pins

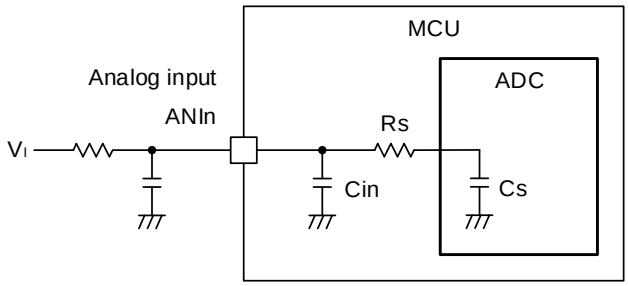


Table 12-22. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

ANIn	Cin [pF]	Rs [kΩ]	Cs [pF]
ANI0, ANI3 to ANI15	8	2.5	8
ANI1, ANI2 (without channel dedicated S&H circuit)			
ANI1, ANI2 (with channel dedicated S&H circuit)	8	10.5	3
ANI16 to ANI28	8	6.7	9

Remark The resistance and capacitance values in **Table 12-22** are not guaranteed values.

CHAPTER 13 D/A CONVERTER (RL78/F25 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

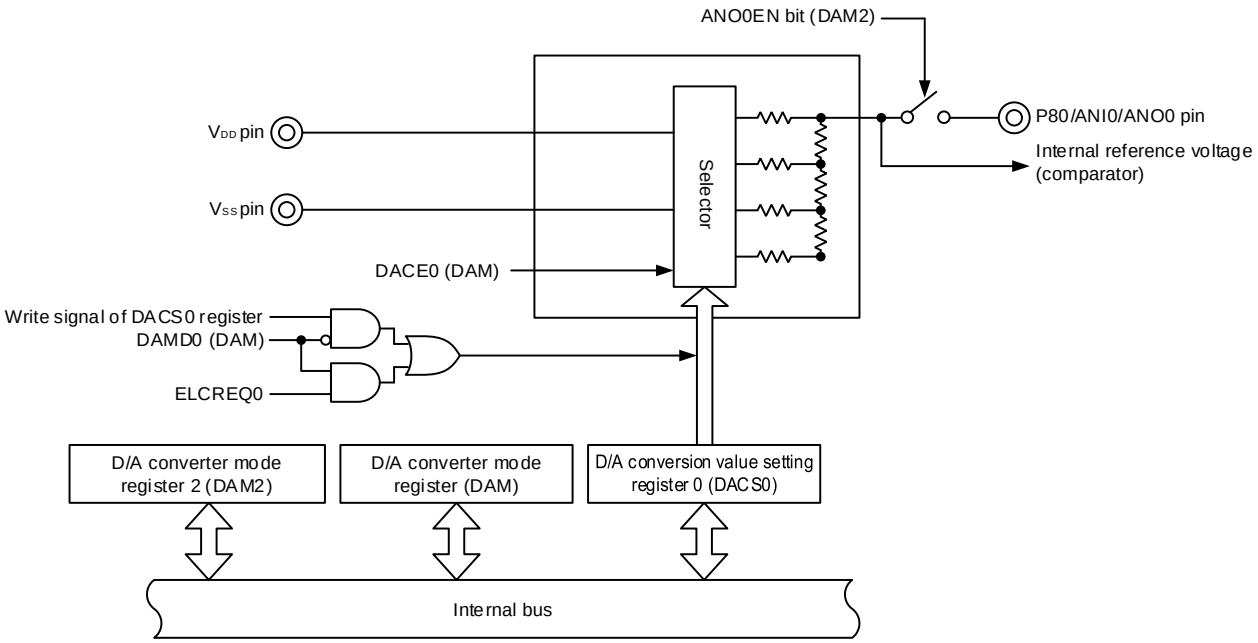
The D/A converter has the following features.

- 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

Table 13-1. D/A Converter Register Configuration

Address	Register Name	Symbol	After Reset	Access Size
F02C1H	Peripheral enable register 1	PER1	00H	1, 8
FFF34H	D/A conversion value setting register 0	DACS0	00H	8
FFF36H	D/A converter mode register	DAM	00H	1, 8
F0227H	D/A converter mode register 2	DAM2	00H	1, 8

Remark See **13.3.1 Port Mode Control Register 8 (PMC8)** and **13.3.6 Port Mode Register 8 (PM8)** for the port mode control register 8 (PMC8) and port mode register 8 (PM8).

13.3.1 Port Mode Control Register 8 (PMC8)

This register switches the P80 to P87 to digital I/O or analog I/O in 1-bit units.

When the D/A converter is used and P80/ANI0/ANO0 is used as ANO0, set PMC80 to analog output and set PM80 to input mode.

For details, see **13.3.6 Port Mode Register 8 (PM8)**.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 13-2. Format of Port Mode Control Register 8 (PMC8)

Address: F0068H After reset: FFH

Symbol	7	6	5	4	3	2	1	0
PMC8	PMC87	PMC86	PMC85	PMC84	PMC83	PMC82	PMC81	PMC80

PMC8n	P8n pin digital I/O or analog I/O selection	R/W
0	Digital I/O (alternate function other than analog I/O)	R/W
1	Analog I/O	

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 PMC8 register as digital I/O to D/A conversion operation enable by using the D/A converter mode register (DAM).

Remark n = 0 to 7

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: F02C1H After reset: 00H

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER1	DACEN Note	CTSUEN	CMPEN Note	TRD0EN	DTCEN	PWMOPEN	0	TRJ0EN

DACEN Note	Control of D/A converter input clock	R/W
0	Stops input clock supply. • SFR used by the D/A converter cannot be written. • The D/A converter is in the reset state.	R/W
1	Enables input clock supply. • SFR used by the D/A converter can be read and written.	

Note Only for RL78/F25.

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), Port mode control register 8 (PMC8), and D/A converter mode register 2 (DAM2)).

2. Be sure to clear the following bits to 0.

RL78/F22: bits 1, 5, and 7

RL78/F25: bit 1

Remark See **5.3.6 Peripheral Enable Registers 0, 1, 2 (PER0, PER1, PER2)** for a description of bits other than DACEN.

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	0	0	0	DACE0	0	0	0	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	

Caution When the D/A converter is not used, set the DACE0 bit to 0 (stops D/A conversion operation) 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/ANI0/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 P80/ANI0/ANO0 pin as an analog input / output port, set bit PM80 to 1.

If bit PM80 is set to 0, this pin cannot be used as an analog input / output 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: 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 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 P80/ANI0/ANO0 pin can be selected by using the Port mode control register 8 (PMC8), the D/A converter mode register (DAM), D/A converter mode register 2 (DAM2), the A/D Channel Select Register (ADANSA0, ADANSB0), and the PM8 register.

Table 13-2. Setting Functions of P80/ANI0/ANO0 Pin

PMC8 Register	PM8 Register	DAM Register	DAM2 Register	ADANSA0, ADANSB0 Register	Functions of P80/ANI0/ANO0 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 output
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 Port mode control register 8 (PMC8) 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.
If the internal reference voltage (D/A converter output) is selected as the comparator reference voltage, disable the comparator output (clear the COE bit of the CMPCTL register to 0) before changing the value of the DACS0 register. After that, enable the comparator output (set the COE bit to 1) after the comparator stabilization wait time (t_{CMP}: 1 μs) has elapsed.

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 PMC8 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 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 mode control register 8 (PMC8) 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 ELC 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 PMC8 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 the DACE0 bit setting is 1 to 0 and then 1, 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 Port mode control register 8 (PMC8). When the P8 register is read while the ports are set to analog pins by using the PMC8 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/F25 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, **Figure 14-1** shows the comparator block diagram, and **Figure 14-2** shows the pin configuration of the comparator.

Table 14-1. Comparator Specifications

Item	Specification
Number of channels	1 ch (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) • Input voltage from the external reference voltage input pin (IVREF0)
Comparator output	• Comparison result • Generation of ELC/PWMOPA event output • Monitor output from register
Interrupt request signal	• An interrupt request is generated on detecting a valid edge of comparison result. • Rising edge, falling edge, or both edges can be selected.
Digital filter function	• One of three sampling frequencies can be selected. • Not using the filter function can be selected.

Figure 14-1. Comparator Block Diagram

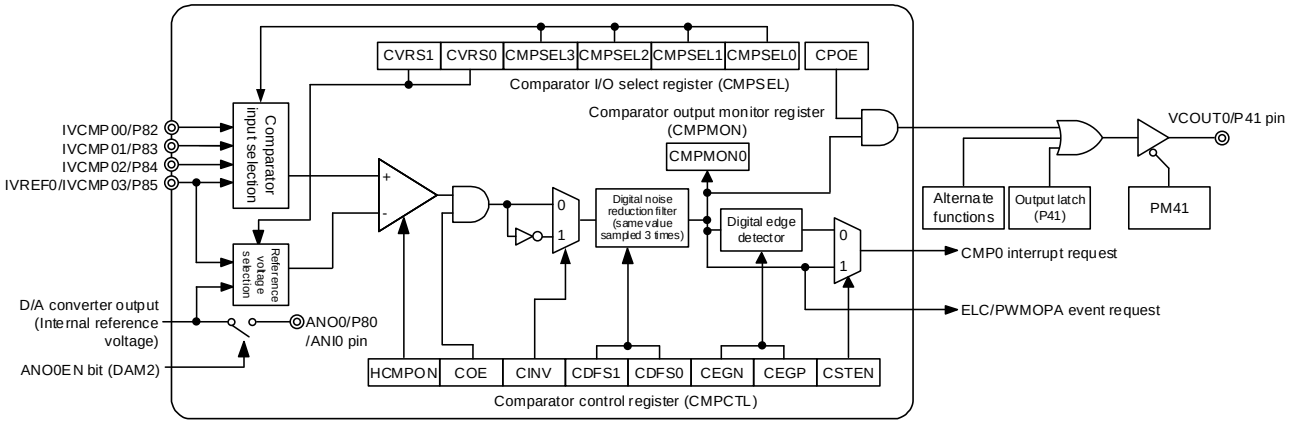


Table 14-2. Comparator Pin Configuration

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

Address	Register Name	Symbol	After Reset	Access Size
F02C1H	Peripheral enable register 1	PER1	00H	1, 8
F0227H	D/A converter mode register 2	DAM2	00H	1, 8
F02A0H	Comparator control register	CMPCTL	00H	1, 8
F02A1H	Comparator I/O select register	CMPSEL	00H	1, 8
F02A2H	Comparator output monitor register	CMPMON	00H	1, 8

Remark See **14.2.5 Port Mode Control Register 8 (PMC8)**, **14.2.7 Port Mode Register 4 (PM4)**, and **14.2.8 Port Mode Register 8 (PM8)** for the port mode control register 8 (PMC8) and port mode register 4, 8 (PM4, PM8).

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: F02C1H After reset: 00H

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER1	DACEN Note 1	CTSUEN	CMPEN Note 1	TRD0EN Note 2	DTCEN	PWMOPEN	0	TRJ0EN

CMPEN Note 1	Control of comparator input clock	R/W
0	Stops input clock supply. • SFR used by the comparator cannot be written. • The comparator is in the reset state.	R/W
1	Enables input clock supply. • SFR used by the comparator can be read and written.	

Notes 1. Only for the RL78/F25.

2. When FRQSEL4 = 1 in the user option byte (000C2H/040C2H), 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.

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/F22: bits 1, 5, and 7

RL78/F25: bit 1

Remark See **5.3.6 Peripheral Enable Registers 0, 1, 2 (PER0, PER1, PER2)** for a description of bits other than CMPEN.

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: F02A0H 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, 7	R/W
0	0	Noise filter not used	R/W
0	1	Noise filter sampling time is $2^3/f_{CLK}$.	
1	0	Noise filter sampling time is $2^4/f_{CLK}$.	
1	1	Noise filter sampling time is $2^5/f_{CLK}$.	

CEGN	CEGP	Selection of valid edge of INTCMP interrupt signal	R/W
0	0	No edge selection	R/W
0	1	Rising edge selection	
1	0	Falling 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 of 1 μs 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 DTCEN57 bit in the DTCEN5 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 1 and also set bits CDFS1 and CDFS0 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).
7. When using a comparator, use it under one or both of the following conditions:
   - Use the noise filter (CDFS[1:0] bits are other than 00B).
   - The maximum slope of the voltage input to IVCMP0n (n: 0 to 3) is ± 3 V/µs or less.

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: F02A1H 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 Note 5	
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 Note 5	
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/F25 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.

5. IVREF0 pin and IVCMP03 pin cannot be used at the same time.

14.2.4 Comparator Output Monitor Register (CMPMON)

This register is used to monitor the comparator output.

The CMPMON register can only be read 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) • Comparator input voltage (IVCMP0n) < reference voltage • Comparator operation disabled (HCMPON = 0) • Comparator output is disabled (COE = 0) When CINV = 1 (converter output is inverted) • Comparator input voltage (IVCMP0n) > reference voltage	R
1	When CINV = 0 (comparator output is not inverted) • Comparator input voltage (IVCMP0n) > reference voltage When CINV = 1 (comparator output is inverted) • Comparator input voltage (IVCMP0n) < reference voltage • Comparator operation disabled (HCMPON = 0) • Comparator output is disabled (COE = 0)	

Caution When comparator operation is enabled (HCMPON = COE = 1) but the noise filter is not in use (CDFS1 and CDFS0 = 00B), use when the maximum slope of the IVCMP0n (n: 0 to 3) input voltage is ±3 V/μs or less. After that, write the software so that the CMPMON0 bit is read twice and the values are only used after the two consecutive values match.

14.2.5 Port Mode Control Register 8 (PMC8)

This register set the P80 to P87 digital I/O or analog input in 1-bit units.

When the comparator is in use, set the pins selected from among P82/ANI2/IVCMP00, P83/ANI3/IVCMP01, P84/ANI4/IVCMP02, and P85/ANI5/IVCMP03/IVREF0 to analog input by using the PMC8 and PM8 registers.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 14-6. Format of Port Mode Control Register 8 (PMC8)

Address: F0068H After reset: FFH

Symbol	7	6	5	4	3	2	1	0
PMC8	PMC87	PMC86	PMC85	PMC84	PMC83	PMC82	PMC81	PMC80

PMC8n	P8n pin digital I/O or analog input selection (n = 0 to 7)	R/W
0	Digital I/O (alternate function other than analog input)	R/W
1	Analog input	

Caution Set the pins to be used for the comparator (P82/ANI2/IVCMP00, P83/ANI3/IVCMP01, P84/ANI4/IVCMP02, and P85/ANI5/IVCMP03/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/ANI0/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 2 (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 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 FFH.

Figure 14-8. Format of Port Mode Register 4 (PM4)

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	Output mode (output buffer on)	R/W
1	Input mode (output buffer off)	

14.2.8 Port Mode Register 8 (PM8)

When using the P82/ANI2/IVCMP00, P83/ANI3/IVCMP01, P84/ANI4/IVCMP02, or P85/ANI5/IVCMP03/IVREF0 Note pin for an analog input port of the comparator, set the PM82 to PM85 bits to 1 corresponding to the port to be used.

If the PM82 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 this register to FFH.

Note IVREF0 pin and IVCMP03 pin cannot be used at the same time.

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	Output mode (output buffer on)	R/W
1	Input mode (output buffer off)	

The P82/ANI2/IVCMP00, P83/ANI3/IVCMP01, P84/ANI4/IVCMP02, and P85/ANI5/IVCMP03/IVREF0 pins are as shown below depending on the settings of the Port Mode Control Register 8 (PMC8), A/D Channel Select Registers (ADANSA0, ADANSB0), and the Port Mode Register 8 (PM8).

Table 14-4. Setting Functions of P82/ANI2/IVCMP00 to P85/ANI5/IVCMP03/IVREF0 Pins

PMC8 Register	PM8 Register	ADANSA0, ADANSB0 Register	P82/ANI2/IVCMP00 to P85/ANI5/IVCMP03/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.	

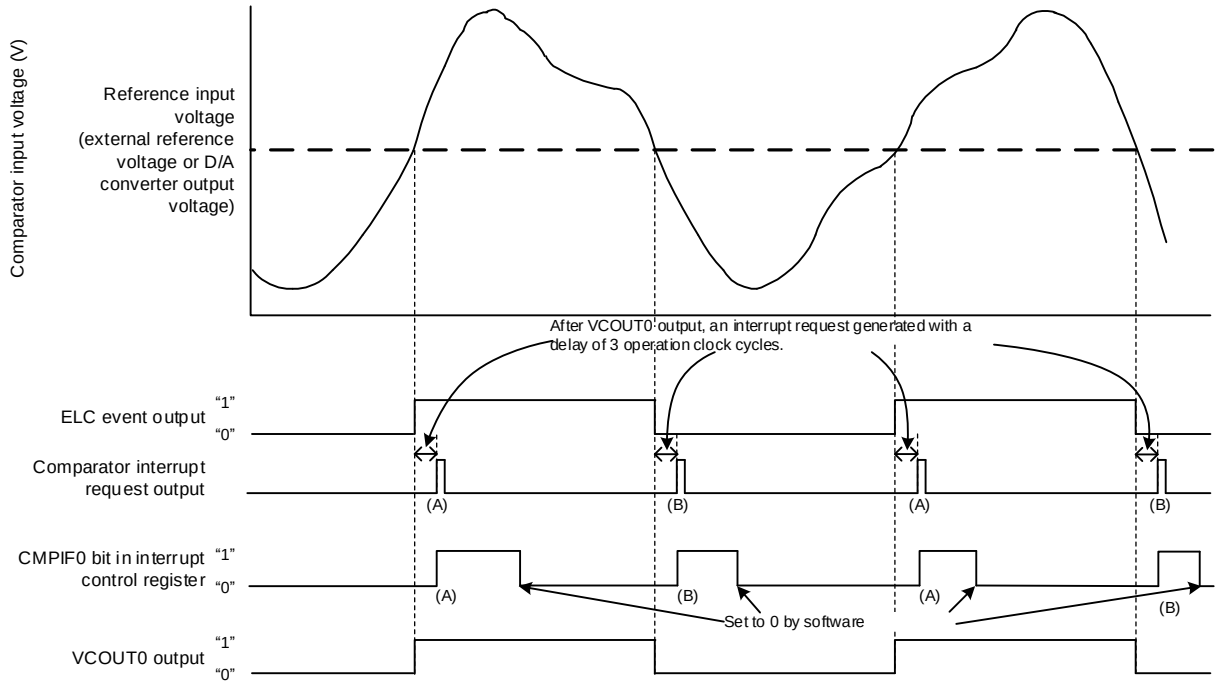
Reset signal generation sets all the P82/ANI2/IVCMP00 to P85/ANI5/IVCMP03/IVREF0 pins to analog input.

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, 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.

To change the level of the comparator reference voltage (IVREF0 input or D/A converter output), disable the comparator output (set the COE bit of the CMPCTL register to 0), change the reference voltage input level, and then enable the comparator output (set the COE bit to 1) after the stabilization wait time (t_{CMP}: 1 µs) has elapsed. Similarly, when switching the analog input pin of the comparator, disable the comparator output before switching the analog input voltage pin and enable the comparator output after the stabilization wait time (t_{WAIT}: 300 ns) has elapsed. When not using a noise filter, the maximum slope of the voltage input to IVCMP0n (n: 0 to 3) must be ±3 V/µs or less. The voltage input to IVREF0 is changed while the comparator output is disabled, so there is no slope requirement.

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

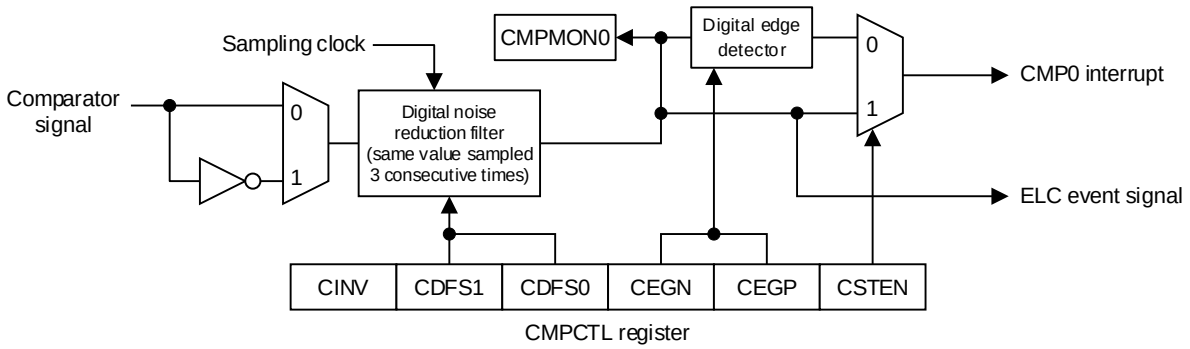
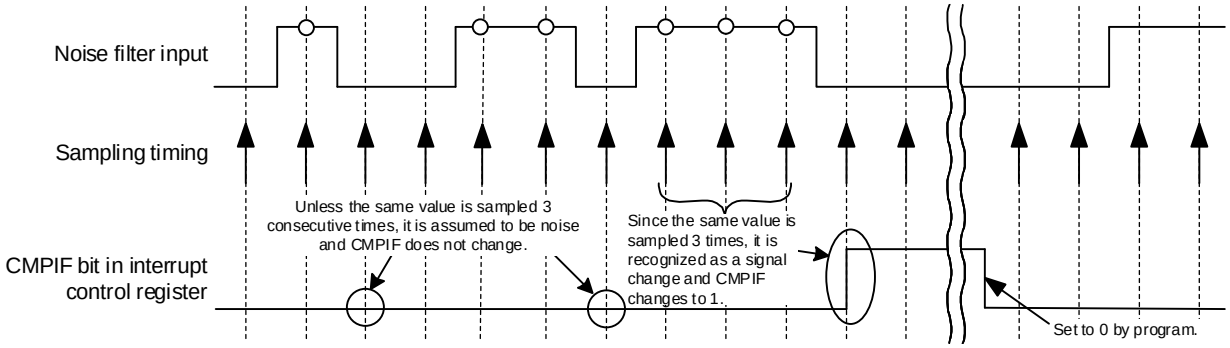


Figure 14-12. Noise Filter and Interrupt Operation Example



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/PWMOPA Event Output

An ELC/PWMOPA 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. Then, set an event input of PWMOPA with cutoff source selection (IN_SEL1 and IN_SEL0 bits) in the OPCTL0 register.

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:

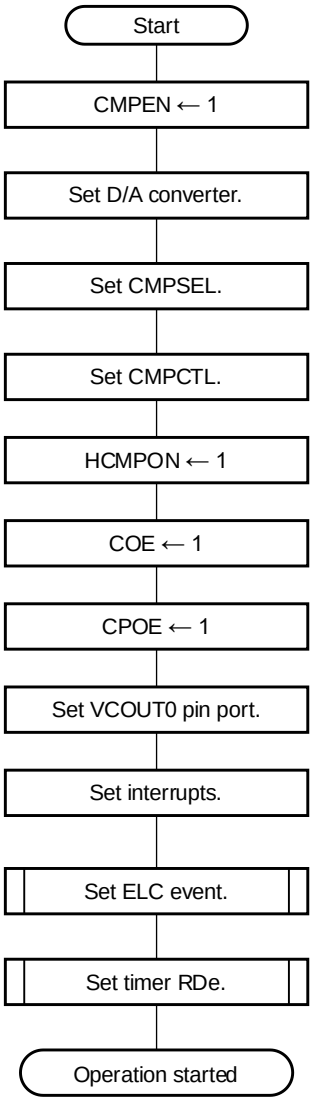
<1> Set the HCMPON bit in the CMPCTL register to 0 (stop the comparator input).
<2> Set the CMPEN bit in the PER1 register to 0.
<3> Set the interrupt flag (CMPIF0 bit in the IF0L register) to 0 (clear any unnecessary interrupt before stopping the comparator).

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 **Figure 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 RDe Operation Triggered by Internal Reference Voltage (D/A Converter Output), INTCMP0 Interrupt, or ELC Event)



Supply clock to comparator (set peripheral enable register 1 (CMPEN bit in the PER1 register).

Set D/A converter to select the internal reference voltage. Note 1
(See **CHAPTER 13 D/A CONVERTER (RL78/F25 Only)**).

Set comparator input (set CVRS1,CVRS0, CMPSEL3 to CMPSEL0, CINV, and CSTEN) (after changing the settings of CVRS1,CVRS0, and CMPSEL3 to CMPSEL0, switching stabilization wait time (300 ns) is required.)

Set comparator control (set CDFS1, CDFS0, CEGP, and CEGN).

Enable comparator operation (after enabling, comparator operation stabilization wait time (1 µs) is required.)

Enable comparator output.

Enable comparator pin output. Note 2

Set P41/VCOUT0 pin for comparator output. Note 2
(set the P41 register to 0 (output 0) and set the PM41 register to 0 (output mode))

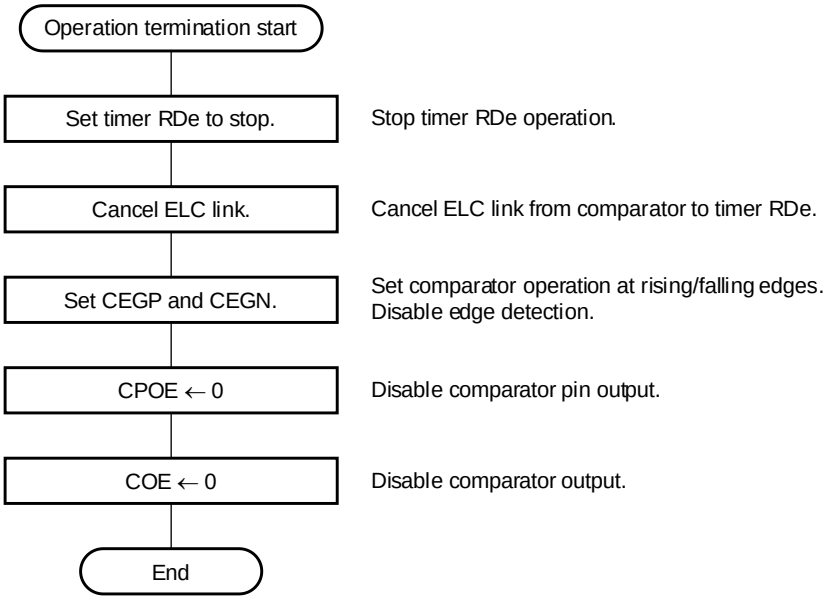
Set interrupt priority level and interrupt mask, and initialize interrupt flags. Note 3
(set the CMPPR00 bit in the PR00L register, CMPMK0 bit in the MK0L register, and CMPIF0 bit in the IF0L register.)

Set ELC event link to timer RDe
(set ELSEL3 to ELSEL0 bits in the ELSELR19 register) Note 4.
(See **CHAPTER 21 EVENT LINK CONTROLLER (ELC) (RL78/F25 Only)**).

Set timer RDe operation, and enable operation. Note 5
(See **CHAPTER 8 TIMER RDe**.)

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 RDe by the ELC event is not used.

Figure 14-14. Comparator Operation Termination Flowchart (when Using the Timer RDe Operation by the ELC Event)



CHAPTER 15 CAPACITIVE SENSING UNIT (CTSU2SLa)

The number of measurement channels of the capacitive sensing unit depends on the product.

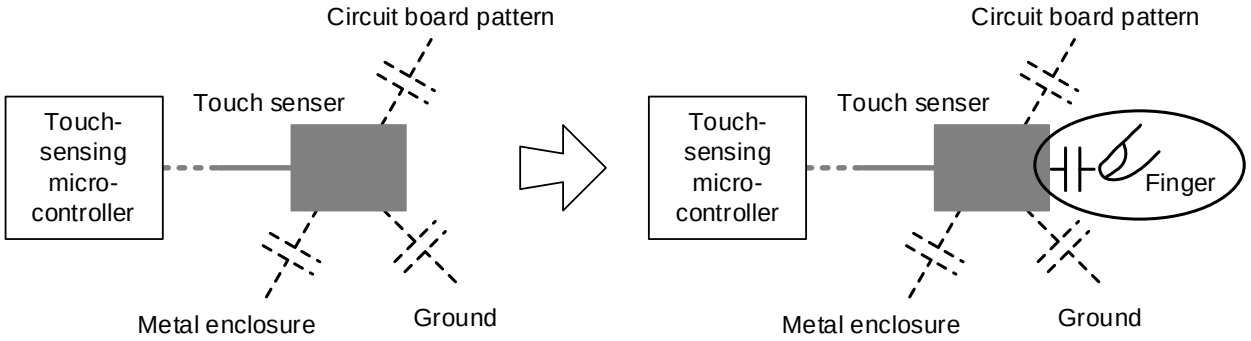
Products	RL78/F25				RL78/F22		
	100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
Number of Measurement Channels	31	31	23	15	12	10	7
TS Pins	TS0 to TS30	TS0 to TS30	TS0 to TS3, TS8 to TS26	TS0 to TS3, TS12 to TS18, TS23 to TS26	TS0 to TS11	TS0 to TS3, TS6 to TS11	TS2, TS3, TS6 to TS10

Remark The allocation of TSm pins to ports differs between the RL78/F25 and RL78/F22 products. For details, see **Table 4-29**, **Remarks 4** in **4.5 Setting of Port Mode Register and Output Latch When Using Alternate Function**.

The Capacitive Sensing Unit (CTSU: Capacitive Touch Sensing Unit) measures the electrostatic capacitance of the capacitive sensor. Changes in the electrostatic capacitance are determined by software, which enables the CTSU to detect whether a finger is in contact with the capacitive sensor. The electrode surface of the capacitive sensor is usually enclosed with an electrical conductor so that a finger does not come into direct contact with the electrode.

As shown in **Figure 15-1**, electrostatic capacitance (parasitic capacitance) exists between the electrode and the surrounding conductors. Because the human body is an electrical conductor, when a finger is placed close to the electrode, the electrostatic capacitance value increases.

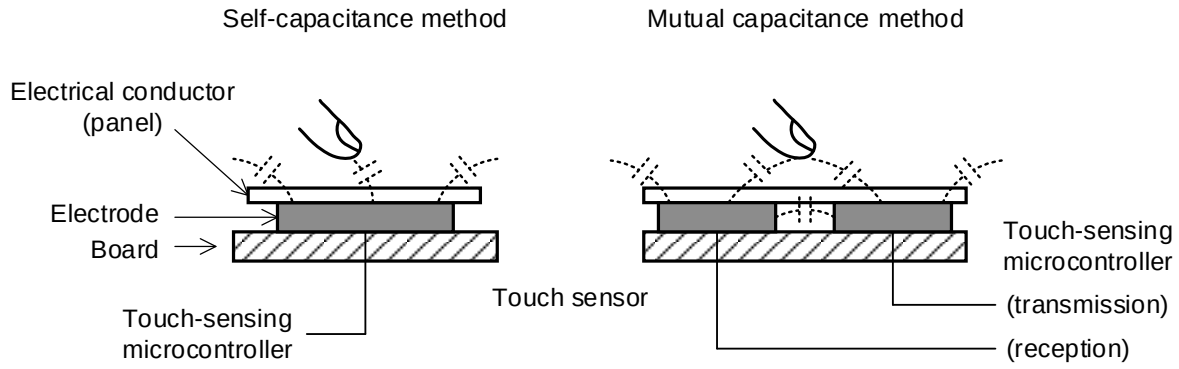
Figure 15-1. Increased Electrostatic Capacitance Because of the Presence of a Finger



Electrostatic capacitance is detected by the self-capacitance and mutual capacitance methods.

In the self-capacitance method, the CTSU detects electrostatic capacitance generated between a finger and a single electrode. In the mutual capacitance method, two electrodes are used; one electrode is used as a transmit electrode and the other electrode is used as a receive electrode. The CTSU detects a change in the electrostatic capacitance generated between these electrodes when a finger is placed close to them.

Figure 15-2. Self-capacitance Method and Mutual Capacitance Method



Electrostatic capacitance is measured by counting clock signal cycles whose frequency changes according to the amount of charged or discharged current for a specified period.

For details of the principles of measurement operation, refer to the following application notes.

- **RL78 Family Capacitive Touch Sensing Unit (CTSU2L) Operation Explanation (R01AN5744)**

Also, see the following application note on how to develop touch applications.

- **RL78 Family Using QE and SIS to Develop Capacitive Touch Applications (R01AN5512)**

15.1 Overview

15.1.1 Overview of CTSU Functions

Table 15-1 lists the CTSU functions and **Figure 15-3** shows a block diagram of the CTSU. **Figure 15-4** shows the configuration of the sensor drive pulse generator.

Table 15-1. CTSU Functions

Item		Configuration
Module type name		CTSU2SLa
Operating voltage condition		V_{DD} = 2.7 V to 5.5 V
Operating clock		2 MHz ≤ f_{CLK} ≤ 40 MHz (f_{CLK}, f_{CLK}/2, f_{CLK}/4, or f_{CLK}/8)
Pins	Electrostatic capacitance measurement pins	TSm (m = 0 to 30), up to 31 channels
	External capacitor connecting for measurement power supply	TSCAP, Recommended to connect 10 nF (0.01 µF) capacitor
Measurement mode	Self-capacitance measurement mode	Electrostatic capacitance is measured from the charged current that flows toward the electrode used in the self-capacitance method.
	Mutual capacitance measurement mode	Electrostatic capacitance is measured from the charged current that flows toward the capacitance generated between the transmit and receive electrodes used in the mutual capacitance method.
	Current measurement mode	Direct measurement of current flowing through a pin
Calibration mode		Characteristic correction of the current control oscillator for measurement
Noise prevention		• Spread spectrum function for sensor drive pulses • Random phase shift function for sensor drive pulses • Majority decision by multiple frequency measurement
Adjustment for each pin		• Offset current adjustment function • Specifying the sensor drive pulse frequency • Measurement time specification
Measurement start conditions		• Software trigger • External trigger (Selectable from 4 trigger factors by CTSUTRG register)
Low-power function		SNOOZE function supported
Automatic operations		• Automatic correction • Automatic judgment
Request output	Data transfer request	• Channel measurement setting write request • Measurement result transfer request
	Interrupt request	• Channel measurement setting write request (INTCTSUWR) • Measurement result transfer request (INTCTSURD) • Measurement end interrupt request (INTCTSUFN)
Transmission power switching		The power for transmission can be switched.

As shown in **Figure 15-3**, the CTSU consists of a state controller, trigger control block, clock control block, channel control block, port control block, sensor drive pulse generator, measurement block, interrupt control block, SNOOZE control block, automatic correction block, automatic judgment block, and control registers.

Figure 15-3. CTSU Block Diagram

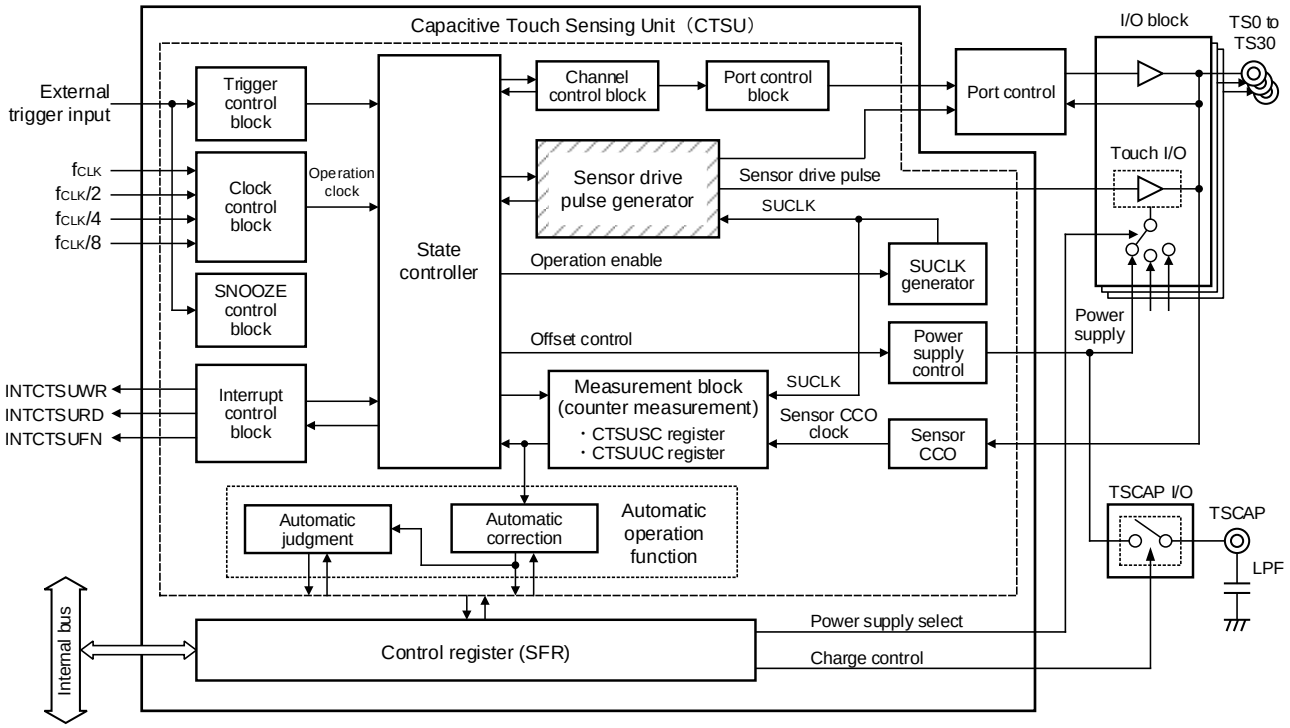


Figure 15-4. Generation of Sensor Drive Pulse

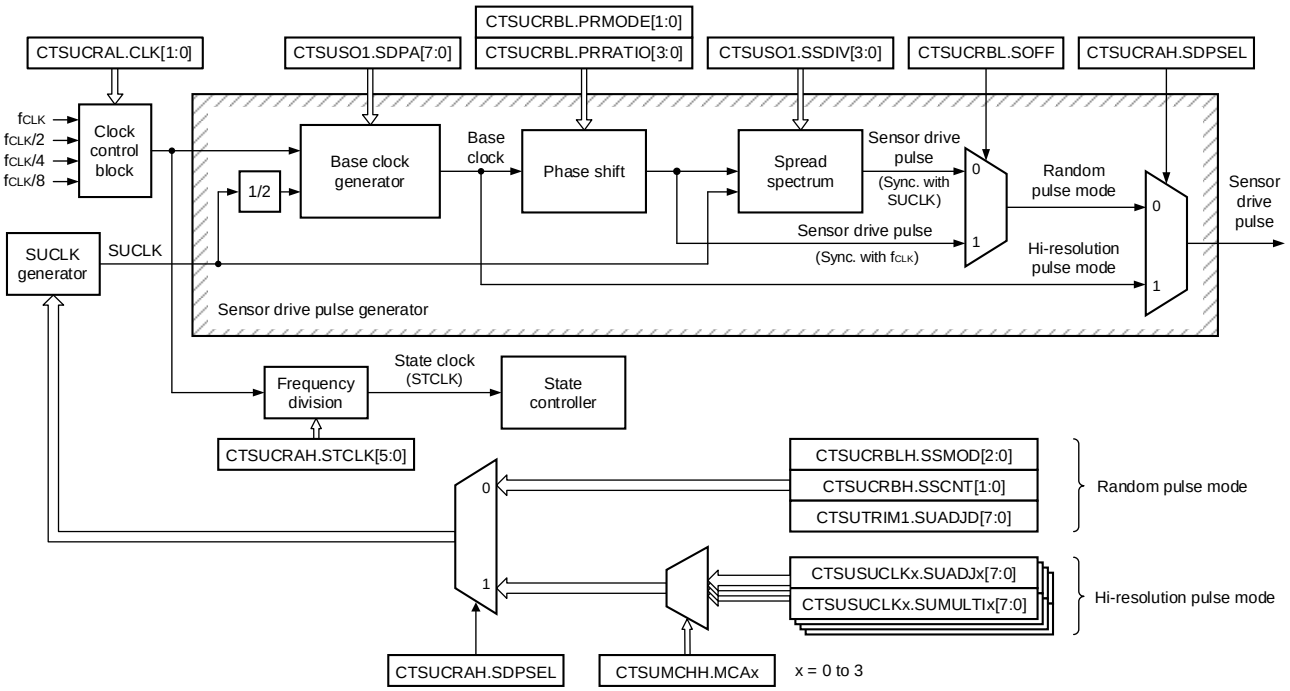


Table 15-2. Input/Output Pins for CTSU

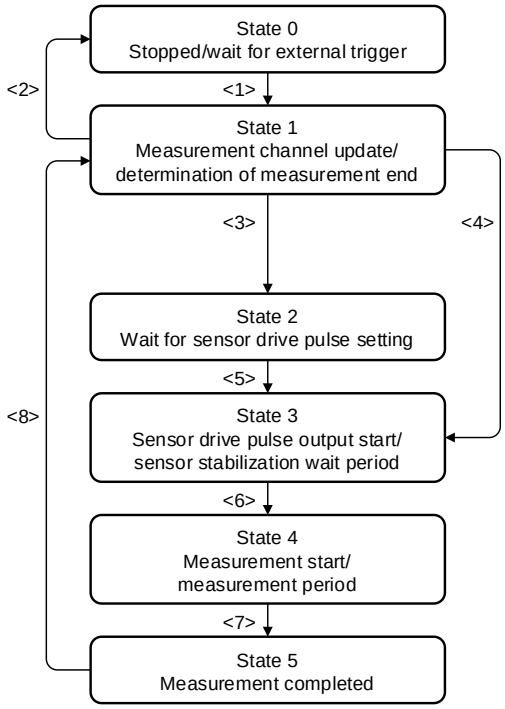
Pin Name	Input/Output	Function
TSm (m = 0 to 30)	I/O	Electrostatic capacitance measurement pins (touch pins)
TSCAP	–	Connection pin for secondary power supply capacitor for measurement

15.1.2 Measurement Status

The CTSU has a built-in state controller, and the measurement status is common to all measurement methods. The state counter indicates the current measurement status and can be read with the CTSUSRL.STC[2:0] bits.

Figure 15-5 shows the state transition diagram of the measurement status.

Figure 15-5. Measurement Status Transition



<1> Conditions for transition to State 1
- CTSUCRAL.CAP = 0: CTSUCRAL.STRT = 1
- CTSUCRAL.CAP = 1: CTSUCRAL.STRT = 1 and the external trigger detected. (When CTSUCRAL.SNZ = 1, after 80 cycles of the base clock.)

<2> Conditions for transition to State 0
- When there is no channel to be measured next and CTSUSRL.DTSR = 0.

<3> Conditions for transition to State 2
- When update of the measurement channel is completed and CTSUSRL.DTSR = 0.

<4> Conditions for transition to State 3
- When CTSUSRL.PS bit = 1 (second measurement) and CTSUSRL.DTSR = 0 in mutual capacitance method (CTSUCRAL.MD1 = 1)

Note When CTSUSRL.DTSR = 1, CTSU waits until the previous measurement result is read.

<5> Conditions for transition to State 3
- Write to the CTSUSO1 register

<6> Conditions for transition to State 4
- When the sensor stability time has elapsed after the sensor drive pulse has been supplied.

<7> Conditions for transition to State 5
- When the measurement period has elapsed after measurement is started.

<8> Conditions for transition to State 1
- After two cycles of the operating clock have elapsed.

The state counter transitions to State 0 when the measurement of the specified channel is completed.

The CTSUCRAL.STRT bit is set to 0 by hardware when a software trigger is used. When an external trigger is used, the STRT bit remains 1 and the CTSU waits for the next trigger.

When operation is forced to stop (simultaneous writing of 0 to the CTSUCRAL.STRT bit and 1 to the CTSUCRAL.INIT bit) during measurement or the trigger wait state, the state transitions to State 0 and measurement stops.

If the channel to be measured is not set in the CTSUMCHL, H, CTSUCHACAL, H, and CTSUCHTRCAL, H registers, the INTCTSUFN interrupt occurs immediately after a transition to State 1, and then the state transitions to State 0.

Cases where there is no channel to measure are as follows:

- When the channel to be measured is not specified in the CTSUCHACAL, H register.
- When the channel specified to the CTSUMCHL, H register in single-scan mode is not the target channel for measurement in the CTSUCHACAL, H register.
- In the case of the self-capacitance method, when there is no receive channel to be measured in the combination of the CTSUCHACAL, H register and the CTSUCHTRCAL, H register.
- In the case of mutual capacitance method, when there is no transmit or receive channel to be measured in the combination of the CTSUCHACAL, H register and the CTSUCHTRCAL, H register.

15.2 Registers to Control the Capacitive Sensing Unit

The following registers are used to control the capacitive sensing unit.

Table 15-3. Capacitive Sensing Unit Register Configuration (1/2)

Address	Register Name	Symbol		After Reset	Access Size
F02C1H	Peripheral enable register 1	PER1		00H	1, 8
F0180H	CTSU control register AL	CTSUCR0	CTSUCRAL	0000H	8, 16
F0181H		CTSUCR1			
F0182H	CTSU control register AH	CTSUCR2	CTSUCRAH	0000H	8, 16
F0183H		CTSUCR3			
F0184H	CTSU control register BL	CTSUSDPRS	CTSUCRBL	0000H	8, 16
F0185H		CTSUSST			
F0186H	CTSU control register BH	–	CTSUCRBH	0000H	8, 16
F0187H		CTSUDCLKC			
F0188H	CTSU measurement channel register L	CTSUMCH0	CTSUMCHL	3F3FH Note	8, 16
F0189H		CTSUMCH1			
F018AH	CTSU measurement channel register H	CTSUMFAF	CTSUMCHH	0000H	8, 16
F018BH		–			
F018CH	CTSU channel enable control register AL	CTSUCHAC0	CTSUCHACAL	0000H	8, 16
F018DH		CTSUCHAC1			
F018EH	CTSU channel enable control register AH	CTSUCHAC2	CTSUCHACAH	0000H	8, 16
F018FH		CTSUCHAC3			
F0194H	CTSU channel transmit/receive control register AL	CTSUCHTRC0	CTSUCHTRCAL	0000H	8, 16
F0195H		CTSUCHTRC1			
F0196H	CTSU channel transmit/receive control register AH	CTSUCHTRC2	CTSUCHTRCAH	0000H	8, 16
F0197H		CTSUCHTRC3			
F019CH	CTSU status register	CTSUST1	CTSUSRL	0000H	8, 16
F019DH		CTSUST			
F01A0H	CTSU sensor offset register 0	–	CTSUSO0	0000H	16
F01A1H		–			
F01A2H	CTSU sensor offset register 1	–	CTSUSO1	0000H	16
F01A3H		–			
F01A4H	CTSU sensor counter register L	–	CTSUSC	0000H	16
F01A5H		–			
F01A6H	CTSU sensor counter register H	–	CTSUUC	0000H	16
F01A7H		–			
F01A8H	CTSU calibration register L	–	CTSUDBGR0	0000H	16
F01A9H		–			
F01AAH	CTSU calibration register H	–	CTSUDBGR1	0000H	16
F01ABH		–			
F01ACH	CTSU sensor unit clock control register AL	–	CTSUSUCLK0	0000H	16
F01ADH		–			
F01AEH	CTSU sensor unit clock control register AH	–	CTSUSUCLK1	0000H	16
F01AFH		–			
F01B0H	CTSU sensor unit clock control register BL	–	CTSUSUCLK2	0000H	16
F01B1H		–			
F01B2H	CTSU sensor unit clock control register BH	–	CTSUSUCLK3	0000H	16
F01B3H		–			

Note The reset value depends on the product.
RL78/F25 product: 3F3FH
RL78/F22 product: 0F0FH

Table 15-3. Capacitive Sensing Unit Register Configuration (2/2)

Address	Register Name	Symbol		After Reset	Access Size
F01C0H	CTSU option setting register L	CTSUOPTAC	CTSUOPTL	0000H	8, 16
F01C1H		CTSUOPTAJ			
F01C2H	CTSU option setting register H	CTSUOPTACSB	CTSUOPTH	0000H	8, 16
F01C3H		–			
F01C4H	CTSU sensor counter automatic correction table access register L	–	CTSUSCTACTL	0000H	16
F01C5H		–			
F01C6H	CTSU sensor counter automatic correction table access register H	–	CTSUSCTACTH	0000H	16
F01C7H		–			
F01CCH	CTSU multi-clock automatic correction table register 1L	–	CTSUMACT1L	0000H	16
F01CDH		–			
F01CEH	CTSU multi-clock automatic correction table register 1H	–	CTSUMACT1H	0000H	16
F01CFH		–			
F01D0H	CTSU multi-clock automatic correction table register 2L	–	CTSUMACT2L	0000H	16
F01D1H		–			
F01D2H	CTSU multi-clock automatic correction table register 2H	–	CTSUMACT2H	0000H	16
F01D3H		–			
F01D4H	CTSU multi-clock automatic correction table register 3L	–	CTSUMACT3L	0000H	16
F01D5H		–			
F01D6H	CTSU multi-clock automatic correction table register 3H	–	CTSUMACT3H	0000H	16
F01D7H		–			
F01D8H	CTSU automatic judgment control register L	CTSUAJCR0	CTSUAJCRL	0000H	8, 16
F01D9H		CTSUAJCR1			
F01DAH	CTSU automatic judgment control register H	CTSUAJCR2	CTSUAJCRH	0001H	8, 16
F01DBH		CTSUAJCR3			
F01DCH	CTSU threshold register L	–	CTSUAJTHRL	0000H	16
F01DDH		–			
F01DEH	CTSU threshold register H	–	CTSUAJTHRH	0000H	16
F01DFH		–			
F01E0H	CTSU moving average result register L	–	CTSUAJMMARL	0000H	16
F01E1H		–			
F01E2H	CTSU moving average result register H	–	CTSUAJMMARH	0000H	16
F01E3H		–			
F01E4H	CTSU baseline average calculation register L	–	CTSUAJBLACL	0000H	16
F01E5H		–			
F01E6H	CTSU baseline average calculation register H	–	CTSUAJBLACH	0000H	16
F01E7H		–			
F01E8H	CTSU baseline average result register L	–	CTSUAJBLARL	0000H	16
F01E9H		–			
F01EAH	CTSU baseline average result register H	–	CTSUAJBLARH	0000H	16
F01EBH		–			
F01ECH	CTSU automatic judgment result register	CTSUAJRR0	CTSUAJRRL	0000H	8, 16
F01EDH		CTSUAJRR1			
F01F8H	CTSU external trigger select register	CTSUTRG		00H	8
F00B8H	CTSU Trimming Register 0	RTRIM	CTSUTRIM0	–	8, 16
F00B9H		DACTRIM			
F00BAH	CTSU Trimming Register 1	SUADJD	CTSUTRIM1	–	8, 16
F00BBH		TRESULT4			
F00BCH	CTSU Trimming Register 2	TRESULT0	CTSUTRIM2	–	8, 16
F00BDH		TRESULT1			
F00BEH	CTSU Trimming Register 3	TRESULT2	CTSUTRIM3	–	8, 16
F00BFH		TRESULT3			

15.2.1 Peripheral Enable Registers 1 (PER1)

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.

When CTSU is to be used, be sure to set bit 6 (CTSUEN) to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Writing to the PER1 register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation clears these registers to 00H.

Figure 15-6. Format of Peripheral Enable Register 1 (PER1)

Address: F02C1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER1	DACEN	CTSUEN	CMPEN	TRD0EN	DTCEN	PWMOPEN	0	TRJ0EN

CTSUEN	Control of capacitive sensing unit input clock supply
0	Stops input clock supply. • SFR used by the capacitive sensing unit (CTSU) cannot be written. • The capacitive sensing unit (CTSU) is in the reset status.
1	Enables input clock supply. • SFR used by the capacitive sensing unit (CTSU) can be read and written.

15.2.2 CTSU Control Register A (CTSUCRAL, CTSUCRAH)

The CTSUCRAL and CTSUCRAH registers can be set by 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears these registers to 0000H.
Set bits other than the STRT and INIT bits while the STRT bit is 0 (measurement operation stopped).

Figure 15-7. Format of CTSU Control Register A

Address: (16-bit access) CTSUCRAH: F0182H
(8-bit access) CTSUCR2: F0182H CTSUCR3: F0183H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCRAH	DCBACK	DCMODE	STCLK[5:0]					

7	6	5	4	3	2	1	0
PCSEL	SDPSEL	POSEL[1:0]		LOAD[1:0]		ATUNE2	0

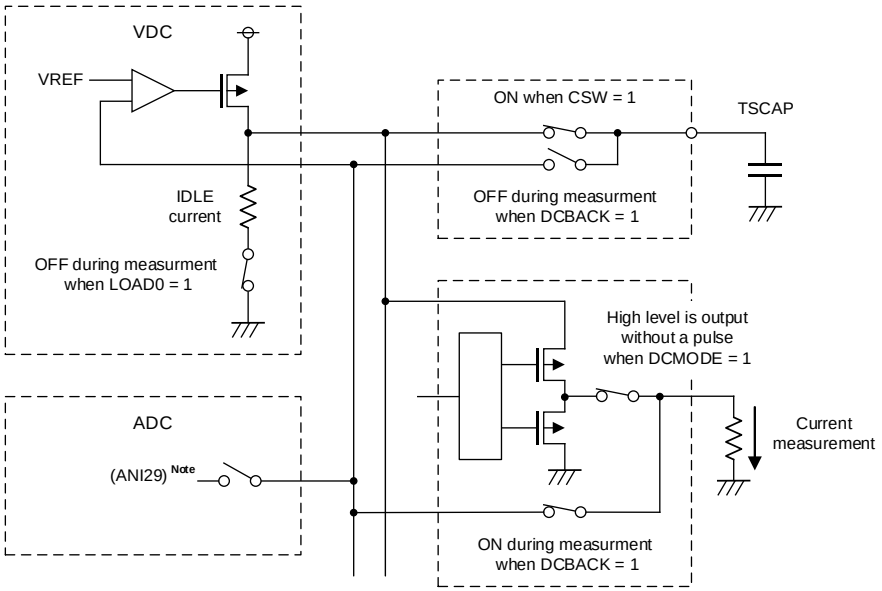
Address: (16-bit access) CTSUCRAL: F0180H
(8-bit access) CTSUCR0: F0180H CTSUCR1: F0181H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCRAL	MD1	MD0	CLK[1:0]		ATUNE1	0	CSW	PON

7	6	5	4	3	2	1	0
TXVSEL	TXVSEL2	PUMPON	INIT	0	SNZ	CAP	STRT

DCBACK	Current Measurement Feedback Select
0	Selects TSCAP-I/O
1	Selects measurement pin

This bit is valid only when DCMODE = 1 to switch the pin for feedback to measurement power.



Note No pin for ANI29 (internal signal only)

DCMODE	Current Measurement Mode Select
0	Normal mode
1	Current measurement mode

When this bit is set to 1, the current is measured without using the sensor drive pulse.
Outputs a fixed value to the measurement reception pin according to the CTSUDBGR0.IOC bit and stops the drive pulse.
IOC = 0: High-level output
IOC = 1: Low-level output

STCLK[5:0]						STCLK Select
0	0	0	0	0	0	Peripheral clock divided by 2
0	0	0	0	0	1	Peripheral clock divided by 4
0	0	0	0	1	0	Peripheral clock divided by 6
0	0	0	0	1	1	Peripheral clock divided by 8
:						:
1	1	1	1	1	0	Peripheral clock divided by 126
1	1	1	1	1	1	Peripheral clock divided by 128

These bits set the division value for f_{CLK} (after prescaler) to obtain STCLK. STCLK is related to the measurement time and the frequency adjustment cycle of SUCLK.
The division value is determined by the following equation.
Division value = (STCLK[5:0] + 1) × 2 (divided by 2 to 128)
Setting STCLK to 0.5 MHz (2 µs) is recommended.

PCSEL	Voltage Booster Clock Select
0	Sensor drive pulse
1	STCLK

This bit selects the clock for the voltage boost circuit.

SDPSEL	Sensor Drive Pulse Select
0	• Random pulse mode The sensor drive pulse is generated synchronized with f_{CLK} by randomly extending the low width using pseudo-random numbers based on the base clock obtained by dividing the operating clock by the CTSUSO1.SDPA[7:0] bits. Using the base clock obtained by dividing the operating clock using the CTSUSO1.SDPA[7:0] bits as a reference, a sensor drive pulse synchronized with f_{CLK} is generated by randomly extending the low width using pseudo-random numbers. In addition, SUCLK is generated according to the settings of CTSUCRBH.SSCNT[1:0], CTSUCRBH. SSMOD[2:0], and CTSUTRIM1.SUADJD[7:0]. The generated SUCLK is used for frequency spreading of the sensor drive pulse.
1	• High-resolution pulse mode The clock (base clock) obtained by dividing SUCLK by the value specified in the CTSUSO1. SDPA[7:0] bits is used as the sensor drive pulse (synchronized with SUCLK). Setting the CTSUSUCLKx registers (x = 0 to 3) provides clock recovery control and updates the CTSUSUCLKx.SUADJx[7:0] bits. Set the CTSUSUCLKx register before setting this bit to 1. Do not change the CTSUSUCLKx register while this bit is 1.
The SDPSEL bit toggles between random pulse mode and hi-resolution pulse mode. The setting of this bit changes how the sensor drive pulse and SUCLK are generated.	

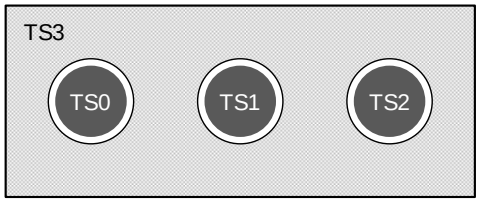
POSEL[1:0]		Non-measured Pin Output Select
0	0	Low-level output (from the I/O port regardless of the TXVSEL and TXVSEL2 settings)
0	1	Hi-z (high impedance) output
1	0	Low-level output (by the TXVSEL and TXVSEL2 settings)
1	1	Output pulses in phase with the transmit channel (by the TXVSEL and TXVSEL2 settings)

The POSEL[1:0] bits select the non-measured pin output.

[Example]

When measurement is performed by the self-capacitance method using the electrode shown below to enable the active shield function:

- Shield pin: TS3
  When the transmit/receive control register is set to 1, a transmit pulse is output and TS3 is used as the shield pin.
- Measurement pins: TS0 to TS2
  Set the POSEL[1:0] bits to 11B. The non-measuring pin outputs the same pulses as the shielded pin.



LOAD[1:0]		Measurement Load Control
0	0	Constant current load mode (2.5 µA/1.25 µA)
0	1	No-load mode
1	0	Setting prohibited
1	1	Resistive load mode (for calibration)
To place the measurement load in the resistive load mode, set these bits to 01B and then set them to 11B. When in resistive load mode, resistive load changes depending on the settings of bits ATUNE1 and ATUNE2.		

ATUNE2	Analog Adjustment 2
0	Follows the ATUNE1 setting.
1	When ATUNE1 is 0: 20 µA (1/1) When ATUNE1 is 1: 160 µA (1/8)
This bit and the ATUNE1 bit change the amount of resistive load on the measurement power supply and set the maximum supply current (measurement range).	

MD1	Measurement Mode Select 1
0	Self-capacitance method (single measurement) When CTSUCHTRCA.CHTRCm (m = 00 to 30) is set to 1 (transmission), the transmit pin outputs the first (same phase) pulse. Set CHTRCm = 0 (receive) for the pin to be measured. When CHTRCm is set to all 0, measurement is performed without transmission.
1	Mutual capacitance method (double-measurement requiring CHTRCm setting) When CHTRCm is set to all 0s, no measurement performed.
This bit selects between self-capacitance method and mutual capacitance method.	

MD0	Measurement Mode Select 0
0	Single scan mode
1	Multi-scan mode
This bit selects between single scan mode and multi-scan mode.	

CLK[1:0]		Peripheral Operating Clock Select
0	0	f_{CLK}
0	1	$f_{CLK}/2$
1	0	$f_{CLK}/4$
1	1	$f_{CLK}/8$
Based on the settings of these bits, the prescaler divides the CPU/peripheral hardware clock (f_{CLK}).		

ATUNE1	Analog Adjustment 1
0	80 µA (1/4)
1	40 µA (1/2)
This bit and the ATUNE2 bit change the amount of resistive load on the measurement power supply and set the maximum supply current (measurement range).	

CSW	Measurement Power Capacitor Connection
0	Turns off the external capacitance connection switch.
1	Turns on the external capacitance connection switch.

This bit controls charging of the LPF capacitance connected to the TSCAP pin by turning the capacitance switch on or off.
After the capacitance switch is turned on, wait for the specified time until the capacitance connected to the TSCAP pin is charged, and then set STRT to 1 to start measurement.
Before setting this bit to 1, output a low level from the I/O port (P12) to which TSCAP is assigned to discharge the charge of the capacitor connected to the TSCAP pin.
Set the CSW bit to 1 before setting the PON bit to 1.
When V_{DD} < 4.5 V, set the PUMPON bit to 1 and then set the CSW bit to 1.

PON	Measurement Power On
0	Power off
1	Power on

This bit controls the supply of the measurement power supply (1.5 V).
Before setting this bit to 1, set the PUMPON bit and set the CSW bit to 1.
Do not set this bit to 1 when the CSW bit is 0.

TXVSEL	TXVSEL2	Transmission Power Supply Select
0	0	The transmission pin is not used. (When using the transmission pin, set the following values.)
1	0	When using the active shield function.
0/1	1	When using the mutual capacitance measurement mode.

PUMPON	Boost Circuit Control
0	Boost power off VCCX3 = V_{DD}
1	Boost power on VCCX3 ≈ 4.5 V

This bit controls the boost power voltage (VCCX3).
If the V_{DD} voltage is lower than 4.5 V, set this bit to 1. If the V_{DD} voltage is 4.5 V or higher, set this bit to 0 or 1.

INIT	Control Block Initialization
0	–
1	Initializes the internal control registers.

When this bit is set to 1, the following registers are initialized:

- CTSUSC, CTSUUC
- CTSUMCHL, H
- CTSUSRL (except CTSUSRL.ICOMP0 and CTSUSRL.ICOMP1)

To forcibly terminate operation, set the STRT bit to 0 and the INIT bit to 1 simultaneously. In this case, operation stops and the internal control registers are initialized.
Do not set the STRT bit to 1 (CTSU operation start) and the INIT bit to 1 simultaneously.
This bit is read as 0.

SNZ	SNOOZE Enable
0	Disables the SNOOZE function.
1	Enables the SNOOZE function.

This bit enables or disables the SNOOZE function when an external trigger is selected as the measurement start condition (CAP = 1). Setting this bit to 1 drives the CTSU hardware into the suspended state to enable low-power operation in the standby state.

<CTSU hardware state control>

PON	SNZ	CAP	STRT	External trigger	Hardware (VDC) state
0	0	0	0	–	Stopped
1	0	–	–	–	Operating
1	1	1	0	–	Suspended
1	1	1	1	None (waiting)	Suspended
1	1	1	1	Provided (active)	Operating (measurement in progress)
Other than above					Setting prohibited

By selecting the external trigger as the measurement start condition and setting the STRT bit to 1 with the SNOOZE function enabled, the CPU can enter STOP mode in the external trigger waiting state. When the external trigger is detected during STOP mode, the CTSU sends a clock request to the clock generating block and enters the SNOOZE state to start measurement. After the measurement end interrupt, clear this bit to 0 by software.

When selecting the hi-resolution mode for the sensor drive pulse (CTSUCRAH.SDPSEL = 1) and using the SNOOZE function, it is prohibited to select the sensor drive pulse for the boost clock (CTSUCRAH.PCSEL = 0).

CAP	Measurement Start Trigger Select
0	Software trigger
1	External trigger

This bit specifies the measurement start condition.
For details, see the STRT bit explanation.

When the CAP bit is set to 1, be sure to set CTSUTRG.CTSUTRGEN bit to 1 as well.
Similarly, when clear the CAP bit, be sure to clear CTSUTRG.CTSUTRGEN bit as well.

STRT	Measurement Operation Start
0	Measurement stopped
1	Measurement operating

This bit specifies the start/stop of CTSU measurement operation.
When CAP is 0, writing 1 to the STRT bit starts measurement. At the end of measurement, this bit is cleared to 0 by hardware.
When CAP is 1, writing 1 to this bit drives the CTSU into the external trigger waiting state. Measurement starts when the external trigger is detected. After the measurement is complete, the CTSU waits for the next external trigger and continues operation. When exit the external trigger waiting state, set the STRT bit to 0 and the INIT bit to 1 simultaneously.
The following table shows the CTSU states.

<CTSU state>

STRT	CAP	CTSU state
0	0	Stopped
0	1	Stopped
1	0	Measurement in progress
1	1	Measurement in progress or waiting for an external trigger Note

Note The state can be read from the CTSUSRL.STC[2:0] bits.
- CTSUSRL.STC[2:0] ≠ 000B: Measurement in progress:
- CTSUSRL.STC[2:0] = 000B: While waiting for an external trigger:

If 1 is written to the STRT bit by software when the STRT bit is already set to 1, this writing is ignored and operation continues.
To forcibly terminate operation by software when the STRT bit is set to 1, set the STRT bit to 0 and the INIT bit to 1 simultaneously.

15.2.3 CTSU Control Register B (CTSUCRBL, CTSUCRBH)

The CTSUCRBL and CTSUCRBH registers can be set by 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears these registers to 0000H.
Set this register when CTSUCRAL.STRT = 0 (measurement operation stopped).

Figure 15-8. Format of CTSU Control Register B

Address: (16-bit access) CTSUCRBH: F0186H
(8-bit access) CTSUDCLKC: F0187H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCRBH	0	0	SSCNT[1:0]		0	SSMOD[2:0]		

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Address: (16-bit access) CTSUCRBL: F0184H
(8-bit access) CTSUSDPRS: F0184H CTSUSST: F0185H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCRBL	SST[7:0]							

7	6	5	4	3	2	1	0
PROFF	SOFF	PRMODE[1:0]		PRRATIO[3:0]			

SSCNT[1:0]		SUCLK Diffusion Control
0	0	CTSUTRIM1.SUADJD[7:0] + 00H
0	1	CTSUTRIM1.SUADJD[7:0] + 20H
1	0	CTSUTRIM1.SUADJD[7:0] + 40H
1	1	CTSUTRIM1.SUADJD[7:0] + 60H
These bits adjust the frequency of the SUCLK generation circuit specified by the CTSUTRIM1.SUADJD[7:0] bits.		

SSMOD[2:0]			SUCLK Diffusion Mode Select
0	0	0	256 cycles
0	0	1	384 cycles
0	1	0	512 cycles
0	1	1	1024 cycles
1	1	1	No diffusion
Other than above			Setting prohibited
These bits sets the spread spectrum period of the SUCLK generation circuit. The SUCLK oscillation frequency is determined by the frequency specified by the CTSUTRIM1.SUADJD[7:0], SSCNT[1:0] bits, and the settings of these bits.			

SST[7:0]	Sensor Stabilization Wait Time Control

The SST[7:0] bits set the period from the start of sensor drive pulse supply until the TSCAP pin voltage becomes stable.

The TSCAP pin voltage is stabilized by supplying sensor drive pulses. The SST[7:0] value is related to the number of cycles as shown below.

<When CTSUCRAH.SDPSEL = 0>

The table below shows the stabilization time in units of base sensor drive pulse cycles.

0	0	0	0	0	0	0	0	2 cycles
0	0	0	0	0	0	0	1	4 cycles
0	0	0	0	0	0	1	0	6 cycles
0	0	0	0	0	0	1	1	8 cycles
:								:
1	1	1	1	1	1	1	0	510 cycles
1	1	1	1	1	1	1	1	512 cycles

Number of cycles = 2 × (value of these bits + 1)

Set the stabilization wait time within the following range.

Number of cycles set by CTSUCRBL.SST[7:0] ≥ (CTSUCRBL.PRRATIO[3:0] + 1)

<When CTSUCRAH.SDPSEL = 1>

The table below shows the stabilization time in units of STCLK cycles.

0	0	0	0	0	0	0	0	Setting prohibited Note
0	0	0	0	0	0	0	1	2 cycles
0	0	0	0	0	0	1	0	3 cycles
0	0	0	0	0	0	1	1	4 cycles
:								:
1	1	1	1	1	1	1	0	255 cycles
1	1	1	1	1	1	1	1	256 cycles

Number of cycles = 1 × (value of these bits + 1)

Note When CTSUCRAH.SDPSEL = 1, do not set these bits to 00H..

PROFF	Pseudo-Random Number OFF Control
0	Pseudo-random number control is enabled.
1	Pseudo-random number control is disabled.

This bit turns off pseudo-random number control.
Generates a pseudo-random number every cycle and outputs 1 or 0. (1-bit random number generation).

SOFF	Spread Spectrum Function OFF Control
0	Spread-spectrum function is ON
1	Spread-spectrum function is OFF

This bit sets enable or disable of the frequency spread function to prevent synchronization system noise.
Set this bit to 1 when using sensor drive pulses (f_{CLK} synchronous).
This bit is valid when CTSUCRAH.SDPSEL = 0 (random pulse mode).

PRMODE[1:0]		Pseudo-Random Number Generation Cycle
0	0	255 cycles
0	1	63 cycles
1	0	31 cycles
1	1	3 cycles

This bit sets the period for updating the pseudo-random number generator.
The basic number of pulses is twice the period selected by the PRMODE[1:0] bits.
These bits are valid when CTSUCRAH.SDPSEL = 0 (random pulse mode).

PRRATIO[3:0]	Phase Shift Frequency

These bits set the phase shift period for pseudo-random number generation.
Depending on the settings of these bits, the number of measurement pulses and measurement time will change as follows:

Number of measurement pulses
= Number of basic pulses (set with PRMODE[1:0] bits) × (Value of PRRATIO[3:0] bits + 1)

Measurement time
= (Number of measurement pulses + (Number of basic pulses − 2) × 0.25) × Base clock period

These bits are valid when CTSUCRAH.SDPSEL = 0 (random pulse mode).

15.2.4 CTSU Measurement Channel Register (CTSUMCHL, CTSUMCHH)

The CTSUMCHL and CTSUMCHH registers can be set by 8-bit or 16-bit memory manipulation instruction.

Reset signal generation set the CTSUMCHH register to 0000H and CTSUMCHL register to 3F3FH (RL78/F25) or 0F0FH (RL78/F22).

Set this register when CTSUCRAL.STRT = 0 (measurement operation stopped).

Figure 15-9. Format of CTSU Measurement Channel Registers

Address: (16-bit access) CTSUMCHH: F018AH
(8-bit access) CTSUMFAF: F018AH — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUMCHH	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	MCA3	MCA2	MCA1	MCA0

Address: (16-bit access) CTSUMCHL: F0188H
(8-bit access) CTSUMCH0: F0188H CTSUMCH1: F0189H — After reset: RL78/F25: 3F3FH, RL78/F22: 0F0FH — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUMCHL	0	0	MCH1[5:0]					

7	6	5	4	3	2	1	0
0	0	MCH0[5:0]					

MCAx	Multi-clock x Enable (x = 0 to 3)
0	Disabled
1	Enabled

These bits are used when measuring using multiple SUCLKs with different frequencies (multi-clock measurement). For measuring with three different frequencies, set 3 of the 4 bits to 1.

When measurement starts with two or more MCAx bits enabled, valid channels are scanned in ascending order from MCA0 (SUCLK0). After the measurement of one measurement channel is completed, if the next valid clock (SUCLKx) is set, the same channel is measured with that clock. If no valid clock is set, measurement shifts to the next measurement channel.

The SUCLKx frequency is set by the CTSUSUCLKx registers.

- MCA0: Enables SUCLK0 set by the CTSUSUCLK0.SUMULTI0[7:0] and CTSUSUCLK0.SUADJ0[7:0] bits
- MCA1: Enables SUCLK1 set by the CTSUSUCLK1.SUMULTI1[7:0] and CTSUSUCLK1.SUADJ1[7:0] bits
- MCA2: Enables SUCLK2 set by the CTSUSUCLK2.SUMULTI2[7:0] and CTSUSUCLK2.SUADJ2[7:0] bits
- MCA3: Enables SUCLK3 set by the CTSUSUCLK3.SUMULTI3[7:0] and CTSUSUCLK3.SUADJ3[7:0] bits

MCH1[5:0]	Measurement Channel 1

- In single scan mode (CTSUCRAL.MD0 = 0), the MCH1[5:0] bits set the transmit channel to be measured. Do not set channel as transmit channel that is not targeted for measurement by the CTSUCHACAL or CTSUCHACAH registers. If such a channels is set, the measurement will be completed immediately after it starts.
- In multi-scan mode (CTSUCRAL.MD0 = 1), the MCH1[5:0] bits indicate the value of the transmit channel that is being measured, and writing to these bits are invalid.

MCH1[5:0] Note						Transmit Channel
0	0	0	0	0	0	TS0
0	0	0	0	0	1	TS1
0	0	0	0	1	0	TS2
0	0	0	0	1	1	TS3
:						:
0	1	1	1	0	1	TS29
0	1	1	1	1	0	TS30
Other than above						Setting prohibited

Note For RL78/F22 products, writing to MCH1[5:4] is invalid.

Do not modify these bits during measurement (CTSUCRAL.STRT = 1). Otherwise, operation is not guaranteed. When measurement is stopped, these bits become 111111B for RL78/F25 and 001111B for RL78/F22 products.

MCH0[5:0]	Measurement Channel 0

- In single scan mode (CTSUCRAL.MD0 = 0), the MCH0[5:0] bits set the receive channel to be measured. Do not set channel as receive channel that is not targeted for measurement by the CTSUCHACAL or CTSUCHACAH registers. If such a channel is set, the measurement will be completed immediately after it starts.
- In multi-scan mode (CTSUCRAL.MD0 = 1), the MCH0[5:0] bits indicate the value of the receive channel that is being measured, and writing to these bits are invalid.

MCH0[5:0] Note						Receive Channel
0	0	0	0	0	0	TS0
0	0	0	0	0	1	TS1
0	0	0	0	1	0	TS2
0	0	0	0	1	1	TS3
:						:
0	1	1	1	0	1	TS29
0	1	1	1	1	0	TS30
Other than above						Setting prohibited

Note For RL78/F22 products, writing to MCH0[5:4] is invalid.

Do not modify these bits during measurement (CTSUCRAL.STRT = 1). Otherwise, operation is not guaranteed. When measurement is stopped, these bits become 111111B for RL78/F25 and 001111B for RL78/F22 products.

15.2.5 CTSU Channel Enable Control Register A (CTSUCHACAL, CTSUCHACAH)

The CTSUCHACAL and CTSUCHACAH registers can be set by 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears these registers to 0000H.
Set these registers when CTSUCRAL.STRT = 0 (measurement operation stopped).

Figure 15-10. Format of CTSU Channel Enable Control Register A

Address: (16-bit access) CTSUCHACAH: F018EH
(8-bit access) CTSUCHAC2: F018EH CTSUCHAC3: F018FH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCHACAH	0	CHAC30	CHAC29	CHAC28	CHAC27	CHAC26	CHAC25	CHAC24

7	6	5	4	3	2	1	0
CHAC23	CHAC22	CHAC21	CHAC20	CHAC19	CHAC18	CHAC17	CHAC16

Address: (16-bit access) CTSUCHACAL: F018CH
(8-bit access) CTSUCHAC0: F018CH CTSUCHAC1: F018DH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCHACAL	CHAC15	CHAC14	CHAC13	CHAC12	CHAC11	CHAC10	CHAC09	CHAC08

7	6	5	4	3	2	1	0
CHAC07	CHAC06	CHAC05	CHAC04	CHAC03	CHAC02	CHAC01	CHAC00

CHACm	Channel m Enable Control (m = 00 to 30)
0	TSm pin measurement disable.
1	TSm pin measurement enable.
Set 1 to CHACm bits to enable capacitance measurement of TSm pins (both transmitting and receiving). Write 0 to unassigned bits.	

Caution RL78/F22 products do not have CTSUCHACAH register.

15.2.6 CTSU Channel Transmit/Receive Control Register A (CTSUCHTRCAL, CTSUCHTRCAH)

The CTSUCHTRCAL and CTSUCHTRCAH registers can be set by 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears these registers to 0000H.
Set these registers when CTSUCRAL.STRT = 0 (measurement operation stopped).

Figure 15-11. Format of CTSU Channel Transmit/Receive Control Register A

Address: (16-bit access) CTSUCHTRCAH: F0196H
(8-bit access) CTSUCHTRC2: F0196H CTSUCHTRC3: F0197H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCHTRCAH	0	CHTRC30	CHTRC29	CHTRC28	CHTRC27	CHTRC26	CHTRC25	CHTRC24

7	6	5	4	3	2	1	0
CHTRC23	CHTRC22	CHTRC21	CHTRC20	CHTRC19	CHTRC18	CHTRC17	CHTRC16

Address: (16-bit access) CTSUCHTRCAL: F0194H
(8-bit access) CTSUCHTRC0: F0194H CTSUCHTRC1: F0195H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCHTRCAL	CHTRC15	CHTRC14	CHTRC13	CHTRC12	CHTRC11	CHTRC10	CHTRC09	CHTRC08

7	6	5	4	3	2	1	0
CHTRC07	CHTRC06	CHTRC05	CHTRC04	CHTRC03	CHTRC02	CHTRC01	CHTRC00

CHTRCm	Channel m Transmit/Receive Control (m = 00 to 30)
0	Reception
1	Transmission

Assign TSm pins as transmit or receive pins depending on the value of CHTRCm.
Write 0 to unassigned bits.
When MD1 is set to 0, if one of these bits is set to 1 (transmission), the corresponding pin can be used for shield signal output. However, when setting as a shield output, do not set two or more bits to 1.

Caution RL78/F22 products do not have CTSUCHTRCAH register.

15.2.7 CTSU Status Register (CTSUSRL)

The CTSUSRL register can be set by 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears these registers to 0000H.

Figure 15-12. Format of CTSU Status Register

Address: (16-bit access) CTSUSRL: F019CH
(8-bit access) CTSUST1: F019CH CTSUST: F019DH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
CTSUSRL	PS	SUCKOVF	SENSOVF	DTSR	0	STC[2:0]		

7	6	5	4	3	2	1	0
ICOMP0	ICOMP1	ICOMPRST	0	0	0	MFC[1:0]	

PS	Mutual Capacitance Measurement Status
0	First measurement
1	Second measurement

This bit indicates whether the measurement is the first or second measurement for each channel in mutual capacitance method measurement (CTSUCRAL.MD1 = 1).
This bit is 0 when measurement is stopped or during self-capacitance method measurement (MD1 = 0).
This bit is read only.

SUCKOVF	SUCLK Counter Overflow
0	No overflow occurred.
1	An overflow occurred.

This bit indicates whether the SUCLK counter has overflowed.
When an overflow occurs, the result of measurement is read from the CTSUUC.UC[15:0] bits as FFFFH.
Even if an overflow occurs, measurement processing continues for the set period.
Occurrence of an overflow generates no interrupt. Therefore, read the measurement result of each channel after measurement is complete (after a measurement end interrupt is generated) to check an overflowed channel.
This bit is cleared when 0 is written after 1 is read by software. Also, this bit is cleared by the CTSUCRAL.INIT bit.

SENSOVF	Sensor Counter Overflow
0	No overflow occurred.
1	An overflow occurred.

This bit indicates whether the sensor counter has overflowed.
When an overflow occurs, the result of measurement is read from the CTSUSC.SC[15:0] bits as FFFFH.
Even if an overflow occurs, measurement processing continues for the set period.
Occurrence of an overflow generates no interrupt. Therefore, read the measurement result of each channel after measurement is complete (after a measurement end interrupt is generated) to check an overflowed channel.
This bit is cleared when 0 is written after 1 is read by software. Also, this bit is cleared by the CTSUCRAL.INIT bit.

DTSR	Data Transfer Status
0	The measurement result has been read.
1	The measurement result has not been read.

This bit indicates whether the measurement result stored in the sensor counter has been read. This bit is set to 1 on completion of measurement and is set to 0 when the CTSUSC or CTSUUC register is read by software or the DTC.
Also, this bit is set to 0 by the CTSUCRAL.INIT bit.
This bit is read only.

STC[2:0]			Measurement Status Counter
0	0	0	State 0
0	0	1	State 1
0	1	0	State 2
0	1	1	State 3
1	0	0	State 4
1	0	1	State 5

These bits indicate the current measurement status. See **15.1.2 Measurement Status** for details.
These bits are read only.

ICOMP0	Voltage Error (Comparison Results of Reference Resistors)
0	Normal
1	Voltage error detected.

This bit indicates that a voltage error has been detected as a result of comparing the reference resistance.
This bit is set to 0 when CTSUCRAL.PON = 0 or writing 1 to ICOMPRST bit.
When this bit is 1 (voltage abnormality detected), the value of the CTSUSU.SC[15:0] bits will be 0000H.
This bit is read only.

ICOMP1	Current Error
0	Normal
1	Current error detected.

This bit indicates that an error has been detected in the output current of the measurement power supply.
This bit is set to 0 when CTSUCRAL.PON = 0 or writing 1 to ICOMPRST bit.
This bit is read only.

ICOMPRST	ICOMP0 and ICOMP1 Reset

Writing 1 to this bit resets the ICOMP1 and ICOMP0 bits.
Reading this bit is 0.

MFC[1:0]		Multi-clock Counter
0	0	MCA0 (SUCLK0)
0	1	MCA1 (SUCLK1)
1	0	MCA2 (SUCLK2)
1	1	MCA3 (SUCLK3)

These bits indicate the clock that is being measured during multi-clock measurement (when CTSUCRAH.SDPSEL = 1, CTSUCRAH.PCSEL = 1, and two or more of the CTSUMCH.MCAn bits (n = 0-3) are 1).
Writing values other than 00B is prohibited.

15.2.8 CTSU Sensor Offset Register (CTSUSO0, CTSUSO1)

The CTSUSO0 and CTSUSO1 registers can be set by 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

Write to these registers after the INTCTSUWR interrupt is generated.

A write operation to CTSUSO1 register causes a transition to State 3. Therefore, be sure to write to CTSUSO0 and then to CTSUSO1. Also, all bits should be set at once when write to CTSUSO1.

Figure 15-13. Format of CTSU Sensor Offset Register

Address: (16-bit access only) CTSUSO1: F01A2H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUSO1	SDPA[7:0]							

7	6	5	4	3	2	1	0
SSDIV[3:0]				0	0	SNUM[7:6]	

Address: (16-bit access only) CTSUSO0: F01A0H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUSO0	SNUM[5:0]						SO[9:8]	

7	6	5	4	3	2	1	0
SO[7:0]							

SDPA[7:0]	Base Clock Setting

Generate the base clock as follows.

<When CTSUCRAH.SDPSEL = 0>

The operating clock is divided to generate a base clock to be the source of sensor drive pulse. These bits are also used for setting the voltage stabilization time of the CTSU hardware.

0	0	0	0	0	0	0	0	Operating clock divided by 2 Note
0	0	0	0	0	0	0	1	Operating clock divided by 4
0	0	0	0	0	0	1	0	Operating clock divided by 6
0	0	0	0	0	0	1	1	Operating clock divided by 8
:								:
1	1	1	1	1	1	1	0	Operating clock divided by 510
1	1	1	1	1	1	1	1	Operating clock divided by 512

<When CTSUCRAH.SDPSEL = 1>

The SUCLK clock is divided to generate a sensor drive pulse.

0	0	0	0	0	0	0	0	SUCLK divided by 2
0	0	0	0	0	0	0	1	SUCLK divided by 4
0	0	0	0	0	0	1	0	SUCLK divided by 6
0	0	0	0	0	0	1	1	SUCLK divided by 8
:								:
1	1	1	1	1	1	1	0	SUCLK divided by 510
1	1	1	1	1	1	1	1	SUCLK divided by 512

Note When using the mutual capacitance method and the spectrum spreading function is OFF (CTSUCRBL.SOFF = 1), setting of SDPA[7:0] = 00000000B is prohibited.

SSDIV[3:0]				Spectrum Diffusion Sampling Cycle Control
0	0	0	0	Divided by 1
0	0	0	1	Divided by 2
:				
1	1	1	0	Divided by 15
1	1	1	1	Divided by 16

These bits set the sampling period of the frequency spreading function.
These bits select the clock division to use when resampling the sensor drive pulse synchronized with f_{CLK} using SUCLK.
Set the sampling cycle to less than 1/4 of the sensor drive pulse cycle.
Set these bits for the pin to be measured after the INTCTSUWR interrupt is generated.
These bits are valid only when CTSUCRAH.SDPSEL is 0 (random pulse mode).

SNUM[7:0]	Measurement Count/Time Setting

<When CTSUCRAH.SDPSEL = 0>
The measurement count is set.
These bits specify how many times the number of measurement pulses (basic measurement unit) set by the CTSUCRBL.PRRATIO[3:0] and CTSUCRBL.PRMODE[1:0] bits are repeated during measurement.
Measurement count = (SNUM[5:0] + 1) × 2
Set the SNUM[7:6] bits to 00B.

<When CTSUCRAH.SDPSEL = 1>
The measurement time based on the STCLK cycle is set.
Measurement time = 8 × (value of SNUM[7:0] bits + 1) × STCLK cycles

Set these bits for the pin to be measured after the INTCTSUWR interrupt is generated.

SO[9:0]	Sensor Offset Adjustment

These bits adjust the input current offset of the sensor CCO.
These bits are used to offset the sensor CCO input current generated from electrostatic capacitance while the electrode is not being touched during touch measurement, thus preventing overflow of the CTSU sensor counter.
Set these bits for the pin to be measured after the INTCTSUWR interrupt is generated.
After an INTCTSUWR interrupt occurs, set these bits to the input current offset of the pin to be measured next.

15.2.9 CTSU Sensor Counter Register (CTSUSC, CTSUUC)

The CTSUSC and CTSUUC registers are accessed using 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

This register is read only.

Figure 15-14. Format of CTSU Sensor Counter Register

Address: (16-bit access only) CTSUUC: F01A6H — After reset: 0000H — R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUUC	UC[15:0]															

Address: (16-bit access only) CTSUSC: F01A4H — After reset: 0000H — R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUSC	SC[15:0]															

UC[15:0]	SUCLK Counter
These bits indicate the count value of the clock (SUCLK × 2). When the counter indicates FFFFH, it is judged that an overflow has occurred.	

SC[15:0]	Sensor Counter
These bits indicate the count value of the clock generated by the sensor CCO. When the counter indicates FFFFH, it is judged that an overflow has occurred.	

Read these registers after the INTCTSURD interrupt occurs.

When the value of this register is read the number of times set by the CTSUCALIB.CNTRDSEL bit, the measurement status transitions to State 0 or State 2.

The values of these registers are cleared just before the measurement state transitions to State 4 on the next measurement. These counters are also cleared by the CTSUCRAL.INIT bit.

- When CTSUDBGR0.CNTRDSEL = 0;
  Reading either the SC[15:0] or UC[15:0] bits once advances the measurement status and both counters are cleared before transitioning to State 4. If measurement results are required, be sure to read SC[15:0].
- When CTSUDBGR0.CNTRDSEL = 1;
  Reading either the SC[15:0] or UC[15:0] bits twice in total advances the measurement status.
  Use this mode when the results of both SC[15:0] and UC[15:0] bits are needed in a 16-bit access.

15.2.10 CTSU Calibration Register (CTSUDBGR0, CTSUDBGR1)

The CTSUDBGR0 and CTSUDBGR1 registers can be set by 16-bit memory manipulation instruction.
Reset signal generation clears these registers to 0000H.

Figure 15-15. Format of CTSU Calibration Register

Address: (16-bit access only) CTSUDBGR1: F01AAH — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUDBGR1	TXREV	CCOCALIB	CCOCLK	DACCLK	SUCARRY	SUMSEL	DACCARRY	DACMSEL
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

Address: (16-bit access only) CTSUDBGR0: F01A8H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUDBGR0	0	0	0	IOCSEL	DCOFF	0	IOC	CNTRDSEL
	7	6	5	4	3	2	1	0
	TSOC	0	0	0	DRV	TSOD	0	0

TXREV	Transmit Pin Inverted Output
0	The pulse output from the transmit pin is not inverted.
1	The pulse output from the transmit pin is inverted.

Setting this bit to 1 inverts the pulse output from the transmit pin.

<When in self-capacitance method (CTSUCRAL.MD1 = 0) >
0: Outputs a signal with the same phase as the receive pin
1: Outputs a signal with the opposite phase to the receive pin
<When in mutual capacitance method (CTSUCRAL.MD1 = 1) >
0: 1st time: Outputs a signal with the same phase as the receive pin,
2nd time: Outputs a signal with the opposite phase to the receive pin
1: 1st time: Outputs a signal with the opposite phase to the receive pin
2nd time: Outputs a signal with the same phase as the receive pin

Set this bit to 0 for normal measurement (setting to 1 is assumed for debugging).

CCOCALIB	CCO Calibration Mode Select

MODE circuit state

CCOCALIB = 0: Normal mode
LDO current → Oscillator for current measurement
SSCNT current → SUCLK oscillator
CCOCALIB = 1: Oscillator calibration mode
LDO current → SUCLK oscillator
SSCNT current → Oscillator for current measurement

The oscillator calibration mode is used when the external current is compared with the output of the current DAC to correct the oscillator characteristics by software.

CCOCLK	CCO Modulation Circuit Clock Select
0	The clock specified by the CTSUCRAL.CLK[1:0] bits
1	SUCLK

DACCLK	DAC Modulation Circuit Clock Select
0	The clock specified by the CTSUCRAL.CLK[1:0] bits
1	SUCLK

SUCARRY	CCO Carry Input (Usually Fixed to 0)
0	Normal measurement.
1	A value of 32 is added to the input data.
This bit is used in the current oscillator input current test mode.	

SUMSEL	Current Oscillator Input Current Matrix Test
0	Normal operation
1	Current oscillator input current test mode
This bit controls the input current for the current oscillator.	

DACCARRY	DAC Upper Current Source Carry Input (Usually Fixed to 0)
0	Normal measurement.
1	A value of 64 is added to the input data.
This bit is used in the current offset DAC current test mode.	

DACMSEL	Current Offset DAC Current Matrix Test
0	Normal operation
1	Current offset DAC current test mode
This bit controls the current for the current offset DAC. Set this bit to 0 when measuring capacitance.	

IOCSEL	TSm Pin Fixed Output Selection Bit (m = 0 to 30)
0	Sensor drive pulse
1	Outputs the fixed level selected by the CTSUDBGR0.IOC bit.
This bit is used for CTSU calibration. When the CTSUDBGR0.TSOD bit is 1 and this bit is 0, sensor drive pulses are output from the TSm pin that is set to be measured (= 1) in the CTSUCHACAL, H registers and received (= 0) in the CTSUCHTRCAL, H registers. This is used in MEC (Multiple Electrode Connection) function. When the CTSUDBGR0.TSOD bit is 1 and this bit is 1, the level specified by the CTSUDBGR0.IOC bit is output to all TS pins.	

DCOFF	Down-conversion OFF
0	Normal operation mode
1	Down-conversion OFF
This bit turns off the LDO output. Set this bit to 0 when measuring capacitance.	

IOC	Transmit Pin Control
The IOC bit selects the output level from the TSm (m = 0 to 30) pin when the CTSUDBGR0.TSOD bit or the CTSUCRAH. DCMODE bit is set to 1. <When CTSUDBGR0.TSOD = 1> 0: Low level is output from the TSm pin. 1: High level is output from the TSm pin. <When CTSUCRAH.DCMODE = 1> 0: High level is output from the TSm pin. 1: Low level is output from the TSm pin. When the CTSUDBGR0.TSOD and CTSUCRAH.DCMODE bits are 0, this bit is invalid.	

CNTRDSEL	Sensor Counter Register Read Count Select
0	Transitions to the next state after single reading.
1	Transitions to the next state after reading twice.
To read both CTSUSC.SC[15:0] and CTSUUC.UC[15:0] bits, set this bit to 1.	

TSOC	Calibration Setting 2
0	Electrostatic capacitance measurement mode
1	Calibration setting 2
The TSOC bit is used to set calibration of the CTSU. When measuring electrostatic capacitance, set this bit to 0.	

DRV	Calibration Setting 1
0	Electrostatic capacitance measurement mode
1	Calibration setting 1
The DRV bit is used to set calibration of the CTSU. When measuring electrostatic capacitance, set this bit to 0. When the DRV bit is set to 1, the state is pseudo to State 3 or State 4. It differs from State 3 and State 4 during measurement in the following points. • Sensor drive pulse is not output. • Enable and reset for the measurement counter is not output.	

TSOD	All TSm Pins Output Control (m = 0 to 30)
0	Electrostatic capacitance measurement mode
1	The signals selected by the IOCSEL bit are output from TSm pins for the Multiple Electrode Connection (MEC) function or for calibration.

This bit is used for the MEC function or to set calibration of the CTSU. When this bit is set to 1, the signal specified by the IOCSEL bit is output from TSm pin.

<Mode selection for measurement pins>

TSOD	IOCSEL	Mode for measurement pins
0	0	Electrostatic capacitance measurement mode is selected.
0	1	–
1	0	When the MEC function is selected, sensor drive pulses are output from all TSm pins set for measurement or calibration. When using the MEC function, set the CTSUCRAL.MD0 bit to 0 (single scan mode).
1	1	The level specified by the IOC bit is output from TSm pin.

15.2.11 CTSU Sensor Unit Clock Control Registers A, B (CTSUSUCLKx) [x: 0 to 3]

The CTSU Sensor Unit Clock Control Registers A (CTSUSUCLK0, CTSUSUCLK1), CTSU Sensor Unit Clock Control Registers B (CTSUSUCLK2, CTSUSUCLK3) are set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 15-16. Format of CTSU Sensor Unit Clock Control Registers A, B

Address: (16-bit access only) CTSUSUCLK3: F01B2H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUSUCLK3	SUMULTI3[7:0]							

7	6	5	4	3	2	1	0
SUADJ3[7:0]							

Address: (16-bit access only) CTSUSUCLK2: F01B0H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUSUCLK2	SUMULTI2[7:0]							

7	6	5	4	3	2	1	0
SUADJ2[7:0]							

Address: (16-bit access only) CTSUSUCLK1: F01AEH — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUSUCLK1	SUMULTI1[7:0]							

7	6	5	4	3	2	1	0
SUADJ1[7:0]							

Address: (16-bit access only) CTSUSUCLK0: F01ACH — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUSUCLK0	SUMULTI0[7:0]							

7	6	5	4	3	2	1	0
SUADJ0[7:0]							

SUMULTIx[7:0]	SUCLK Multiplication Rate Setting

Set the multiplication rate for STCLK (assuming a frequency division of 0.5 MHz for f_{CLK}) to generate SUCLK.
The SUCLK generated by this setting updates the SUADJx bit.
The target clock frequency must be set between 16 MHz and 38 MHz.

0	0	0	0	0	0	0	0	× 1
				:				:
0	0	1	1	1	1	1	1	× 64
				:				:
0	1	1	1	1	1	1	1	× 128
				:				:
1	1	1	1	1	1	1	1	× 256

SUADJx[7:0]	SUCLK Frequency Adjustment

These bits set the initial value of the SUCLK frequency.
SUCLK is updated by the clock recovery control.
The updated value should not be rewritten.

15.2.12 CTSU Option Setting Register (CTSUOPTL, CTSUOPTH)

The CTSUOPTL and CTSUOPTH registers can be set by 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears these registers to 0000H.
Set these registers when CTSUCRAL.STRT = 0 (measurement operation stopped).

Figure 15-17. Format of CTSU Option Setting Register

Address: (16-bit access) CTSUOPTH: F01C2H
(8-bit access) CTSUOPTACSB: F01C2H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUOPTH	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	SCACTB[3:0]			

Address: (16-bit access) CTSUOPTL: F01C0H
(8-bit access) CTSUOPTAC: F01C0H CTSUOPTAJ: F01C1H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUOPTL	0	0	0	0	0	0	AJINTC	AJFEN

7	6	5	4	3	2	1	0
0	0	MTUCFEN	DTCLESS	0	MAJIRIMD	MCACFEN	CCOCFEN

SCACTB[3:0]				Sensor Counter Automatic Correction Table Number Setting
0	0	0	0	Accessible to sensor counter automatic correction table register 0
0	0	0	1	Accessible to sensor counter automatic correction table register 1
0	0	1	0	Accessible to sensor counter automatic correction table register 2
0	0	1	1	Accessible to sensor counter automatic correction table register 3
:				:
1	0	1	0	Accessible to sensor counter automatic correction table register 10
1	0	1	1	Accessible to sensor counter automatic correction table register 11
Other than above				Setting is invalid.

Set the number of the sensor counter automatic correction table to be accessed.
The range of values is 0 to 11.
When a value is written to the CTSUSCTACTH register, the value of these bits is automatically incremented.

AJINTC	Automatic Judgment Interrupt Control
0	Generates an INTCTSUFN interrupt when touch is detected.
1	Regardless of touch/non-touch judgment, the INTCTSUFN interrupt is generated when measurement is completed.

When CTSUCRAL.SNZ = 1 (SNOOZE function enabled) and CTSUOPTL.AJFEN = 1 (automatic judgment function enabled), set this bit to 1 to generate an INTCTSUFN interrupt at measurement completion regardless of touch/non-touch judgment.

AJFEN	Automatic Judgment Function Enable
0	Automatic judgment function is disabled
1	Automatic judgment function is enabled

Setting this bit to 1 enables the automatic judgment function.
When setting this bit to 1, also set the CTSUOPTL.CCOCFEN bit to 1 (CCO characteristics correction function enabled).

MTUCFEN	Mutual Capacitance Calculation Enable
0	Do not subtract
1	Subtract the first measurement data from the second measurement data

When CTSUCRAL.MD1 = 1 (mutual self-capacitance method) and this bit is set to 1, the value of the sensor counter (CTSUSC.SC[15:0]) is the result of subtracting the first measurement data from the second measurement data.
When CTSUCRAL.MD1 = 0 (self-capacitance method), setting this bit to 1 is prohibited.
If the subtraction results in a sensor counter value below 0000H, the value is 0000H.
If an overflow occurs in either the first or second measurement, the sensor counter overflow flag is set, but the sensor counter value is subtracted and the result is output.

Since the result is stored after the second measurements, the measurement data transfer request is not output at the end of the first measurement. Also, CTSUSRL.DTSR is not set to 1 (measurement result is not read).
When the second measurement is completed, a measurement data transfer request is output and CTSUSRL.DTSR is set to 1.

Set this bit to 1 when CTSUCRAL.MD1 = 1 and AJFEN = 1 (automatic judgment function enabled).

After the second measurement is completed, if this bit is set to 0 and the measurement data is read, the second measurement data (value without subtraction) can be read.

DTCLESS	Data Transfer Request Disable
0	Write request for each channel and data transfer request are enabled.
1	Write request for each channel and data transfer request are disabled.

When this bit is set to 1, when measurement is performed with CTSUCRAL.MD0 = 0 (single scan mode), the setting register write request and the measurement data transfer request for each channel are not output.
When CTSUCRAL.MD0 = 1 (multi-scan mode), set this bit to 0.
When this bit is set to 1, status transitions to the next without writing to the CTSUSO0, 1 registers if the CTSUSO0, 1 registers are set before measurement.
Since the measurement data transfer request signal is not output, CTSUSRL.DTSR will not be set to 1. However, even if the CTSUSC and CTSUUC registers are not read, the status transitions to the next state.

MAJIRIMD	Value Majority Mode Enable
0	Value majority mode disabled
1	Value majority mode enabled

Setting this bit to 1 enables Value Majority Mode (VMM).
Three SUCLKx clocks with different frequencies are used, and the two closest measurement values are added together and stored as the sensor counter value.[Note]

When AJFEN = 1 (automatic judgment function enabled), this added value is used to judge whether the touched or non-touched state is detected.

Setting this bit is prohibited except when CCOCFEN = 1 (CCO characteristics correction function enabled) and MCACFEN = 1 (multi-clock correction function enabled).
When this bit is set to 1, register write request per channel is output only before the first measurement.

When used with CTSUCRAL.MD0 = 0 (self-capacitance method) and CTSUOPTL.DTCLESS = 1 (write/data transfer request disabled), set CTSUMACTxL, H (x = 1, 2, 3) and CTSUSO0, 1 registers before starting measurement.

When used with CTSUCRAL.MD0 = 1 (mutual capacitance method), write CTSUMACTxL, H used for DTC transfer, and then write CTSUSO0, 1 registers.
Since the results are stored after the third measurement, no measurement data transfer request is output after the first and second measurements are completed, and CTSUSRL.DTSR is not set to 1 (the measurement result has not been read).
After the third measurement is completed, the measurement data transfer request is output and CTSUSRL.DTSR is set to 1.

When setting this bit to 1, also set the CCOCFEN and MCACFEN bits to 1.

When this bit is 0 and AJFEN = 1, majority decision is not performed, and a touch/non-touch decision is made for each measurement result, and the result is stored in the CTSUAJRRL.TJR[3:0] bits. When CTSUAJCRL.JC[1:0] = 01B (touch decision if two or more are detected), a majority decision is performed based on the judgment results, and the result is stored in the CTSUAJRRL.FJR bits (JMM: Judgment Majority Mode).

Note If the sensor counter value exceeds FFFFH as a result of adding two similar measurement results, the sensor counter value becomes FFFFH and the sensor counter overflow flag becomes 1.

MCACFEN	Multi-clock Correction Function Enable
0	Multi-clock correction function is disabled.
1	Multi-clock correction function is enabled.

Enables the multi-clock correction function for multi-clock measurement results. When this bit is set to 1, the CCOCFEN bit is also set to 1. When CCOCFEN = 0, setting this bit to 1 is prohibited .
After the measurement is completed, if this bit is set to 0 and the measurement data is read, the measurement data before correction can be read.

CCOCFEN	CCO Characteristics Correction Function Enable
0	CCO characteristics correction function is disabled.
1	CCO characteristics correction function is enabled. [Note]

When this bit is set to 1, the measured value is corrected using the sensor counter auto-correction table n [Note].
After the measurement is completed, if this bit is set to 0 and the measurement data is read, the measurement data before correction can be read.

Note If the sensor counter value before correction is below FFFFH and exceeds FFFFH due to CCO correction, the sensor counter value becomes FFFFH and the sensor counter overflow flag becomes 1.

15.2.13 CTSU Sensor Counter Automatic Correction Table Access Register (CTSUSCTACTL, CTSUSCTACTH)

The CTSUSCTACTL and CTSUSCTACTH registers can be set by 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

This register is used to access the sensor counter automatic correction table n (n = 0 to 11). The table to be accessed is selected by the CTSUOPTH.SCACTB[3:0] bits.

Writing a value to the CTSUSCTACTH register automatically increments CTSUOPTH.SCACTB[3:0] bits (table number to access).

Set these registers when CTSUCRAL.STRT = 0 (measurement operation stopped).

Figure 15-18. Format of CTSU Sensor Counter Automatic Correction Table Access Register

Address: (16-bit access only) CTSUSCTACTH: F01C6H After reset: 0000H (n = 0 to 10) FFFFH (n = 11) R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUSCTACTH	SCNTACCOUNTn[15:0]															

(n = 0 to 11; specified by CTSUOPTH.SCACTB[3:0])

Address: (16-bit access only) CTSUSCTACTL: F01C4H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUSCTACTL	SCNTACCOEFFn[15:0]															

(n = 0 to 11; specified by CTSUOPTH.SCACTB[3:0])

SCNTACCOUNTn[15:0]	Sensor Counter Measurement Value Setting
These bits set the measurement value to be compared. After the measurement of each channel is completed, if the actual measured value is between SCNTACCOUNT(n) and SCNTACCOUNT(n+1), a correction coefficient is derived for the actual measured value from the correction coefficient n, n+1 determined by SCNTACCOEFF(n) and SCNTACCOEFF(n+1), and then the corrected measured value is output to CTSUSC register.	

SCNTACCOEFFn[15:0]	Sensor Counter Correction Coefficient Setting
These bits set the correction coefficient for the measured value specified in the SCNTACCOUNTn[15:0] bits. The coefficient is a fixed-point number with 4 bits for the integer part and 12 bits for the decimal part. The setting range is 0000H to FFFFH. Maximum FFFFH setting: 15.999755859375 Minimum [resolution] 0001H setting: 2^{-12} = 0.000244140625	

The sensor counter automatic correction tables are used when the CTSUOPTL.CCOCFEN bit is set to 1 (CCO characteristics correction function is enabled).

For the sensor counter measured value in the sensor counter auto-correction table n (n = 0 to 11), set the n-th measured value to be compared with the measured value and the correction factor for the measured value. The n-th measurement should be greater than the (n-1)-th measurement. Also, the 0th measured value is fixed at 0000H and the 11th measured value is fixed at FFFFH. When writing, set the same values.

Writing to the SCNTACCOUNT[15:0] bits automatically increments the value of the CTSUOPTH.SCACTB[3:0] bits, allowing access to the next table.

15.2.14 CTSU Multi-Clock Automatic Correction Table Registers 1, 2, 3 (CTSUMACT1L, 1H, 2L, 2H, 3L, 3H)

The CTSUMACT1L, CTSUMACT1H, CTSUMACT2L, CTSUMACT2H, CTSUMACT3L, and CTSUMACT3H registers can be set by 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

The correspondence between each register and multi-clock is as follows.

CTSUMACT1L, 1H: Multi-clock MCA1 (SUCLK1)
CTSUMACT2L, 2H: Multi-clock MCA2 (SUCLK2)
CTSUMACT3L, 3H: Multi-clock MCA3 (SUCLK3)

These registers do not correspond to MCA0 (SUCLK0). The MCA0 sensor offset adjustment bits for multi-clock n (SOn) are set by CTSUSO0.SO[9:0].

Figure 15-19. Format of CTSU Multi-Clock Automatic Correction Table Registers 1, 2, 3

Address: (16-bit access only) CTSUMACT1H: F01CEH — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUMACT1H	OFFSETCOEFF1[15:0]															

Address: (16-bit access only) CTSUMACT1L: F01CCH — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUMACT1L	0	0	0	0	0	0	SO1[9:0]									

Address: (16-bit access only) CTSUMACT2H: F01D2H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUMACT2H	OFFSETCOEFF2[15:0]															

Address: (16-bit access only) CTSUMACT2L: F01D0H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUMACT2L	0	0	0	0	0	0	SO2[9:0]									

Address: (16-bit access only) CTSUMACT3H: F01D6H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUMACT3H	OFFSETCOEFF3[15:0]															

Address: (16-bit access only) CTSUMACT3L: F01D4H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUMACT3L	0	0	0	0	0	0	SO3[9:0]									

OFFSETCOEFFn[15:0]	Offset Coefficient Bit for Multi-clock n

These bits set the signed offset coefficient for multi-clock MCAn (n = 1 to 3) when CTSUOPTL.MCACFEN = 1. Multi-clock MCA0 has no dedicated bits.

The sensor counter for multi-clock measurement is corrected by the following formula.

Corrected SC[15:0] = α × Measurement SC[15:0] − OFFSETCOEFFn[15:0]
α = (SUMULTI0 + 1)/(SUMULTIn + 1): frequency coefficient (n = 1 to 3)

When CTSUCRAL.MD0 = 0 and CTSUOPTL.MAJIRIMD = 1, OFFSETCOEFFn (n = 1 to 3) can be set before measurement.

When CTSUCRAL.MD0 = 1 and CTSUOPTL.MAJIRIMD = 1, the correction value for each clock is set each time an INTCTSUWR interrupt occurs.

When CTSUCRAL.MD0 = 1 and CTSUOPTL.MAJIRIMD = 0, when an INTCTSUWR interrupt occurs, write OFFSETCOEFFn (n = 1 to 3) corresponding to CTSUSRL.MFC[1:0] = 00B to 11B to OFFSETCOEFF1.

However, OFFSETCOEFFn does not exist in CTSUSRL.MFC[1:0] = 00B, write 0000H as a dummy value to OFFSETCOEFF1.

SOn[9:0]	Sensor Offset Adjustment Bits for Multi-clock n

These bits configure the sensor offset adjustment for multi-clock MCAn (n = 1 to 3) when CTSUOPTL.MCACFEN = 1. The function is the same as CTSUSO0.SO[9:0] and corresponds to each multi-clock MCAn (n = 1 to 3).

When CTSUCRAL.MD0 = 0 and CTSUOPTL.MAJIRIMD = 1, it is possible to set SO1 to 3 before measurement and perform the measurement. MCA0 is set by CTSUSO0.SO[9:0].
When CTSUCRAL.MD0 = 1 and CTSUOPTL.MAJIRIMD = 1, write SOn (n = 1 to 3) each time an INTCTSUWR interrupt occurs.

When CTSUCRAL.MD0 = 1 and CTSUOPTL.MAJIRIMD = 0, do not write to the SOn bit when an INTCTSUWR interrupt occurs. Instead, write the offset value of each multi-clock MCAn (n = 1 to 3) to CTSUSO0.SO[9:0].

When using DTC to transfer data to these registers with an INTCTSUWR interrupt request, set DTC according to the value set in CTSUOPTL.MAJIRIMD.

(1) CTSUOPTL.MAJIRIMD = 1
CTSUCRAL.MD0 = 1 Note 1, CTSUOPTL.MCACFEN = 1, (CTSUOPTL.AJFEN = 1)

The setting register write request for each channel is output only once per channel measurement.
Set the CTSUMACTx (x = 1, 2, 3) registers according to the multi-clock to be measured as the order of the DTC chain transfer [following the CTSUAJTHR, CTSUAJMMAR, CTSUAJBLAC, CTSUAJBLAR, and CTSUAJRRL registers] Note 2, then write the CTSUSO0, and finally CTSUSO1 register. (Bits other than CTSUSO0.SO[9:0] are common.)

(2) CTSUOPTL.MAJIRIMD = 0
CTSUCRAL.MD0 = 1 Note 3, CTSUOPTL.MCACFEN = 1, (CTSUOPTL.AJFEN = 1)

A setting register write request for each channel is output for the number of times the multi-clock is enabled for each channel measurement.
As for the order of chain transfer of DTC, [following CTSUAJTHR, CTSUAJMMAR, CTSUAJBLAC, CTSUAJBLAR, and CTSUAJRRL registers] Note 2, set only the next multi-clock to be measured in the CTSUMACT1 register, then write the CTSUSO0, and finally CTSUSO1 register.
When writing to CTSUMACT1, if the first measurement results in CTSUSRL.MFC[1:0] = 00B, write 0000H to the OFFSETCOEFFn[15:0] bits as a dummy.

Notes 1. When CTSUCRAL.MD0 = 0, setting the CTSUMACTx (x = 1, 2, 3) and CTSUSO registers before measurement eliminates the need for DTC transfers (CTSUOPTL.DTCLESS = 1 is possible). However, bits other than CTSUSO0.SO[9:0] (CTSUSO1.SDPA[7:0], CTSUSO1.SSDIV[3:0], CTSUSO0.SNUM[5:0]) are common settings.

2. If CTSUOPTL.AJFEN = 0, data transfer in [ ] is not required.

3. When CTSUCRAL.MD0=0, if only one of MCAn (n = 1 to 3) is selected, setting one of CTSUMACTx (x = 1, 2, 3) and the CTSUSO0, CTSUSO1 registers before measurement, no DTC transfer is required (CTSUOPTL. DTCLESS = 1 is possible).

15.2.15 CTSU Automatic Judgment Control Register (CTSUAJCRL, CTSUAJCRH)

The CTSUAJCRL and CTSUAJCRH registers can be set by 8-bit or 16-bit memory manipulation instruction.
Reset signal generation clears these registers to 0001H (upper 16-bit) and 0000H (lower 16-bit).
Set these registers when CTSUCRAL.STRT = 0 (measurement operation stopped).

Figure 15-20. Format of CTSU Automatic Judgment Control Register

Address: (16-bit access) CTSUAJCRH: F01DAH
(8-bit access) CTSUAJCR2: F01DAH CTSUAJCR3: F01DBH After reset: 0001H R/W

Symbol	15	14	13	12	11	10	9	8
CTSUAJCRH	AJBMAT[3:0]				AJMMAT[3:0]			

7	6	5	4	3	2	1	0
0	0	JC[1:0]		0	0	0	BLINI

Address: (16-bit access) CTSUAJCRL: F01D8H
(8-bit access) CTSUAJCR0: F01D8H CTSUAJCR1: F01D9H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
CTSUAJCRL	THOT[7:0]							

7	6	5	4	3	2	1	0
TLOT[7:0]							

AJBMAT[3:0]	Baseline Average Number Setting

Each time the number of measurements set by these bits is completed, the baseline average result (CTSUAJBLARH.AJBLAR[15:0]) is updated.

AJBMAT[3:0]				Baseline Average Number
0	0	0	0	Stop baseline update
0	0	0	1	2^{1+1} (= 4 times)
0	0	1	0	2^{2+1} (= 8 times)
:				:
1	1	1	0	2^{14+1} (= 32768 times)
1	1	1	1	2^{15+1} (= 65536 times)

Set the baseline average number to exceed the measured value moving average number specified by AJMMAT[3:0] bits.

When this bit is changed and measurement is started, each register status is as follows.

- CTSUAJBLARL.AJBLAC[15:0] is set to 0000H.
- CTSUAJBLACH.AJBLACT[31:16] is set to 0000H.
- CTSUAJBLACL.AJBLACT[15:0] is set to 0000H.
- CTSUAJBLARH.AJBLAR[15:0] is retained.

AJMMAT[3:0]	Measured Value Moving Average Number Setting

These bits specify the number of moving averages of the measured value.

AJMMAT[3:0]				Measured Value Moving Average Number
0	0	0	0	2^0 (1 time)
0	0	0	1	2^1 (2 times)
0	0	1	0	2^2 (4 times)
:				:
1	0	1	0	2^{10} (1024 times)
1	0	1	1	2^{11} (2048 times)
Other than above				Setting prohibited

JC[1:0]	Judgment Condition Setting

These bits can only be set when performing multi-clock measurement. Set these bits to 00B when not performing multi-clock measurement.

When automatic judgment is performed with multi-clock measurement, the final judgment result bit (CTSUAJRRL.FJR) is updated according to the threshold judgment result (CTSUAJRRL.TJR[3:0]) for each multiple clock.

JC[1:0]		Judgment Condition
0	0	Judges as “touch” when there is one or more touch judgments.
0	1	Judges as “touch” when there are two or more touch judgments.
1	0	Judges as “touch” when there are three or more touch judgments
1	1	Judges as “touch” only when there are four touch judgments.

If the setting conditions do not match, the final result will be a non-touch judgment.
If more touch judgments are selected than the number of multi-clock measurements, the final result will always be a non-touch judgment.

When CTSUOPTL.MAJIRIMD = 1 (Value majority mode: VMM enabled) and CTSUOPTL.AJFEN = 1 (automatic judgment function enabled), set JC[1:0] to 00B.
In this case, the multi-clock measurement result is determined by adding the two closest measurement results, and the result is stored in CTSUAJRRL.TJR0.

BLINI	Baseline Initialization
0	Perform baseline operations.
1	Initialize the results of baseline operations.

To initialize the baseline calculation results, it is necessary to perform measurement with this bit set to 1.
When this bit is set to 1 and measurement is performed, each register status is as follows.

- CTSUAJBLARL.AJBLAC[15:0] is set to 0000H.
- CTSUAJBLACH.AJBLACT[31:16] is set to 0000H.
- CTSUAJBLACL.AJBLACT[15:0] is set to 0000H.
- CTSUAJBLARH.AJBLAR[15:0] is set to the value of CTSUAJMMARH.AJMMAR[26:11].

For the first judgment after reset, be sure to make the measurement with this bit set to 1.
If measurement is performed without setting 1, all subsequent judgment results are invalid.
To start a baseline average value calculation, clear the value to 0 after the measurement is completed.

THOT[7:0]	Touch Judgment Criterion Setting

These bits specify the number of consecutive “touch” detections before a “touch” is reported.

0	0	0	0	0	0	0	0	One judgment exceeding the threshold H
0	0	0	0	0	0	0	1	2 consecutive judgments exceeding the threshold H
:								:
1	1	1	1	1	1	1	0	255 consecutive judgments exceeding the threshold H
1	1	1	1	1	1	1	1	256 consecutive judgments exceeding the threshold H

When CTSUAJRRL.TJRn is 0 (non-touch), TJRn becomes 1 (touch detected) by touch judgments for the set number of consecutive judgments (n is determined by the MCAn settings).

Do not rewrite this bit after the start of automatic judgment measurement.

The number of consecutive judgments exceeding the threshold H can be checked by the same judgment consecutive count result (CTSUAJRRL.SJCCR[7:0]).

TLOT[7:0]	Non-Touch Judgment Criterion Setting

These bits specify the number of consecutive “non-touch” detections before a “non-touch” is reported.

0	0	0	0	0	0	0	0	One judgment exceeding the threshold L
0	0	0	0	0	0	0	1	2 consecutive judgments exceeding the threshold L
:								:
1	1	1	1	1	1	1	0	255 consecutive judgments exceeding the threshold L
1	1	1	1	1	1	1	1	256 consecutive judgments exceeding the threshold L

When CTSUAJRRL.TJRn is 1 (touch detected), TJRn becomes 0 (non-touch) by non-touch judgments for the set number of consecutive judgments (n is determined by the MCAn settings).

Do not rewrite this bit after the start of automatic judgment measurement.

The number of consecutive judgments exceeding the threshold L can be checked by the same judgment consecutive count result (CTSUAJRRL.SJCCR[7:0]).

15.2.16 CTSU Threshold Register (CTSUAJTHRL, CTSUAJTHRH)

The CTSUAJTHRL and CTSUAJTHRH registers can be set by 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

When using CTSUCRAL.MD0 = 1 (multi-scan mode), read this register and save it to RAM when an INTCTSURD interrupt occurs. When the INTCTSUWR interrupt occurs when measuring the same channel again, write back the read value. If a value other than the read value is written back, operation is not guaranteed.

Set these registers when the measurement state is State 0 or State 2.

Figure 15-21. Format of CTSU Threshold Register

Address: (16-bit access only) CTSUAJTHRH: F01DEH — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUAJTHRH	AJTHH[15:0]															

Address: (16-bit access only) CTSUAJTHRL: F01DCH — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUAJTHRL	AJTHL[15:0]															

AJTHH[15:0]	Upper Threshold Setting

Set the touch judgment threshold as a relative value from the baseline.
The value is a signed integer between −32768 and 32767 Note 1.
The judgment differs between the plus setting and the minus setting Note 2.

- Plus setting: 7FFFH to 0000H
  When the moving average calculation result of the sensor counter measurement value becomes larger than the baseline plus this setting value, it is judged that the threshold value H has been exceeded.
- Minus setting: 8000H to FFFFH
  When the moving average calculation result of the sensor counter measurement value becomes smaller than the baseline plus this setting value, it is judged that the threshold value H has been exceeded.

AJTHL[15:0]	Lower Threshold Setting

Set the non-touch judgment threshold as a relative value from the baseline.
The value is a signed integer between −32768 and 32767 Note 1.
The judgment differs between the plus setting and the minus setting Note 2.

- Plus setting: 7FFFH to 0000H
  When the moving average calculation result of the sensor counter measurement value becomes smaller than the baseline plus this setting value, it is judged that the threshold value L has been exceeded.
- Minus setting: 8000H to FFFFH
  When the moving average calculation result of the sensor counter measurement value becomes larger than the baseline plus this setting value, it is judged that the threshold value L has been exceeded.

Notes 1. When CTSUOPTL.MAJIRIMD = 1, the baseline is determined by adding the value of the two approximate multi-clock measurements results from the three multi-clock measurements.
When CTSUOPTL.MTUCFEN = 1, the baseline is determined by subtracting the first measurement data from the second measurement data. Set the appropriate threshold from the baseline at each setting.

2. The threshold settings should be set to the same polarity for both the H and L settings.
The values should be set so that the relationship is |AJTHHn[15:0]| > |AJTHLn[15:0]|.
In the case of a minus setting, the value will be |8000H| > |FFFFH|.

15.2.17 CTSU Moving Average Result Register (CTSUAJMMARL, CTSUAJMMARH)

The CTSUAJMMARL and CTSUAJMMARH registers can be set by 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

When using CTSUCRAL.MD0 = 1 (multi-scan mode), read this register and save it to RAM when an INTCTSURD interrupt occurs. When the INTCTSUWR interrupt occurs when measuring the same channel again, write back the read value. If a value other than the read value is written back, operation is not guaranteed.

Set these registers when the measurement state is State 0 or State 2.

Figure 15-22. Format of CTSU Moving Average Result Register

Address: (16-bit access only) CTSUAJMMARH: F01E2H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUAJMMARH	AJMMAR[26:11]															

Address: (16-bit access only) CTSUAJMMARL: F01E0H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8
CTSUAJMMARL	AJMMAR[10:3]							

7	6	5	4	3	2	1	0
AJMMAR[2:0]			0	AJMMATI[3:0]			

AJMMR[26:0]	Measured Value Moving Average Result
These bits store the moving average calculation result of the measured value. AJMMAR[26:11] represents the integer part and AJMMAR[10:0] represents the decimal part. Only the integer part is compared to the baseline average result. When AJMMATI[3:0] is the initial value 0000B, the first moving average calculation is not performed and the measured value is stored.	

AJMMATI[3:0]	Measured Value Moving Average Count
These bits indicate the number of moving averages for the current measurement. Set 0000B at the first measurement. If the initial value is set, the value is incremented by +0001B each time measurement is performed, and stops when it reaches the same value as CTSUAJCRH.AJMMAT[3:0]. After that, if CTSUAJCRH.AJMMAT[3:0] is changed to a larger than the value of these bits, +0001B is performed, and if it is changed to a smaller value, these bits are updated.	

AJMMATI[3:0]				Measured Value Moving Average Count
0	0	0	0	2^0 (1 time)
0	0	0	1	2^1 (2 times)
0	0	1	0	2^2 (4 times)
:				:
1	0	1	0	2^{10} (1024 times)
1	0	1	1	2^{11} (2048 times)
Other than above				Setting prohibited

15.2.18 CTSU Baseline Average Calculation Register (CTSUAJBLACL, CTSUAJBLACH)

The CTSUAJBLACL and CTSUAJBLACH registers can be set by 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

When using CTSUCRAL.MD0 = 1 (multi-scan mode), read this register and save it to RAM when an INTCTSURD interrupt occurs. When the INTCTSUWR interrupt occurs when measuring the same channel again, write back the read value. If a value other than the read value is written back, operation is not guaranteed.

Set these registers when the measurement state is State 0 or State 2.

Figure 15-23. Format of CTSU Baseline Average Calculation Register

Address: (16-bit access only) CTSUAJBLACH: F01E6H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUAJBLACH	AJBLACT[31:16]															

Address: (16-bit access only) CTSUAJBLACL: F01E4H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUAJBLACL	AJBLACT[15:0]															

AJBLACT[31:0]	Auto-judgment Baseline Average Calculation Value
These bits store the value during the calculation to obtain the baseline average result. AJBLACT[31:16] represents the integer part and AJBLACT[15:0] represents the decimal part. If CTSUAJCRH.BLINI = 1 or CTSUAJRRL.TJRn = 1 (n: target multi-clock) or CTSUAJCRH.AJBMAT[3:0] is updated, these bits will be 0000000H in the next measurement. If CTSUAJCRH.BLINI = 0 and CTSUAJRRL.TJRn = 0 (n: target multi-clock) and CTSUAJCRH.AJBMAT[3:0] has not been updated, the baseline calculation begins.	

15.2.19 CTSU Baseline Average Result Register (CTSUAJBLARL, CTSUAJBLARH)

The CTSUAJBLARL and CTSUAJBLARH registers can be set by 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

When using CTSUCRAL.MD0 = 1 (multi-scan mode), read this register and save it to RAM when an INTCTSURD interrupt occurs. When the INTCTSUWR interrupt occurs when measuring the same channel again, write back the read value. If a value other than the read value is written back, operation is not guaranteed.

Set these registers when the measurement state is State 0 or State 2.

Figure 15-24. Format of CTSU Baseline Average Result Register

Address: (16-bit access only) CTSUAJBLARH: F01EAH — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUAJBLARH	AJBLAR[15:0]															

Address: (16-bit access only) CTSUAJBLARL: F01E8H — After reset: 0000H — R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUAJBLARL	AJBLAC[15:0]															

AJBLAR[15:0]	Baseline Average Result
These bits store the baseline average calculation result. Depending on the CTSUAJCRH.BLINI setting, subsequent measurement operations differ. • CTSUAJCRH.BLINI = 1: Contains the value of CTSUAJMMARH.AJMMAR[26:11]. • CTSUAJCRH.BLINI = 0: The value of these bits is updated each time the number of measurements specified as the baseline average count is reached.	

AJBLAC[15:0]	Baseline Average Count
These bits indicate the current count value of the baseline average calculation. The count value range is from 0000H to FFFFH. When CTSUAJCRH.BLINI = 1 or CTSUAJRRL.TJRn = 1 (n: target multi-clock) or CTSUAJCRH.AJBMAT[3:0] is updated, these bits will be 0000H in the next measurement. If CTSUAJCRH.BLINI = 0 and CTSUAJRRL.TJRn = 0 (n: target multi-clock) and CTSUAJCRH.AJBMAT[3:0] has not been updated, start counting the baseline average.	

15.2.20 CTSU Automatic Judgment Result Register (CTSUAJRRL)

The CTSUAJRRL register can be set by 8-bit or 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

When using CTSUCRAL.MD0 = 1 (multi-scan mode), read this register and save it to RAM when an INTCTSURD interrupt occurs. When the INTCTSUWR interrupt occurs when measuring the same channel again, write back the read value. If a value other than the read value is written back, operation is not guaranteed.

Set this register when the measurement status is State 0 or State 2.

Figure 15-25. Format of CTSU Automatic Judgment Result Register

Address: (16-bit access) CTSUAJRRL: F01ECH
(8-bit access) CTSUAJRR0: F01ECH CTSUAJRR1: F01EDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
CTSUAJRRL	SJCCR[7:0]							

7	6	5	4	3	2	1	0
0	0	0	FJR	TJR3	TJR2	TJR1	TJR0

SJCCR[7:0]	Remaining Number of Consecutive Detections

These bits indicate how many more consecutive touch or non-touch detections will invert the value of the TJRn flag.

If the TJRn is 0 (non-touch state), this indicates the remaining required number of consecutive times after loading CTSUAJCRL.THOT[7:0] in the initial judgment Note and subtracting the number of consecutive times that the threshold H is exceeded thereafter.

If the TJRn is 1 (touch state), this indicates the remaining required number of consecutive times after loading CTSUAJCRL.TLOT[7:0] in the initial judgment Note and subtracting the number of consecutive times that the threshold L is exceeded.

11111111B: Remaining 256 consecutive “Threshold H exceeded judgment” or “Threshold L exceeded judgment” is required.

11111110B: Remaining 255 consecutive “Threshold H exceeded judgment” or “Threshold L exceeded judgment” is required.

:

00000001B: Remaining 2 consecutive “Threshold H exceeded judgment” or “Threshold L exceeded judgment” is required.

00000000B: Remaining 1 consecutive “Threshold H exceeded judgment” or “Threshold L exceeded judgment” is required.

If the threshold judgment result is not consecutive, CTSUAJCRL.THOT[7:0] or CTSUAJCRL.TLOT[7:0] bits are saved again.

Note Initial judgment means the first judgment with CTSUAJCRH.BLINI = 1 after reset release.

FJR	Final Judgment Result
0	Non-touch
1	Touch

This bit stores the final judgment result.

When multi-clock measurement is used, the result of TJR0 to TJR3 bits and the result of judgment based on the condition set by CTSUAJCRH.JC[1:0] bits is stored.

If multi-clock measurement is not specified, the same value as the TJR0 bit is stored.

This bit is read only.

TJRn	Threshold Judgment Result (n = 0 to 3)
0	Non-touch (threshold L exceeded)
1	Touch (threshold H exceeded)

This bit stores the threshold judgment result.
The judgment result is determined by the following conditions.

- Threshold (H/L) set by CTSUAJTHRH.AJTHH[15:0] / CTSUAJTHRL.AJTHL[15:0] bits.
- Threshold (H/L) exceeded judgment consecutive count set by CTSUAJCRL.THOT[7:0] / CTSUAJCRL. TLOT[7:0] bits.

When using multi-clock measurement, the judgment result for CTSUMCHH.MCA[3:0] (SUCLK0 to SUCLK3) is stored in TJR[3:0] bits.
When multi-clock measurement is not specified, the judgment result is stored in the TJR0 bit.

When MAJIRIMD = 1 (voting majority mode enabled) and CTSUOPTL.AJFEN = 1 (automatic judgment function enabled), the result of the determination based on the addition of the two approximate measurements is stored in the TJR0 bit. In this case, consider CTSUAJCRH.JC[1:0] = 00B (touch judgment with one or more measurements) and make a judgment using only the TJR0 bit.

In the case of multi-clock measurement, the average value calculation of the baseline is stopped for the multi-clock that detects “touch” judgment from “non-touch”. The multi-clock that detects “non-touch” judgment from “touch” restarts the average value calculation of the baseline.

The value of this bit is cleared by measuring with CTSUAJCRH.BLINI = 1.

* Write 0 to this bit in the INTCTSUWR interrupt for the initial judgment.

15.2.21 CTSU External Trigger Select Register (CTSUTRG)

The CTSUTRG register can be set by 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 15-26. Format of CTSU External Trigger Select Register

Address: (8-bit access only) CTSUTRG: F01F8H After reset: 00H R/W

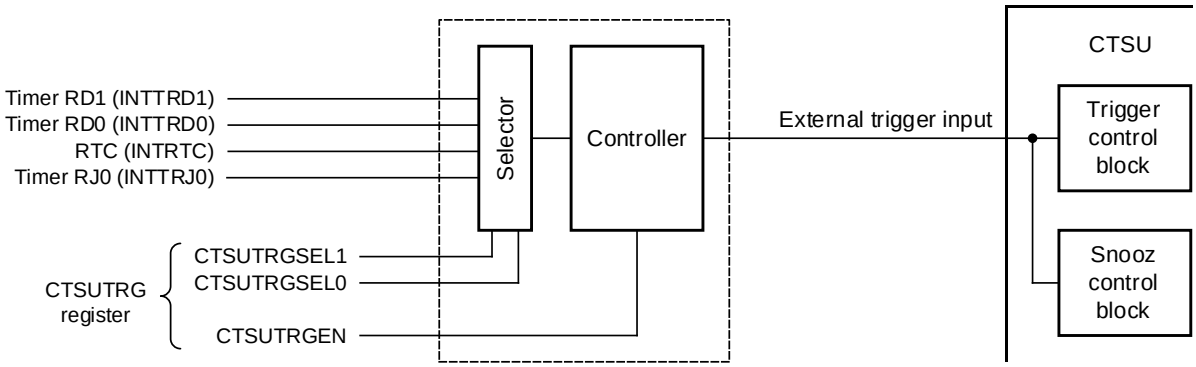
Symbol	7	6	5	4	3	2	1	0
CTSUTRG	CTSUTRGEN	0	0	0	0	0	CTSUTRGSEL[1:0]	

CTSUTRGEN	CTSU External Trigger Enabled
0	CTSU external trigger disabled.
1	CTSU external trigger enabled. When the CTSUCRAL.CAP bit is set to 1, be sure to set CTSUTRGEN bit to 1 as well. Similarly, when clear the CTSUCRAL.CAP bit, be sure to clear CTSUTRGEN bit as well.

CTSUTRGSEL[1:0] Note		CTSU External Trigger Selection
0	0	Timer RJ0 (INTTRJ0) is used for CTSU external trigger.
0	1	RTC pretimed signal/alarm match detection (INTRTC) is used for the CTSU external trigger.
1	0	Timer RD0 Input Capture B/Compare Match B (INTTRD0) is used for the CTSU external trigger.
1	1	Timer RD1 Input Capture B/Compare Match B (INTTRD1) is used for the CTSU external trigger.

Note The CTSUTRGSEL[1:0] bits should be rewritten with the CTSU external trigger disabled (CTSUTRGEN = 0).

Figure 15-27. External Trigger Selection Circuit (Outline)



Remark For the setting procedure when using the external trigger function, see **15.4.6 External Trigger.**

15.2.22 CTSU Trimming Register A (CTSUTRIM0, CTSUTRIM1)

The CTSUTRIM0 and CTSUTRIM1 registers can be accessed with 8-bit or 16-bit memory manipulation instruction.
These registers store adjustment data measured at the factory for each individual chip.
Do not rewrite these registers.

Figure 15-28. Format of CTSU Trimming Register A

Address: (16-bit access) CTSUTRIM1: F00BAH
(8-bit access) SUADJD: F00BAH TRESULT4: F00BBH After reset: Unique value for each chip R(/W)

Symbol	15	14	13	12	11	10	9	8
CTSUTRIM1	TRESULT4[7:0]							

7	6	5	4	3	2	1	0
SUADJD[7:0]							

Address: (16-bit access) CTSUTRIM0: F00B8H
(8-bit access) RTRIM: F00B8H DACTRIM: F00B9H After reset: Unique value for each chip R(/W)

Symbol	15	14	13	12	11	10	9	8
CTSUTRIM0	DACTRIM[7:0]							

7	6	5	4	3	2	1	0
RTRIM[7:0]							

TRESULT4[7:0]	Test Result 4 Storage
The RESULT4[7:0] bits hold the coefficient of variation for the 120 kΩ reference load resistance. These bits store the initial values measured at the factory. Do not modify these bits.	

SUADJD[7:0]	SUCLK Frequency Adjustment
These bits are used to adjust the SUCLK frequency in random pulse mode (CTSUCRAH.SDPSEL = 0). These bits store values that result in a SUCLK frequency of approximately 32 MHz.	

DACTRIM[7:0]	Offset Current Adjustment
These bits are used to adjust the coefficient of the DAC for offset current.	

RTRIM[7:0]	Reference Resistor Adjust Input
These bits are used to adjust the resistance value of the internal reference resistor.	

15.2.23 CTSU Trimming Register B (CTSUTRIM2, CTSUTRIM3)

The CTSUTRIM2 and CTSUTRIM3 registers can be accessed with 8-bit or 16-bit memory manipulation instruction.
These registers store adjustment data measured for each individual chip at the factory.
Do not rewrite these registers.

Figure 15-29. Format of CTSU Trimming Register B

Address: (16-bit access) CTSUTRIM3: F00BEH
(8-bit access) TRESULT2: F00BEH TRESULT3: F00BFH After reset: Unique value for each chip R(/W)

Symbol	15	14	13	12	11	10	9	8
CTSUTRIM3	TRESULT3[7:0]							

7	6	5	4	3	2	1	0
TRESULT2[7:0]							

Address: (16-bit access) CTSUTRIM2: F00BCH
(8-bit access) TRESULT0: F00BCH TRESULT1: F00BDH After reset: Unique value for each chip R(/W)

Symbol	15	14	13	12	11	10	9	8
CTSUTRIM2	TRESULT1[7:0]							

7	6	5	4	3	2	1	0
TRESULT0[7:0]							

TRESULT3[7:0]	Test Result 3
The TRESULT3[7:0] bits hold the coefficient of variation for the 60 kΩ reference load resistance. These bits store the initial values measured at the factory. Do not modify these bits.	

TRESULT2[7:0]	Test Result 2
The TRESULT2[7:0] bits hold the coefficient of variation for the 30 kΩ reference load resistance. These bits store the initial values measured at the factory. Do not modify these bits.	

TRESULT1[7:0]	Test Result 1
The TRESULT1[7:0] bits hold the coefficient of variation for the 15 kΩ reference load resistance. These bits store the initial values measured at the factory. Do not modify these bits.	

TRESULT0[7:0]	Test Result 0
The TRESULT0[7:0] bits hold the coefficient of variation for the 7.5 kΩ reference load resistance. These bits store the initial values measured at the factory. Do not modify these bits.	

15.3 Operation

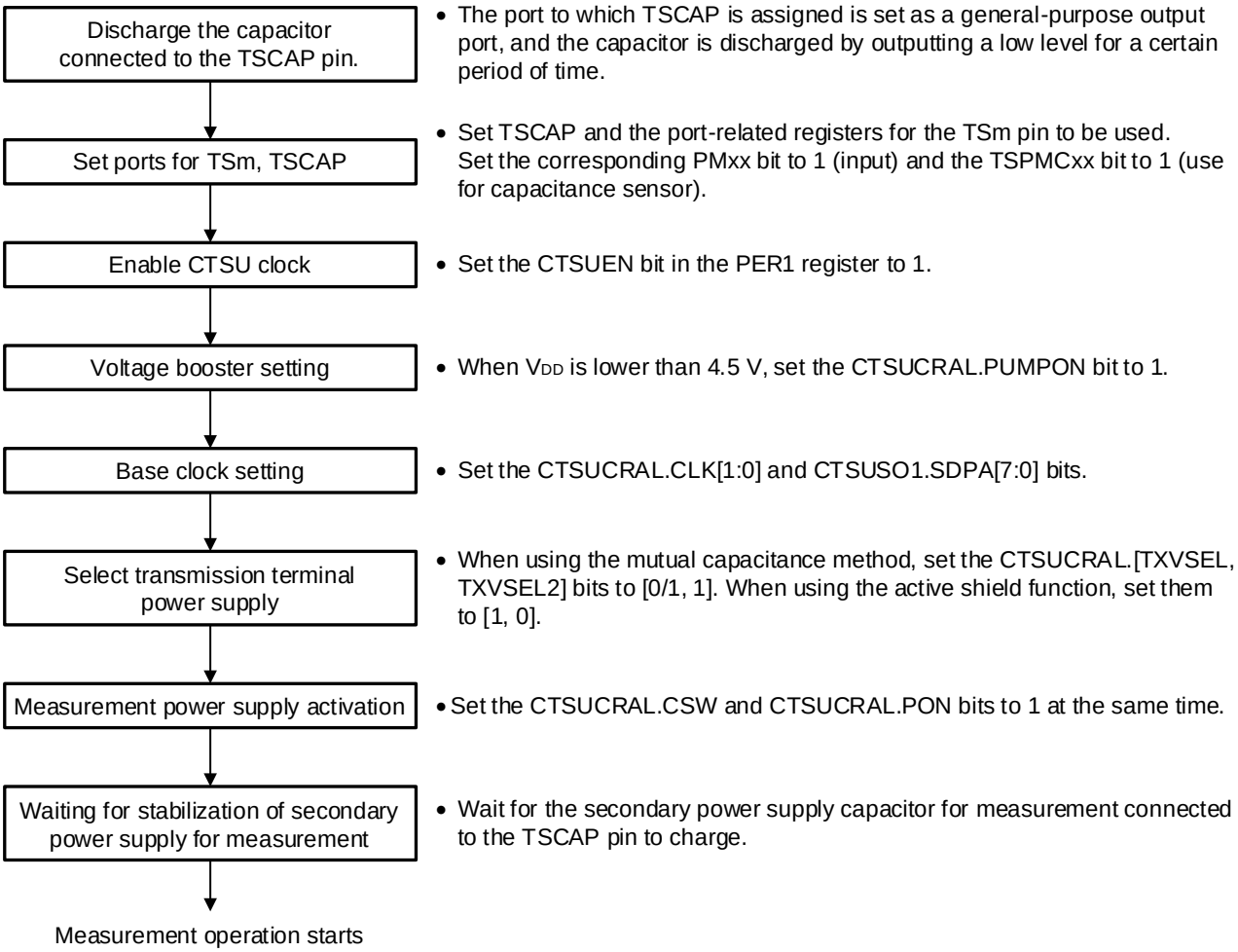
15.3.1 Principles of Measurement Operation

CTSU generates a clock by supplying the controlled current, which is proportional to the amount of current flowing through the switched capacitor, from the circuit that generates the TSCAP power supply (LDO) to the current controlled oscillator (CCO). The counter measures the clock frequency that changes when the human body approaches, and judges touch/non-touch. (See **Figure 15-3. CTSU Block Diagram.**)

15.3.2 Initial Setting Flowchart

Figure 15-30 shows the CTSU initial setting flowchart.

Figure 15-30. CTSU Initial Setting Flowchart



15.3.3 Measurement Modes

The CTSU supports self-capacitance and mutual capacitance methods.

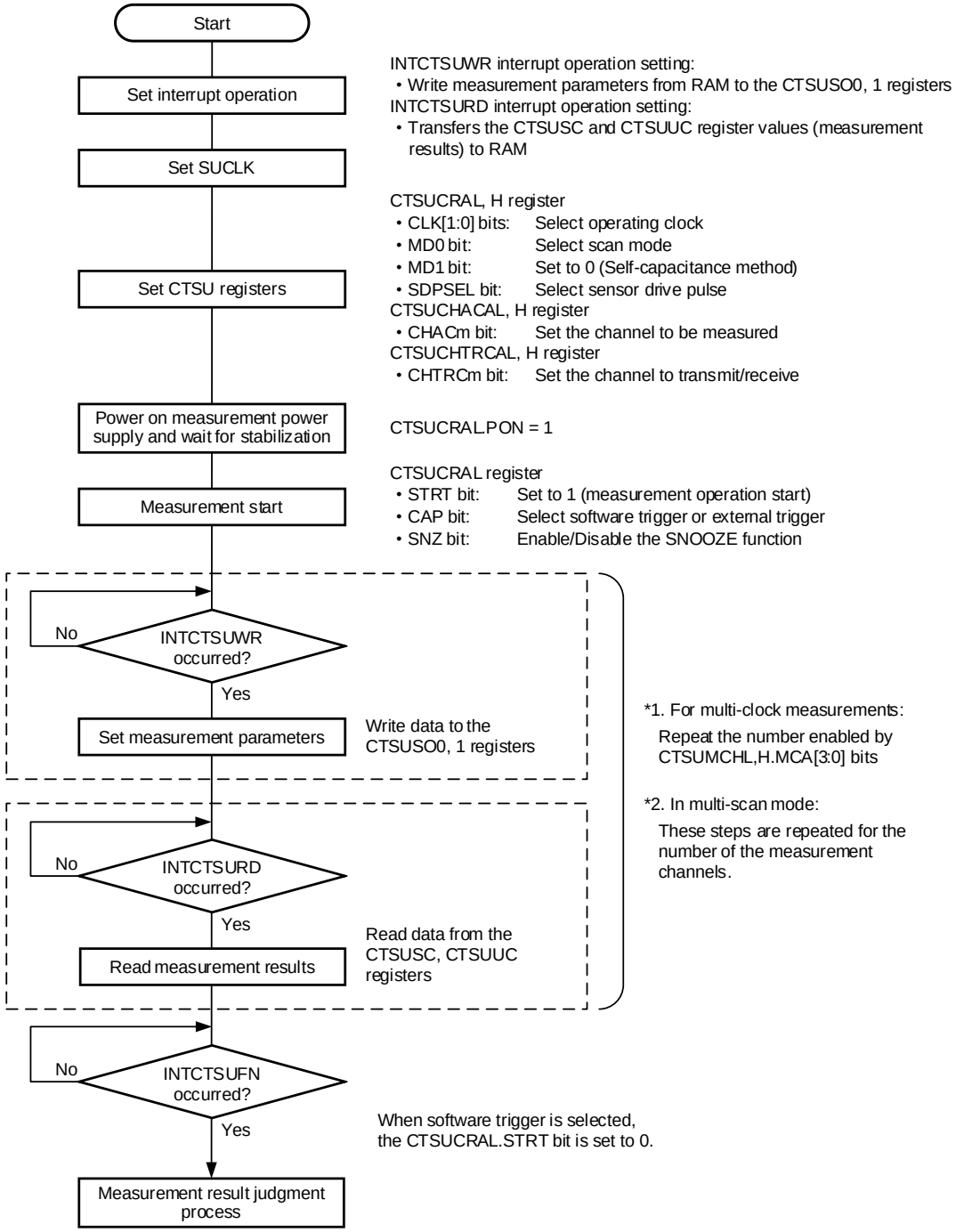
In the self-capacitance method, one touch pin (TSm) is assigned to each touch key, and the capacitance due to the proximity of the human body is measured for each.

The mutual capacitance method measures the capacitance between two electrode pairs (transmitter and receiver pins).

(1) Self-Capacitance Method Operation

Figure 15-31 shows the control flow for self-capacitance method.

Figure 15-31. Control Flow for Self-Capacitance Method

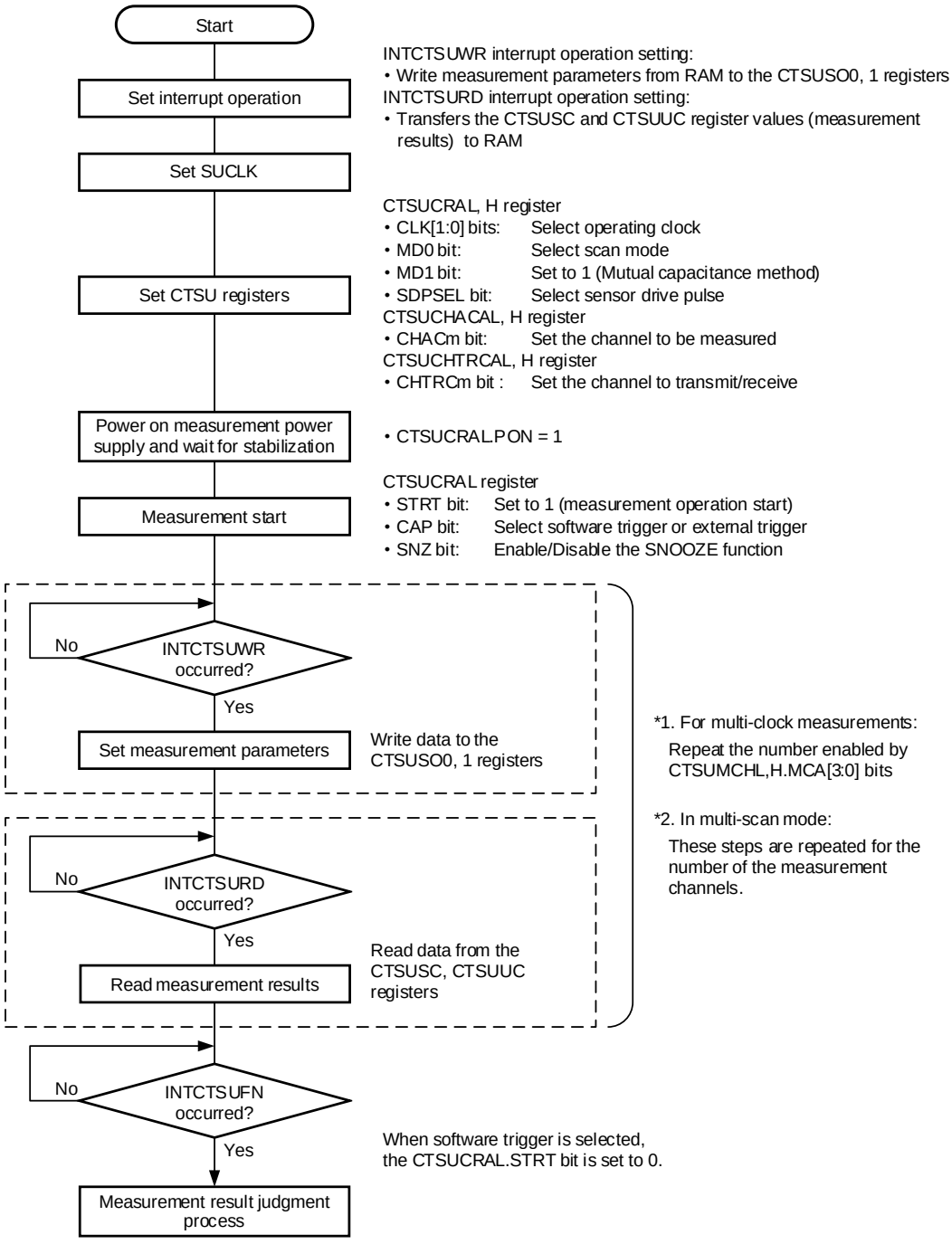


(2) Mutual Capacitance Method Operation

In mutual capacitance method, measurement is performed during the high period of the sensor drive pulse on the receive channel by applying the edge to the target transmit channel to be measured. A single measurement target is measured twice, at the rising and falling edges. The difference between the data of these two measurements is used to determine whether or not the electrode is touched, thus achieving higher touch sensitivity. Measures capacitance at all combinations of the set transmit and receive pins.

Figure 15-32 shows the control flow for mutual capacitance method.

Figure 15-32. Control Flow for Mutual Capacitance Method



15.3.4 Scan Mode

The CTSU has two scan modes: single scan mode and multi-scan mode. The scan mode is selected by the CTSUCRAL.MD0 bit.

(1) Single Scan Mode

Measurement is performed only once using the combination of the receive channel specified by the MCH0[5:0] bits and the transmit channel specified by the MCH1[5:0] bits of the CTSUMCHL, H register.

(2) Multi-Scan Mode

Among the channels set as measurement targets in the CTSUCHACAL, H register, measurements are performed once for each combination of receive and transmit channels selected in the CTSUCHTRCAL, H register.

Both receive and transmit channels are selected in ascending order of channel number. When measurements have been completed for all combinations of receiving channels corresponding to one transmitting channel, the transmitting channel will be switched.

15.3.5 Multi-clock Measurement

In multi-clock measurement, multiple clocks with different frequencies are sequentially switched. Multi-clock measurement is valid only in high resolution pulse mode (CTSUCRAH.SDPSEL = 1).

Specify the clock (SUCLKx) used for measurement using the CTSUMCHH.MCAx bits (x = 0 to 3). The frequency of each clock can be set with the CTSUSUCLKx registers.

Measurements are performed in ascending order starting from SUCLK0. When measurement of all specified clocks is completed, measurement of the next channel stars.

The majority mode is determined by measuring at three different frequencies and comparing the results of each of the three. There are two majority modes: JMM (Judgment Majority Mode) and VMM (Value Majority Mode).

- JMM: Each measurement result is judged as touch or non-touch, and touch/non-touch decision is based on the majority of the results.
- VMM: The two closest values of the three measurement results are added together to determine touch or non-touch.

JMM and VMM are set by the MAJIRIMD bit in the CTSUOPTL register.

15.3.6 Automatic Judgment Function

The CTSU has a function to automatically judge touch/non-touch.

The INTCTSUFN interrupt of automatic detection function can be used to return from the SNOOZE mode to normal mode.

The measurement value that is the target of automatic judgment is the value of CTSUSC.SC[15:0] bits. The moving average and baseline average are calculated from the measured values, and touch/non-touch is judged from the difference. The moving average and baseline average counts are set by AJMMAT[3:0] bits and AJBMAT[3:0] bits in the CTSUAJCRH register, respectively.

Each operation of the automatic judgment function is described below.

(1) Touch Judgment

When the difference between the moving average and the baseline average exceeds the threshold H (CTSUAJTHRH. AJTHH[15:0] bits) a number of consecutive times specified by the CTSUAJCRL.THOT [7:0] bits (or continuously falls below the threshold if AJTHH[15:0] is set to negative), the CTSUAJRRL.TJRn bit (n = 0 to 3) corresponding to SUCLKn is set to 1 (touch detection).

When multi-clock measurement is specified, the CTSUAJRRL.FJR bit is set to 1 (touch judgment) when the value of the TJR0 to TJR3 bits matches the judgment condition specified by the CTSUAJCRH.JC[1:0] bit.

When multi-clock measurement is not specified, the value of the TJR0 bit is the value of the FJR bit as it is.

(2) Non-Touch Judgment

When the difference between the moving average and the baseline average falls below the threshold L (CTSUAJTHRL.AJTHL[15:0] bits) a number of consecutive times specified by CTSUAJCRL.TLOT[7:0] bits (or exceeds AJTHL[15:0] continuously if AJTHL[15:0] is set to negative), the CTSUAJRRL.TJRn bit (n = 0 to 3) corresponding to SUCLKn is set to 0 (non-touch judgment).

When multi-clock measurement is specified, the CTSUAJRRL.FJR bit is set to 0 (non-touch judgment) if the value of the TJR0-TJR3 bits does not satisfy the judgment condition specified by the CTSUAJCRH.JC[1:0] bit.

When multi-clock measurement is not specified, the value of the TJR0 bit is the value of the FJR bit as it is.

(3) Baseline Average Calculation

When the CTSUAJRRL.TJRn bit (n = 0 to 3) is set to 1 (touch judgment), the baseline average calculation is stopped from the next measurement. When the baseline average calculation stops, the value during the baseline average calculation (CTSUAJBLACL, H register) and the value of the baseline average count (CTSUAJBLARL.AJBLAC[15:0] bits) are initialized.

When the TJRn bit is set to 0 (non-touch), the baseline average calculation restarts from the next measurement.

When multi-clock measurement is specified, the baseline average calculation is stopped/restarted for each bit of TJR0 to TJR3 corresponding to SUCLKn.

15.3.7 Multiple Electrode Connection Function

The Multiple Electrode Connection (MEC) function treats multiple electrodes as a single electrode. This function allows multiple electrodes to be measured at once.

When CTSUCRAL.MD1 = 0 (self-capacitance method), CTSUCRAL.MD0 = 0 (single scan mode), CTSUDBGR0.TSOD = 1 and CTSUDBGR0.IOCSEL = 0 (multiple electrode connection function), sensor drive pulses are output from all TSm pins for which the CTSUCHACAL/H.CHACm bit (m = 0 to 30) is set to 1 (measurement target) and the CTSUCHTRCAL/H.CHTRCm bit is set to 0 (receive). By using this function, the total capacitance of all electrodes connected to the selected TSm pin can be measured.

When “touch” is detected using this function, it is not possible to determine which electrode’s capacitance has changed.

To determine which electrodes have been touched, set the TSOD bit to 0 and the IOCSEL bit to 0 (capacitance measurement mode) and measure the capacitance of each electrode.

15.3.8 Interrupts

The CTSU supports the following three interrupts:

- CTSU setting register write request interrupt for each channel (INTCTSUWR)
- CTSU measurement data transfer request interrupt (INTCTSURD)
- CTSU measurement end interrupt (INTCTSUFN)

(1) INTCTSUWR Interrupt

Store the setting data for each measurement channel in RAM, and the setting data is written to the CTSU register using an interrupt or DTC. The INTCTSUWR interrupt occurs when the measurement status transitions from State 1 to State 2.

For this interrupt request, write setting data to the CTSUSO0, 1 registers. When data is written to the CTSUSO1 register, the measurement status transitions to State 3.

(2) INTCTSURD Interrupt

The INTCTSURD interrupt occurs when measurement on one channel ends and the measurement status transitions from State 5 to State 1.

Read the measurement result (sensor counter value) from the CTSUSC register.

If it is necessary to read the CTSUUC register, set CTSUDBGR0.CNTRDSEL = 1 (transition to next state after two reads).

When the CTSUSC and CTSUUC registers are read the number of times specified by the CNTRDSEL bit, the measurement status transitions to State 0 or State 2.

(3) INTCTSUFN Interrupt

The INTCTSUFN interrupt occurs when the measurement of all specified channels is completed and the measurement status transitions from State 1 to State 0.

When CTSUCRAL.SNZ = 1 (SNOOZE function enabled), CTSUOPTL.AJFEN = 1 (automatic detection function enabled), and CTSUOPTL.AJINTC = 0 (INTCTSUFN interrupt is generated when a touch is detected), the INTCTSUFN interrupt is generated when at least one of the following conditions is satisfied.

- When there is a channel that has been judged as “touch”.
- When a voltage error is detected (CTSUSRL.ICOMP0 = 1).

When using the INTCTSUFN interrupt as a cause for returning from SNOOZE mode, if the above conditions are not met, the device will return to standby mode.

If the measurement result is to be judged by software at the completion of measurement, set AJINTC = 1 (regardless of touch/non-touch judgment, the INTCTSUFN interrupt is generated when measurement is completed.).

15.4 Usage Notes

15.4.1 Evaluation of Detection Operation by Capacitance Sensing Unit (CTSU2SLa)

In the final process of product development, the system should be operated under conditions close to those at the time of product shipment to determine the validity of the touch sensor detection results. For evaluation, “QE for Capacitive Touch” (development support tool for capacitive touch sensors) is available. Monitor and fully evaluate the capacitance measurement status. If the results are not as expected, use “QE for Capacitive Touch” to adjust CapTouch parameters (mainly touch threshold) and re-evaluate.

Especially when CTSU is used for mutual capacitive touch, the output voltage of the transmitting electrode may fluctuate depending on the port output status, causing the capacitance measurement (count value) to change. If this occurs, adjust the touch threshold value in “QE for Capacitive Touch” and set the threshold value to account for the count value fluctuation. Such fluctuations do not occur with the self-capacitance method.

15.4.2 Port Settings When Using CTSU

When using CTSU, set the relevant registers on the port assigned the TSm (m = 0 to 30) function and the port assigned the TSCAP (P12) used in the system to the input port (corresponding PMxx bit set to 1) and the port assigned the capacitive sensor (corresponding TSPMCxx bits set to 1).

15.4.3 TSCAP Pin

A capacitor must be connected to the TSCAP pin to stabilize the internal voltage of the CTSU. The wiring between the TSCAP pin and the capacitor and between the capacitor and GND should be as thick and short as possible.

The conditions for the capacitor connected to the TSCAP pin are shown in the table below.

Item	Symbol	Conditions
External capacitance for stabilizing connected to TSCAP pin	C_{TSCAP}	10 nF ± 10% **Note**

Note Use a multilayer ceramic capacitor with a nominal capacitance of 10nF, for which the sum of the capacitance tolerance and change in the capacitance under the usage conditions will be no greater than ±10%.

The capacitor connected to the TSCAP pin should be fully discharged by outputting a low level from the port to which the TSCAP function is assigned before setting CTSUCRAL.CSW = 1 (turn on the external capacitance connection switch).

15.4.4 Changing Clock and Port Settings During Measurement Operation

Do not change or stop the peripheral hardware clock during measurement operation (CTSUCRAL.STRT = 1). And, do not change the port setting of the measurement pins (TSm/TSCAP).

To change, perform a forced stop of CTSU (write CTSUCRAL.INIT = 1 and STRT = 0 simultaneously). To re-measure CTSU, initialize it by using either of the following procedures.

- Write PER1.CTSUEN = 0, then write CTSUEN = 1 to initialize.
- Initialize after writing CTSUCRAL.[PON, CSW] = 00B, CTSUCRAL.SNZ = 0

15.4.5 Software Trigger

When 10B (f_{CLK}/4) or 11B (f_{CLK}/8) is selected by CTSUCRAL.CLK[1:0] bits, to resume measurement immediately after the end of measurement, write 1 to CTSUCRAL.STRT bit after the following time elapse after the interrupt occurs.

- When CLK[1:0] = 10B (f_{CLK}/4): more than 3 cycles
- When CLK[1:0] = 11B (f_{CLK}/8): more than 7 cycles

15.4.6 External Trigger

If an external trigger is input during measurement, the trigger is ignored. The next external event becomes effective one cycle after the operating clock when INTCTSUFN occurs.

To cancel the external trigger mode (forced stop), write 0 in the CTSUCRAL.STRT bit and write 1 in the CTSUCRAL.INIT bit simultaneously.

When using an external trigger, the following setting procedure should be used.

<1> Perform the initial settings of CTSU.
<2> With CTSUCRAL.STRT = 0 (measurement operation stopped) and CTSUTRG.CTSUTRGEN = 0 (external trigger disabled), select the external trigger factor using CTSUTRG.CTSUTRGSEL[1:0] bits.
<3> Set CTSUCRAL.CAP = 1 (external trigger) and CTSUTRG.CTSUTRGEN = 1 (external trigger enabled) , then set CTSUCRAL.STRT = 1 (measurement operation starts).
<4> Start the operation of the peripheral function that generates the external trigger factor.

15.4.7 Reading the CTSU Sensor Counter Register (CTSUSC, CTSUUC)

Do not read the CTSU sensor counter registers (CTSUSC, CTSUUC) during measurement (measurement status is State 3 to State 5). If they are read, their values cannot be guaranteed.

15.4.8 Forced Stop

To force a stop during measurement, write 0 to the CTSUCRAL.STRT bit and 1 to the CTSUCRAL.INIT bit simultaneously.

Writing 1 to the INIT bit initializes the following registers in addition to initializing the internal measurement state.

- CTSUSC and CTSUUC registers
- CTSUMCHL, H registers
- CTSUSRL register (except ICOMP0 and ICOMP1 bits)

When forced stop is performed, an interrupt request may be generated depending on the internal status. At the timing when a forced stop is requested, stop or disable processing of interrupts and DTCs is performed.

If the CTSU was controlled by DTC, when stopping DTC transfer, perform a forced stop of the CTSU as well and perform initialization processing.

15.4.9 Setting of the Sampling Cycle on Spread Spectrum

When the frequency spread function is ON (CTSUCRBL.SOFF = 0), the sampling period of the sensor drive pulse should be set less than 1/4 of the sensor drive pulse period by CTSUCRAL.CLK[1:0] bits and CTSUSO1.SDPA[7:0] bits .

15.4.10 Transmit Pin Setting When Using the Self-capacitance Method

When using the self-capacitance method, the transmit pin is not used for measurement. The transmission pin outputs a pulse that is in phase with the measurement pulse. Use it as an active shield on the board. In the self-capacitance method, do not select multiple transmit pins at the same time.

15.4.11 Notes on Noise Countermeasures

If the V_{DD} power supply fluctuates due to noise or other causes (for example, changes in the input/output current of port), the measurement results may vary. Also, if ripple noise is applied to the V_{DD} power supply, the control current of the capacitance sensor unit may change (decrease) depending on the frequency band of the ripple noise, resulting in a decrease in the measured capacitance value.

For details, refer to the application note **“Capacitive Touch Ripple Noise Prevention Guide (R30AN0453)”**.

Design the power supply circuit with these points in mind.

Table 15-4 shows the measured capacitance reduction characteristics due to V_{DD} power supply ripple noise (reference value).

Table 15-4. Capacitive Sensing Unit (CTSL2SLa) Power Supply Ripple Noise Characteristics (Reference Value)

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = V_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

Item	Symbol	Conditions (Cp = 20 pF)		MIN.	TYP.	MAX.	Unit
Measured capacitance reduction characteristics Note	C_{down}	Ripple frequency < 20 kHz	Amplitude: 100 mVpp	–	–	0.02	pF
		20 kHz ≤ Ripple frequency ≤ 2 MHz	Amplitude: 40 mVpp	–	–	0.06	
			Amplitude: 60 mVpp	–	–	0.10	
			Amplitude: 100 mVpp	–	–	0.33	
		2 MHz < Ripple frequency	Amplitude: 100 mVpp	–	–	0.01	

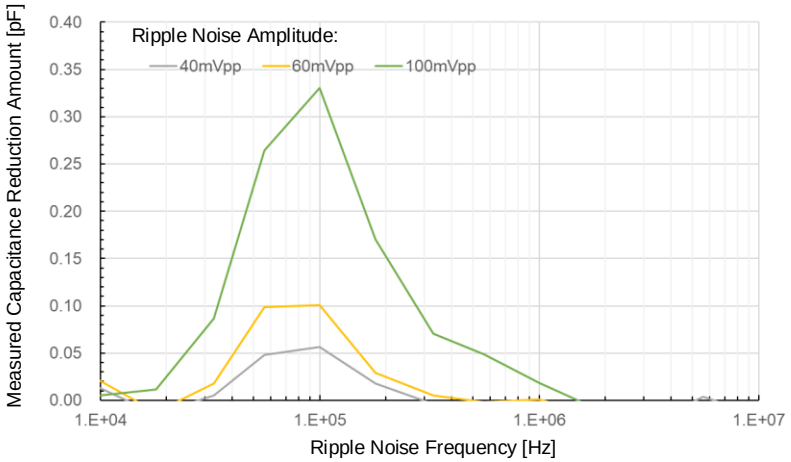
Note This is the value under the following conditions.

- This is the value when using the Self-capacitance method (CTSUCRAL.MD1 = 0).
- This is the value when the measured current range is 40 µA (CTSUCRAL.ATUNE1 = 1, CTSUCRAH.ATUNE2 = 0). For an overview of measured current range, refer to "**3.4 Current Range**" in the application note "**Capacitive Sensor MCU QE for Capacitive Touch Advanced Mode Parameter Guide (R30AN0428)**".
- The target value for offset adjustment is 37.5%. For an overview of offset adjustment, refer to "**3.2 Offset Tuning Target**" in the application note "**Capacitive Sensor MCU QE for Capacitive Touch Advanced Mode Parameter Guide (R30AN0428)**".

Remark Cp: Parasitic capacitance

Figure 15-33. Measured Capacitance Reduction Amount (Reference Value)

Conditions Note:
Measured current range = 40 µA (CTSUCRAL.ATUNE1 = 1, CTSUCRAH.ATUNE2 = 0)
Target value of offset tuning = 37.5%
T_A = −40 to 125 °C, V_{DD} = 2.7 to 5.5 V, Cp = 20 pF



Note For details about the conditions, refer to the application note "**Capacitive Sensor MCU, QE for Capacitive Touch Advanced Mode Parameter Guide (R30AN0428)**".

Remark Cp: Parasitic capacitance

CHAPTER 16 SERIAL ARRAY UNIT

Serial array unit has two serial channels. Each channel can achieve 3-wire serial (CSI), UART, simplified I^2C, and simplified I^2S communication.

Function assignment of each channel supported by the RL78/F22 and RL78/F25 is as shown below.

- RL78/F25 Products and RL78/F22 48-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C	Simplified I^2S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function)		IIC01	–
1	0	CSI10 (supporting SPI function) Note 1	UART1	IIC10	Supported
	1	CSI11 (supporting SPI function)		IIC11	–

- RL78/F22 32-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C	Simplified I^2S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function)		IIC01	–
1	0	CSI10 (supporting SPI function) Note 1	UART1	IIC10	Supported
	1	–		–	–

- RL78/F22 24-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C	Simplified I^2S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function) Note 2		IIC01	–
1	0	–	UART1	–	Supported
	1	–		–	–

Notes 1. 48-pin and 32-pin products do not have SSI10 pin.

2. 24-pin products do not have SSI01 pin.

Caution Most of the following descriptions in this chapter use the units and channels of the 100-pin products of RL78/F25 as an example.

16.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/F22 and RL78/F25 has the following features.

16.1.1 3-wire Serial I/O (CSI00, CSI01, CSI10, CSI11)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel.

3-wire serial communication is clocked communication performed by using three communication lines: one for the serial clock (SCK), one for transmitting serial data (SO), one for receiving serial data (SI).

For details about the settings, see **16.5 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 SCK cycle time (t_{KCY}) characteristics (see **CHAPTER 37** and **CHAPTER 38 ELECTRICAL SPECIFICATIONS**).

16.1.2 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 **16.7 UART (UART0, UART1) Communication**.

[Data transmission/reception]

- Data length of 7, 8, 9, and 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]

- Transmission end interrupt/buffer empty interrupt
- Reception end 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

16.1.3 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 **16.9 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 **16.9.3 (2) Processing Flow** for details.

Remarks 1. To use an I²C bus of full function, see **CHAPTER 17 SERIAL INTERFACE IICA**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 16-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	16 bits
Buffer register	Serial data register mn (SDRmn) Note
Serial clock I/O	SCK00, SCK01, SCK10, 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	SSI00, SSI01, SSI10, SSI11 pins (for 3-wire serial I/O)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Serial slave select enable register m (SSEm) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn)
	• Port input mode registers 1, 3, 5 to 7, 12 (PIM1, PIM3, PIM5 to PIM7, PIM12) • Port output mode registers 1, 3, 6, 7, 12 (POM1, POM3, 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)
	• Unit select register (UTSEL) • I^2S function control register (I2SER)

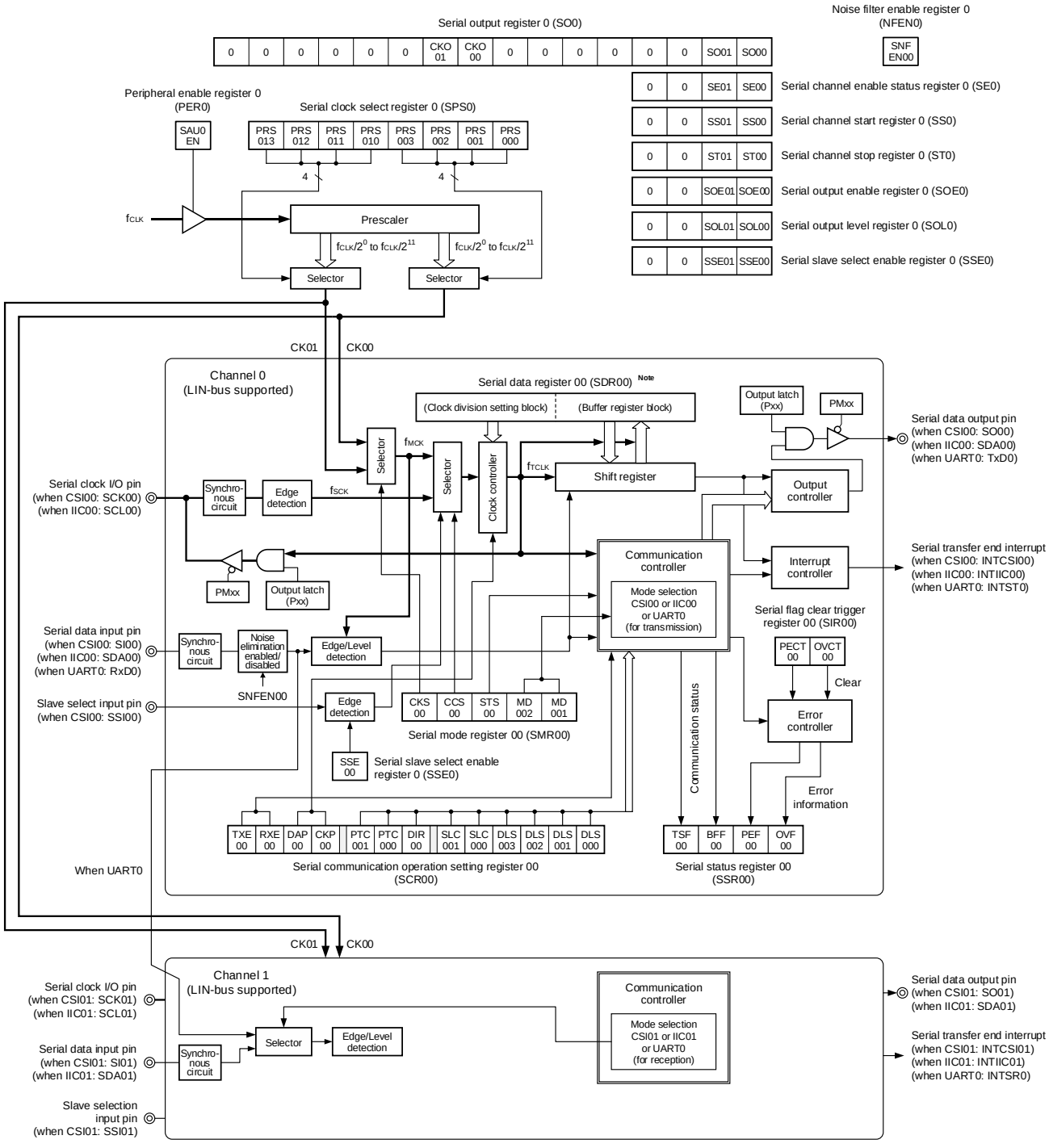
Note When SEmn is 1, the lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

- CSIp communication … SDRpL (CSIp data register)
- UARTq reception … SDRmnL (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 16-1 shows the block diagram of the sal array unit 0.

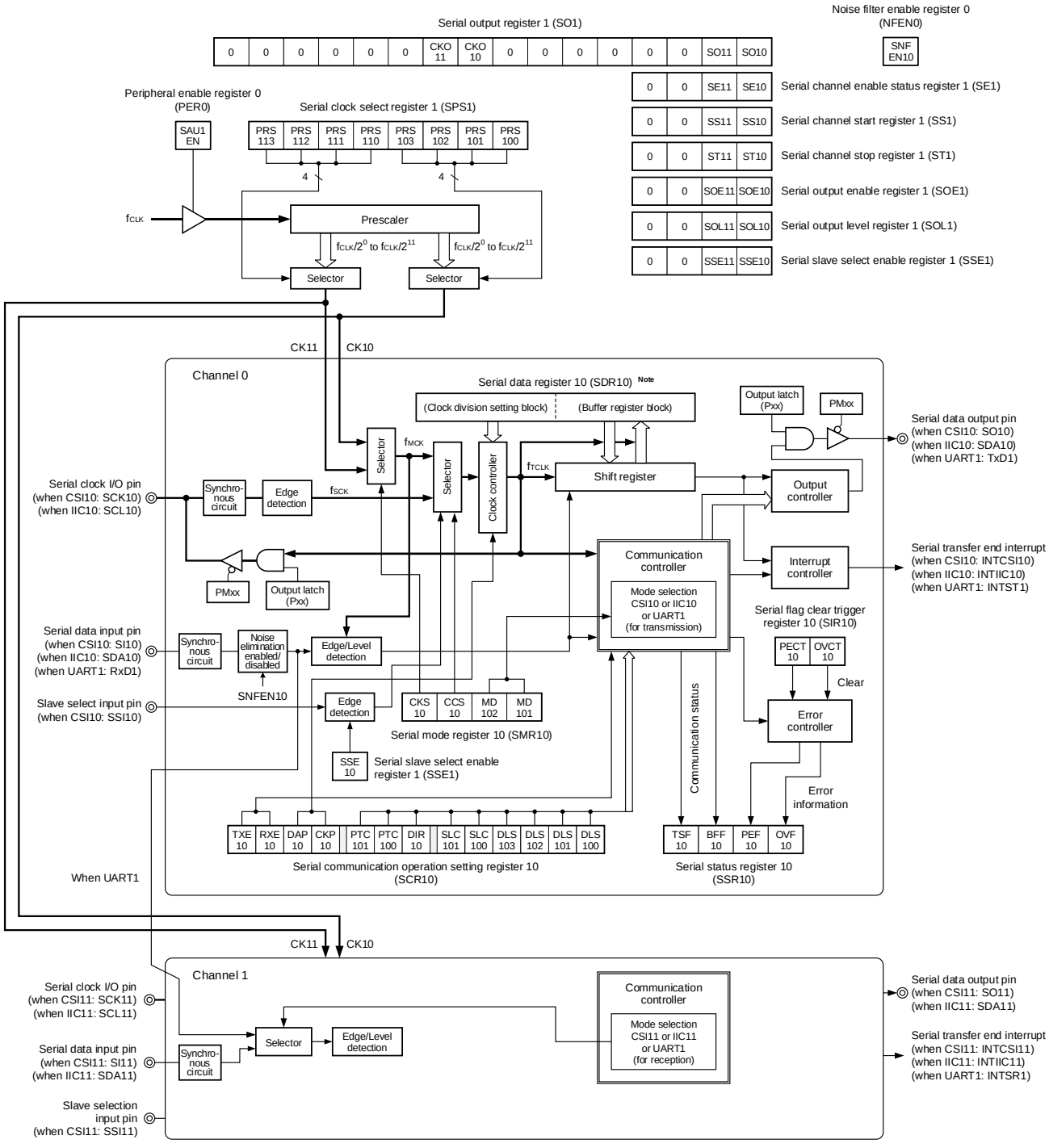
Figure 16-1. Block Diagram of Serial Array Unit 0



Note If operation is stopped (SE0n = 0), the upper 7 bits set the clock division, and the lower bits have no meaning. If operation is in progress (SE0n = 1), the serial data register 0n functions as the buffer register.

Figure 16-2 shows the block diagram of the serial array unit 1.

Figure 16-2. Block Diagram of Serial Array Unit 1



Note If operation is stopped (SE1n = 0), the upper 7 bits set the clock division, and the lower bits have no meaning. If operation is in progress (SE1n = 1), the serial data register 1n functions as the buffer register.

(1) Shift Register

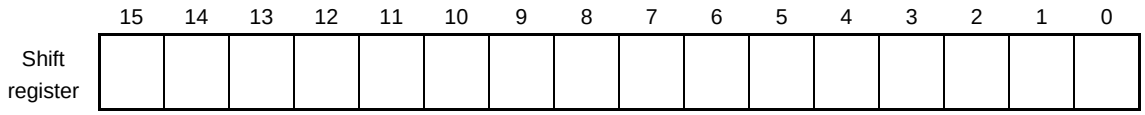
This is a 16-bit register that converts parallel data into serial data or vice versa.

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

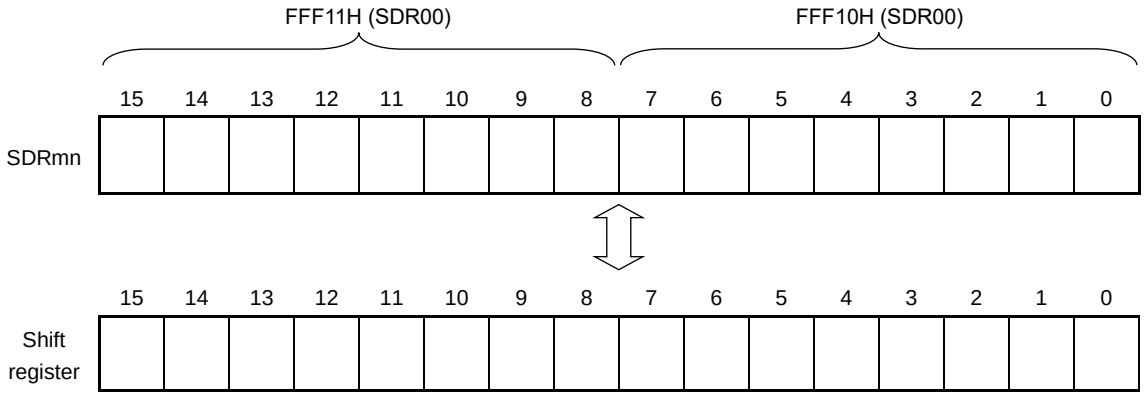
Reset signal generation clears the SDRmn register to 0000H.

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 16-3. 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 **16.3 Registers Controlling Serial Array Unit**.

16.3 Registers Controlling Serial Array Unit

Caution When accessing a register of SAUm (m = 0, 1), select the target unit number m with the UTSEL register before accessing.

Table 16-2. Serial Array Unit Register Configuration

Address	Register Name	Symbol		After Reset	Access Size
F02C0H	Peripheral enable register 0	PER0		00H	1, 8
F0070H	Noise filter enable register 0	NFEN0		00H	1, 8
F0073H	Input switch control register	ISC		00H	1, 8
F01F0H	Unit select register	UTSEL		00H	1, 8
F01F6H	I^2S function control register	I2SER		00H	8
F0100H	Serial status register m0	SSRm0L	SSRm0	0000H	8, 16
F0101H		–			
F0102H	Serial status register m1	SSRm1L	SSRm1	0000H	8, 16
F0103H		–			
F0104H	Serial flag clear trigger register m0	SIRm0L	SIRm0	0000H	8, 16
F0105H		–			
F0106H	Serial flag clear trigger register m1	SIRm1L	SIRm1	0000H	8, 16
F0107H		–			
F0108H	Serial mode register m0	SMRm0		0020H	16
F010AH	Serial mode register m1	SMRm1		0020H	16
F010CH	Serial communication operation setting register m0	SCRm0		0087H	16
F010EH	Serial communication operation setting register m1	SCRm1		0087H	16
F0110H	Serial channel enable status register m	SEmL	SEm	0000H	1, 8, 16
F0111H		–			
F0112H	Serial channel start register m	SSmL	SSm	0000H	1, 8, 16
F0113H		–			
F0114H	Serial channel stop register m	STmL	STm	0000H	1, 8, 16
F0115H		–			
F0116H	Serial clock select register m	SPSmL	SPSm	0000H	8, 16
F0117H		–			
F0118H	Serial output register m	SOm		0303H	16
F011AH	Serial output enable register m	SOEmL	SOEm	0000H	1, 8, 16
F011BH		–			
F0120H	Serial output level register m	SOLmL	SOLm	0000H	8, 16
F0121H		–			
F0122H	Serial slave select enable register m	SSEmL	SSEm	0000H	8, 16
F0123H		–			
FFF10H	Serial data register 00	SDR00L	SDR00	0000H	8, 16
FFF11H		–			
FFF12H	Serial data register 01	SDR01L	SDR01	0000H	8, 16
FFF13H		–			
FFF48H	Serial data register 10	SDR10L	SDR10	0000H	8, 16
FFF49H		–			
FFF4AH	Serial data register 11	SDR11L	SDR11	0000H	8, 16
FFF4BH		–			

Remarks 1. For port-related registers, refer to the following sections.

16.3.17 Port Input Mode Registers (PIM1, PIM3, PIM5 to PIM7, PIM12)

16.3.18 Port Output Mode Registers (POM1, POM3, POM6, POM7, POM12)

16.3.19 Port Mode Registers (PM1, PM3 to PM7, PM12)

16.3.20 Port Input Threshold Control Registers (PITHL1, PITHL3 to PITHL7, PITHL10, PITHL12, PITHL15)

2. m: Unit number (m = 0, 1), The unit number m is selected by SSEL0 bit of the UTSEL register.

16.3.1 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 16-4. Format of Peripheral Enable Register 0 (PER0)

Address: F02C0H 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. • SFR used by serial array unit m cannot be written. • Serial array unit m is in the reset status.
1	Enables input clock supply. • 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 UTSEL and I2SER registers, ISC register, and port related registers (Px, PMx, PIMx, POMx, PMCx, PITHLx, PSRSEL, and NFEN0)).

2. Be sure to clear the bit 6.

Remark m: Unit number (m = 0, 1)

16.3.2 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 16-5. Format of Serial Clock Select Register m (SPSm)

Address: F0116H, F0117H 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	f_{CLK} = 40 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz	40 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz	20 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz	10 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz	5 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz	2.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz	1.25 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz	625 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz	313 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz	156 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz	78.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz	39.1 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz	19.5 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 (SAU).

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
Unit number m is selected by SSEL0 bit of the UTSEL register.

16.3.3 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 16-6. Format of Serial Mode Register mn (SMRmn)

Address: F0108H, F0109H (SMRm0), F0110AH, F010BH (SMRm1), 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

CKSmn	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.	

CCSmn	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 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.	

STSmn	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.	

SISmn0	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

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

MDmn2	MDmn1	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

MDmn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (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),
Unit number m is selected by SSEL0 bit of the UTSEL register.

16.3.4 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, 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.

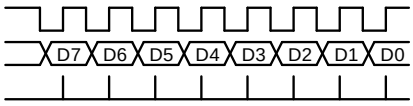
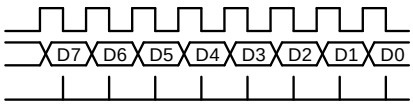
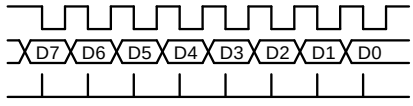
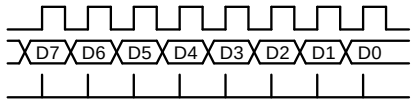
Reset signal generation sets the SCRmn register to 0087H.

Figure 16-7. Format of Serial Communication Operation Setting Register mn (SCRmn)

Address: F010CH, F010DH (SCRm0), F010EH, F010FH (SCRm1), 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

TXEmn	RXEmn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAPmn	CKPmn	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.			

PTCmn1	PTCmn0	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.			

Note 0 is always added regardless of the contents of data.

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

DIRmn	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	SLCmn0	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. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I^2C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the CSI mode.		

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), Unit number m is selected by SSEL0 bit of the UTSEL register.

16.3.5 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 **16.2 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 16-8. 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)

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 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.

4. Do not write the lower 8 bits with an 8-bit memory manipulation instruction if operation is stopped (SEmn = 0). If these bits are written to, the higher 7 bits are cleared to 0.

Remarks 1. For the function of the SDRmn register when operation is in progress (SEmn = 1), see **16.2 Configuration of Serial Array Unit**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.3.6 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 16-9. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0104H, F0105H (SIRm0), F0106H, F0107H (SIRm1), 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 flag 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 SSRmn 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),
Unit number m is selected by SSEL0 bit of the UTSEL register.

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.

16.3.7 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 16-10. Format of Serial Status Register mn (SSRmn)

Address: F0100H, F0101H (SSRm0), F0102H, F0103H (SSRm1), 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

TSFmn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.

<Clear conditions>
- 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).
- Communication ends.

<Set condition>
- Communication starts.

BFFmn	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>
- Transferring transmit data from the SDRmn register to the shift register ends during transmission.
- Reading receive data from the SDRmn register ends during reception.
- 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).

<Set conditions>
- 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).
- 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).
- A reception error occurs.

FEFmn	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).

<Clear condition>
- 1 is written to the FECTmn bit of the SIRmn register.

<Set condition>
- A stop bit is not detected when UART reception ends.

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

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>
- 1 is written to the PECTmn bit of the SIRmn register.

<Set conditions>
- The parity of the transmit data and the parity bit do not match when UART reception ends (parity error).
- 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>
- 1 is written to the OVCTmn bit of the SIRmn register.

<Set conditions>
- 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).
- Transmit data is not ready for slave transmission or transmission and reception in CSI mode.

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 (OVFmn = 1) is detected.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),
Unit number m is selected by SSEL0 bit of the UTSEL register.

16.3.8 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 a 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears the SSm register to 0000H.

Figure 16-11. Format of Serial Channel Start Register m (SSm)

Address: F0112H, F0113H After reset: 0000H R/W

Symbol	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	SSm0

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 SSm 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),
Unit number m is selected by SSEL0 bit of the UTSEL register.

2. When the SSm register is read, 0000H is always read.

16.3.9 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 16-12. Format of Serial Channel Stop Register m (STm)

Address: F0114H, F0115H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	STm1	STm0

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 STm register to 0.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),
Unit number m is selected by SSEL0 bit of the UTSEL register.

2. When the STm register is read, 0000H is always read.

16.3.10 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 SEmL.

Reset signal generation clears the SEm register to 0000H.

Figure 16-13. Format of Serial Channel Enable Status Register m (SEm)

Address: F0110H, F0111H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SEm1	SEm0

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: overrun 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),
Unit number m is selected by SSEL0 bit of the UTSEL register.

16.3.11 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 16-14. 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
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1	SOEm0

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 SOEm register to 0.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),
Unit number m is selected by SSEL0 bit of the UTSEL register.

16.3.12 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 16-15. 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
SOm	0	0	0	0	0	0	CKOm1	CKOm0	0	0	0	0	0	0	SOm1	SOm0

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 10 and 7 to 2 of the SOm register to 0.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),
Unit number m is selected by SSEL0 bit of the UTSEL register.

16.3.13 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 16-16. Format of Serial Output Level Register m (SOLm)

Address: F0120H, F0121H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL m1	SOL m0

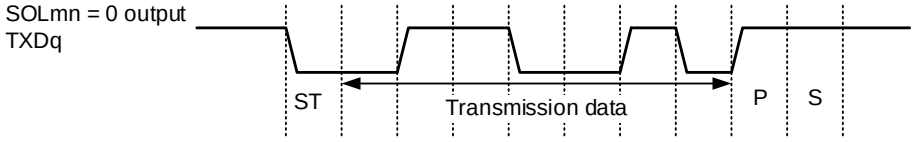
SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution Be sure to clear bits 15 to 2 of the SOLm register to 0.

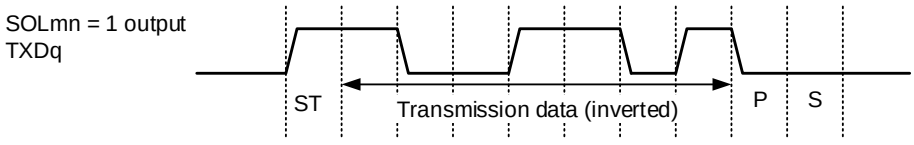
Figure 16-17 shows examples in which the level of transmission data is inverted during UART transmission. This figure is an example of the condition that the data length = 7 bits, the parity bit output is enabled, and the parity bit is 1.

Figure 16-17. Examples of Inverted Transmission Data

(a) Normal output (SOLmn = 0)



(b) Inverted output (SOLmn = 1)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), q: UART number (q = 0, 1),
Unit number m is selected by SSEL0 bit of the UTSEL register.

16.3.14 Serial Slave Select Enable Register m (SSEm)

The SSEm register controls the SSImn pin input of the channel during CSI communication and in slave mode.

While a high-level signal is being input to the SSImn pin, no transmission/reception operation is performed even if a serial clock is input. While a low-level signal is being input to the SSImn 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.

Set the SSEm register by a 16-bit memory manipulation instruction.

Set the lower 8 bits of the SSEm register with an 8-bit memory manipulation instruction with SSEmL.

Figure 16-18. Format of Serial Slave Select Enable Register m (SSEm)

Address: F0122H, F0123H After reset: 0000H R/W

Symbol	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	SSEm0

SSEmn	Channel n SSImn input setting in CSI communication and slave mode
0	Disables SSImn pin input.
1	Enables SSImn pin input.

Cautions 1. Writing is prohibited other than during CSI communication and in slave mode.

2. These bits can be set only when channel operation is stopped (SEmn = 0).

3. Be sure to clear bits 15 to 2 of the SSEm register to 0.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),
Unit number m is selected by SSEL0 bit of the UTSEL register.

16.3.15 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 16-19. 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	ISC4	ISC3	ISC2	0	ISC0

ISC4	Switching external input (INTP6)
0	Uses the input signal of the INTP6 pin as an external interrupt input.
1	Uses the input signal of the LRxD2 pin as an external interrupt input.

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).

Cautions 1. Bits 7 to 5 and 1 should always be set to 0.

2. Be sure to set ISC3 and ISC4 bits to 0 in RL78/F22 products.

16.3.16 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 16-20. 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	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. 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. Clear the SNFEN00 bit to 0 to use the other than RxD0 pin.	

Caution Be sure to clear bits 7 to 3 and 1 to 0.

16.3.17 Port Input Mode Registers (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 16-21. Format of Port Input Mode Registers (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

16.3.18 Port Output Mode Registers (POM1, POM3, POM6, POM7, POM12)

These registers set the output mode of ports 1, 3, 6, 7, and 12 in 1-bit units.

Set the POM1, POM3, POM6, POM7, and POM12 registers by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the POM1, POM3, POM6, POM7, and POM12 registers to 00H.

Figure 16-22. Format of Port Output Mode Registers (POM1, POM3, 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 F0053H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM3	0	0	0	0	0	POM32	0	0

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, 3, 6, 7, 12; n = 0 to 7)
0	Normal output mode
1	N-ch open-drain output (EV_{DD0} 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.

16.3.19 Port Mode Registers (PM1, PM3 to PM7, PM12)

These registers set input/output of ports1, 3 to 7, and 12 in 1-bit units.

When using the serial data output or serial clock output, or when using serial data input/output (SDAmn), set the bit in the port mode register (PMmn) corresponding to each port to 0. And set the bit in the port register (Pmn) corresponding to each port to 1.

Also, when using the serial data input or serial clock input, set the PMmn bit corresponding to each port to 1.

At this time, the bit in the Pmn bit may be 0 or 1.

Example 1:

When using P12/**SO10**/**TXD1** for serial data output

Set the PM12 bit to 0.

Set the P12 bit of the P1 register to 1.

Example 2:

When using P11/SI10/**SDA10**/RXD1 for serial data input/output

Set the PM11 bit to 0.

Set the P11 bit of the P1 register to 1.

Example 3:

When using P16/**SI00**/SDA00/**RXD0** for serial data input

Set the PM16 bit to 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 16-23. Format of Port Mode Registers (PM1, PM3 to PIM7, PIM12)

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	1	1	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: 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 to 7, 12; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

16.3.20 Port Input Threshold Control Registers (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, P41, P43, P50, P52 to P54, P60 to P63, P70, P71, P73, P75 to P77, P107, P120, 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 16-24. Format of Port Input Threshold Control Register (PITHL1, PITHL3 to PITHL7, PITHL10, PITHL12, PITHL15)

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	PITHL41	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	PITHL120

Address: F002FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PITHL15	0	0	0	PITHL154	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 (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.

16.3.21 Unit Select Register (UTSEL)

The UTSEL register is used to select the target unit for accessing the serial array unit (SAU) and timer array unit (TAU) registers.

Set the UTSEL register by a 1-bit or 8-bit memory manipulation instruction.

The SAU0 and SAU1 registers are assigned to the same address.

Before accessing the SAU0 or SAU1 registers, select the unit to be accessed with the SSEL0 bit of the UTSEL register.

Figure 16-25. Format of Unit Select Register (UTSEL)

Address: F01F0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
UTSEL	0	0	0	0	0	SSEL0	0	TSEL0

SSEL0	Serial array unit selection
0	SAU0 is selected.
1	SAU1 is selected.

TSEL0	Timer array unit selection
0	TAU0 is selected.
1	TAU1 is selected.

16.3.22 I²S Function Control Register (I2SER)

The I2SER register is used to enable or disable the Simplified I²S communication function.

Set the I2SER register by an 8-bit memory manipulation instruction.

Reset signal generation sets the I2SER register to 00H.

Figure 16-26. I²C Function Control Register (I2SER)

Address: F01F6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
I2SER	0	0	0	0	0	0	0	I2SEN

I2SEN	Control simplified I²S communication function
0	Disable simplified I²S communication function.
1	Enable simplified I²S communication function

Remark Simplified I²S communication (supports only master communication) is realized by using the Serial Array Unit (CSI10), channels 2 and 3 of Timer Array Unit 1, and a dedicated logic circuit.
For details, see **16.10 Simplified I²S Communication**.

16.4 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.

16.4.1 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.

Caution 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 (PIM1, PIM3, PIM5 to PIM7, PIM12)**
- **Port output mode registers (POM1, POM3, POM6, POM7, POM12)**
- **Port mode registers (PM1, PM3 to PM7, PM12)**
- **Port registers (P1, P3 to P7, P12)**
- **Unit select register (UTSEL)**
- **I^2S function control register (I2SER)**

Remark m: Unit number (m = 0, 1),
Unit number m is selected by SSEL0 bit of the UTSEL register.

16.4.2 Stopping the Operation by Channels

The stopping of the operation by channels is set using each of the following registers.

(a) Serial channel stop register m (STm)

Set the STmn bit to 1 (stops the communication operation).

(b) Serial channel enable status register (SEm)

When the STmn bit is set to 1, the SEmn bit becomes 0 (operation stops).

(c) Serial output enable register m (SOEm)

Set the SOEmn bit to 0 (stops output by serial communication operation).

* For channel n, whose serial output is stopped, the SOmn bit value can be set by software.

(d) Serial output register m (SOm)

Set the CKOmn and SOmn bits to 1 (serial clock or 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.

(e) Serial slave select enable register (SSEm)

Set the SSEmn bit to 0 (disable the input value of the SSImn pin).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),
Unit number m is selected by SSEL0 bit of the UTSEL register.

16.5 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) Communication

This is a clocked communication function that uses three lines: serial clock (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, CSI00, CSI01, CSI10, and CSI11 support the slave select input function. For details, refer to **16.6 Clock Synchronous Serial Communication with SPI Function**.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics (see **CHAPTER 37** and **CHAPTER 38 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.

- RL78/F25 Products and RL78/F22 48-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C	Simplified I^2S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function)		IIC01	–
1	0	CSI10 (supporting SPI function) Note 1	UART1	IIC10	Supported
	1	CSI11 (supporting SPI function)		IIC11	–

- RL78/F22 32-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C	Simplified I^2S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function)		IIC01	–
1	0	CSI10 (supporting SPI function) Note 1	UART1	IIC10	Supported
	1	–		–	–

- RL78/F22 24-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C	Simplified I^2S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function) Note 2		IIC01	–
1	0	–	UART1	–	Supported
	1	–		–	–

Notes 1. 48-pin and 32-pin products do not have SSI10 pin.

2. 24-pin products do not have SSI01 pin.

3-wire serial I/O (CSI00, CSI01, CSI10, CSI11) performs the following six types of communication operations.

- Master transmission (See **16.5.1 Master Transmission**.)
- Master reception (See **16.5.2 Master Reception**.)
- Master transmission/reception (See **16.5.3 Master Transmission/Reception**.)
- Slave transmission (See **16.5.4 Slave Transmission**.)
- Slave reception (See **16.5.5 Slave Reception**.)
- Slave transmission/reception (See **16.5.6 Slave Transmission/Reception**.)

16.5.1 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	SCK00, SO00	SCK01, SO01	SCK10, SO10	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 • DAPmn = 0: Data output starts from the start of the serial clock operation. • 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 • CKPmn = 0: Forward • 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)

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 0 (operation clock selected by CKSmn bit).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 00B (CSI mode).
The MDmn0 bit selects interrupt source of channel n (0: transfer end, 1: buffer empty).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 10B (transmission only).
DAPmn and CKPmn bits select data and clock phase.
PTCmn1 and PTCmn0 bits set to 00B.
The DIRmn bit selects data transfer sequence (0: MSB first, 1: LSB first).
SLCmn1 and SLCmn0 bits are set to 00B.
DLSmn3 to DLSmn0 bits select data length (6H (7 bits) to FH (16 bits)).

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set baud rate to bits 15 to 9 of the SDRmn register.
When operation is in progress (SEmn = 1): Set the transmit data in the SDRmn register.

Note Lower 8 bit in the SDRmn register: SDRpL register.

(d) Serial output register m (SOm)

Set CKOmn bit and SOmn bit of the target channel.
The CKOmn bit is set to 1 when CKPmn bit of the SCRmn register is 0 and 0 when CKPmn bit is 1 to start communication.

(e) Serial output enable register m (SOEm)

Set the SOEmn bit of the target channel to 1 (serial output enable).

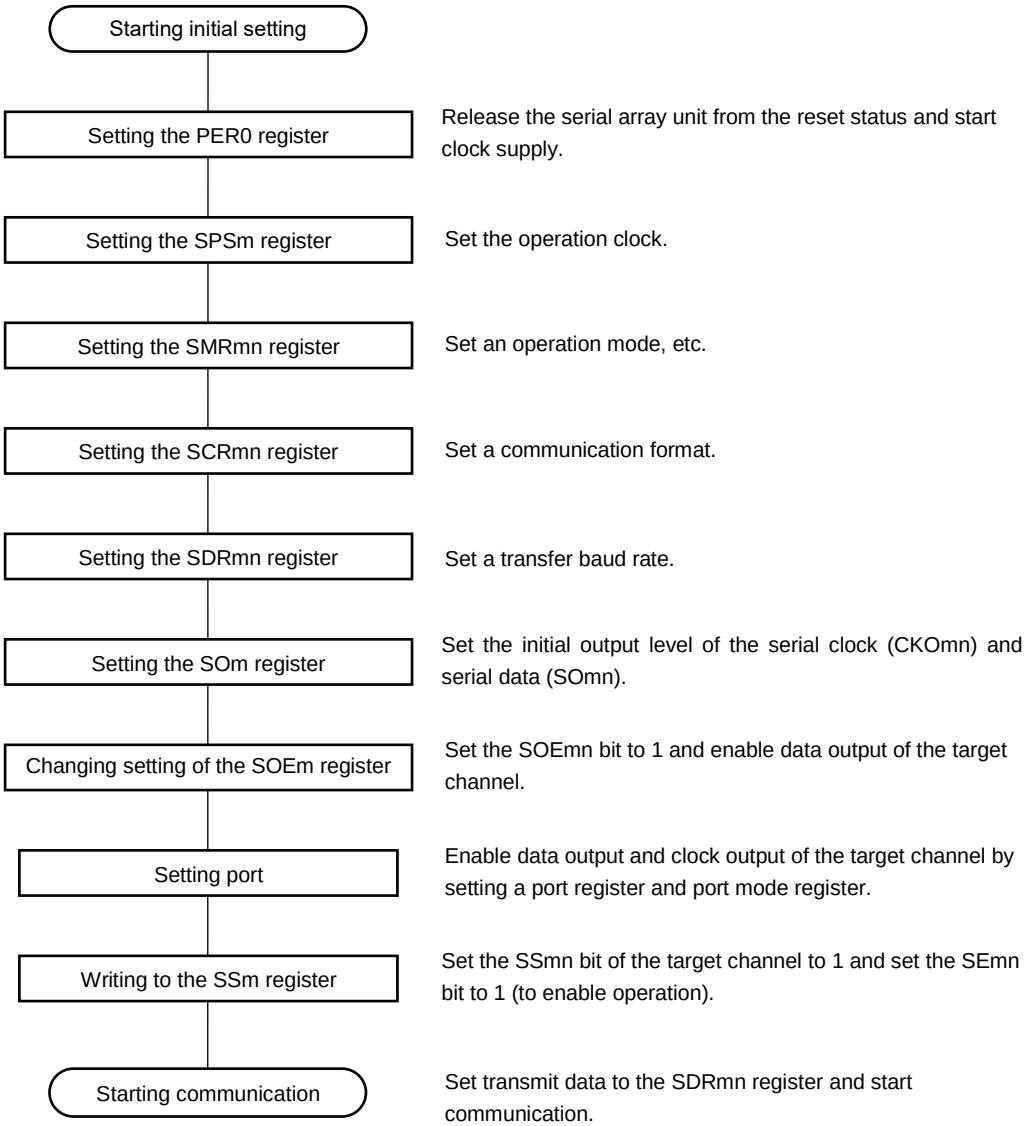
(f) Serial channel start register m (SSm)

Set the SSmn bit of the target channel to 1 (operation start).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

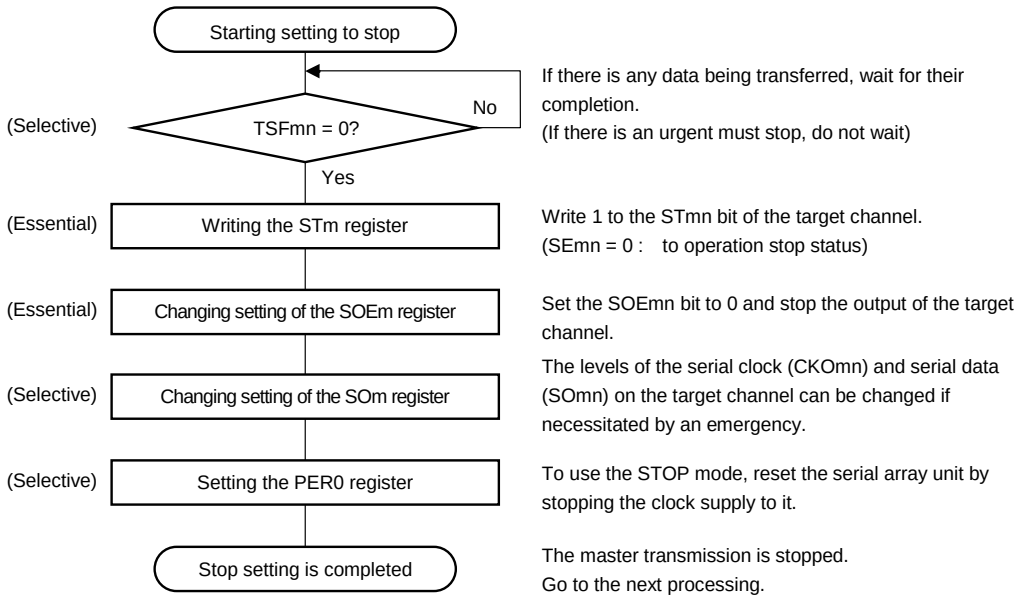
(2) Operation Procedure

Figure 16-27. Initial Setting Procedure for Master Transmission



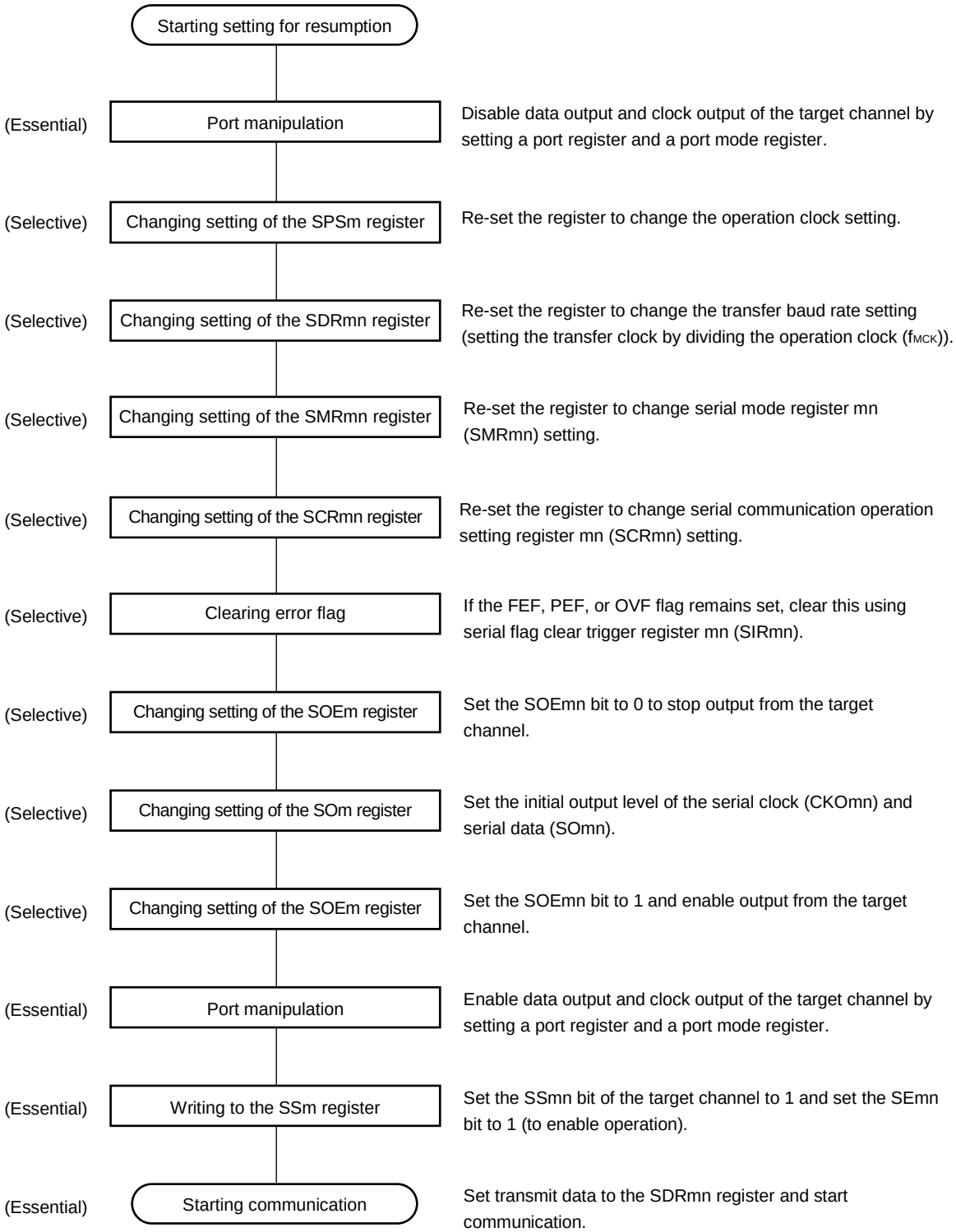
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

Figure 16-28. Procedure for Stopping Master Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

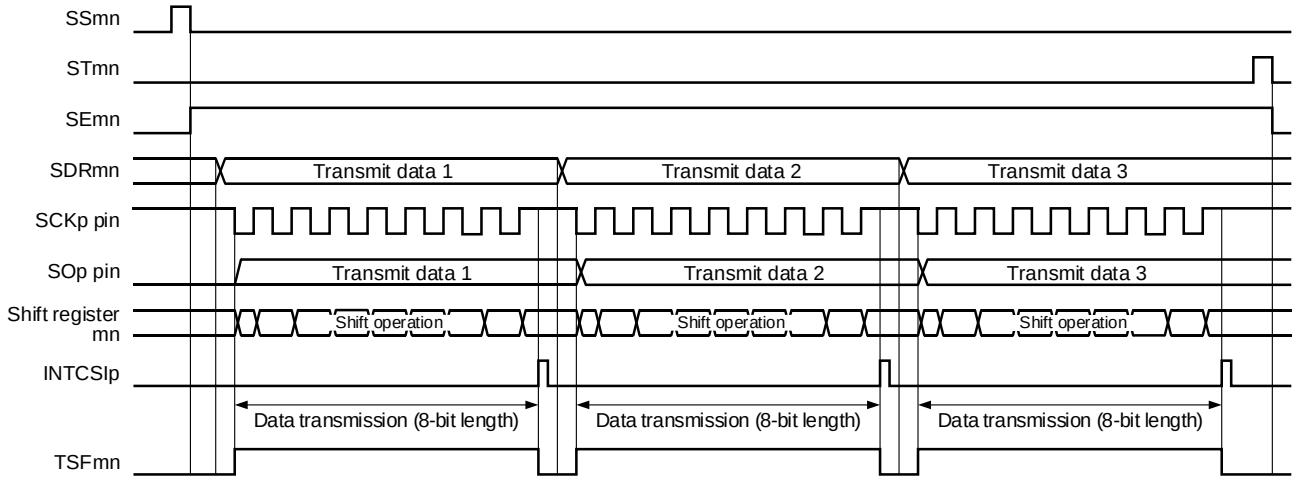
Figure 16-29. Procedure for Resuming Master Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

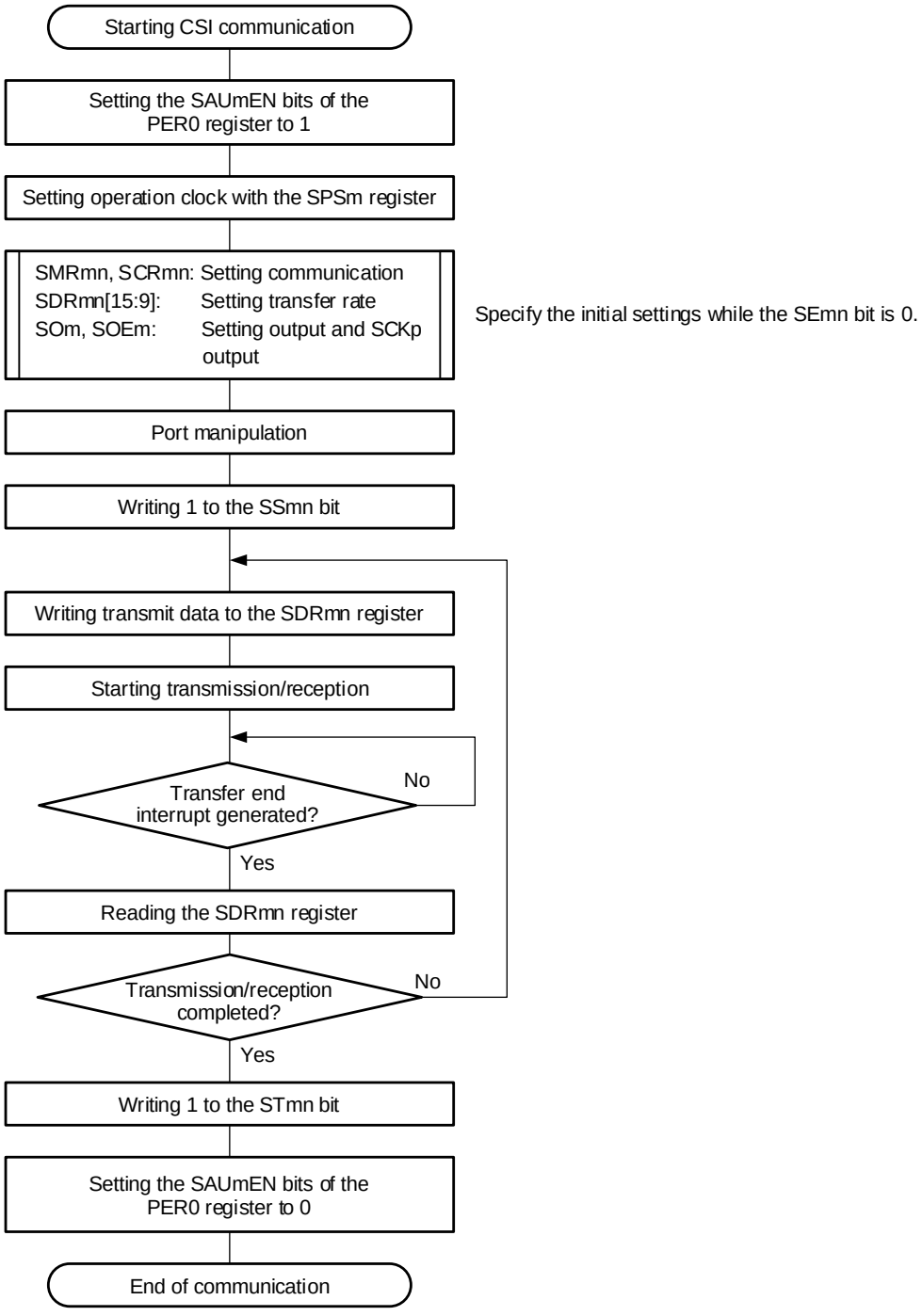
(3) Processing Flow (in Single-Transmission Mode)

Figure 16-30. 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)

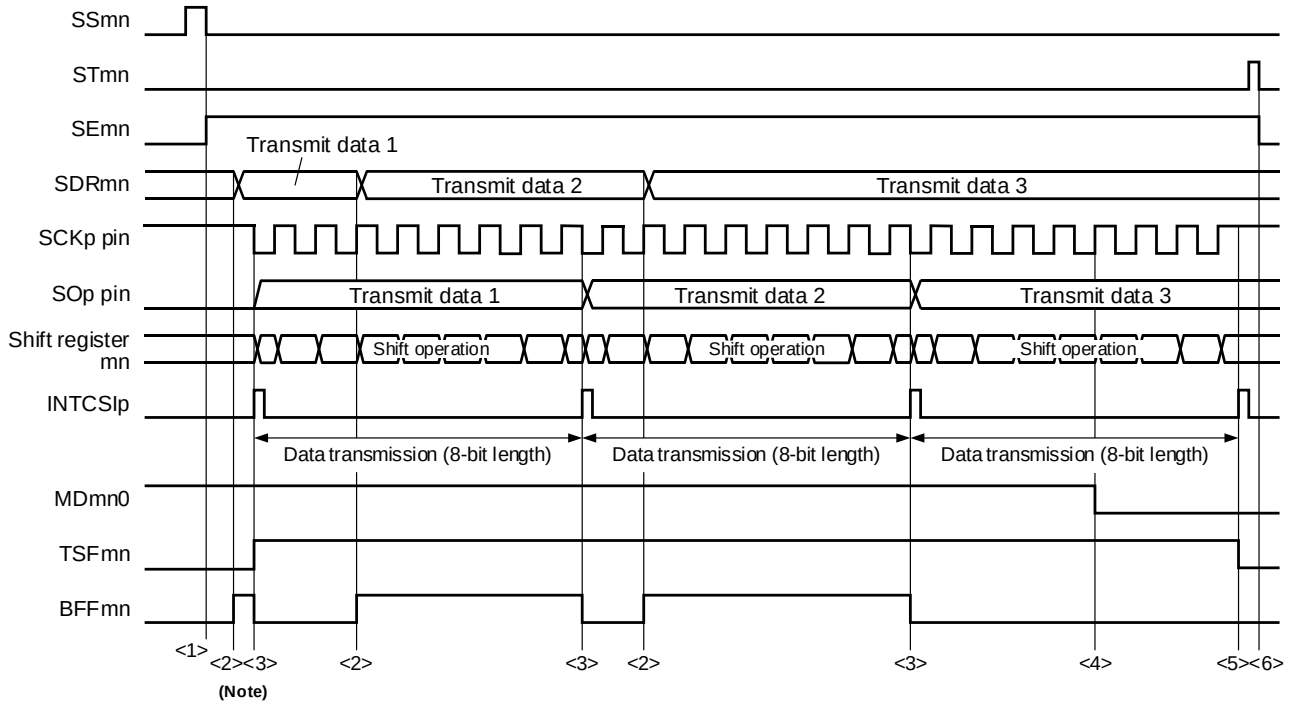
Figure 16-31. Flowchart of Master Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(4) Processing Flow (in Continuous Transmission Mode)

Figure 16-32. Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



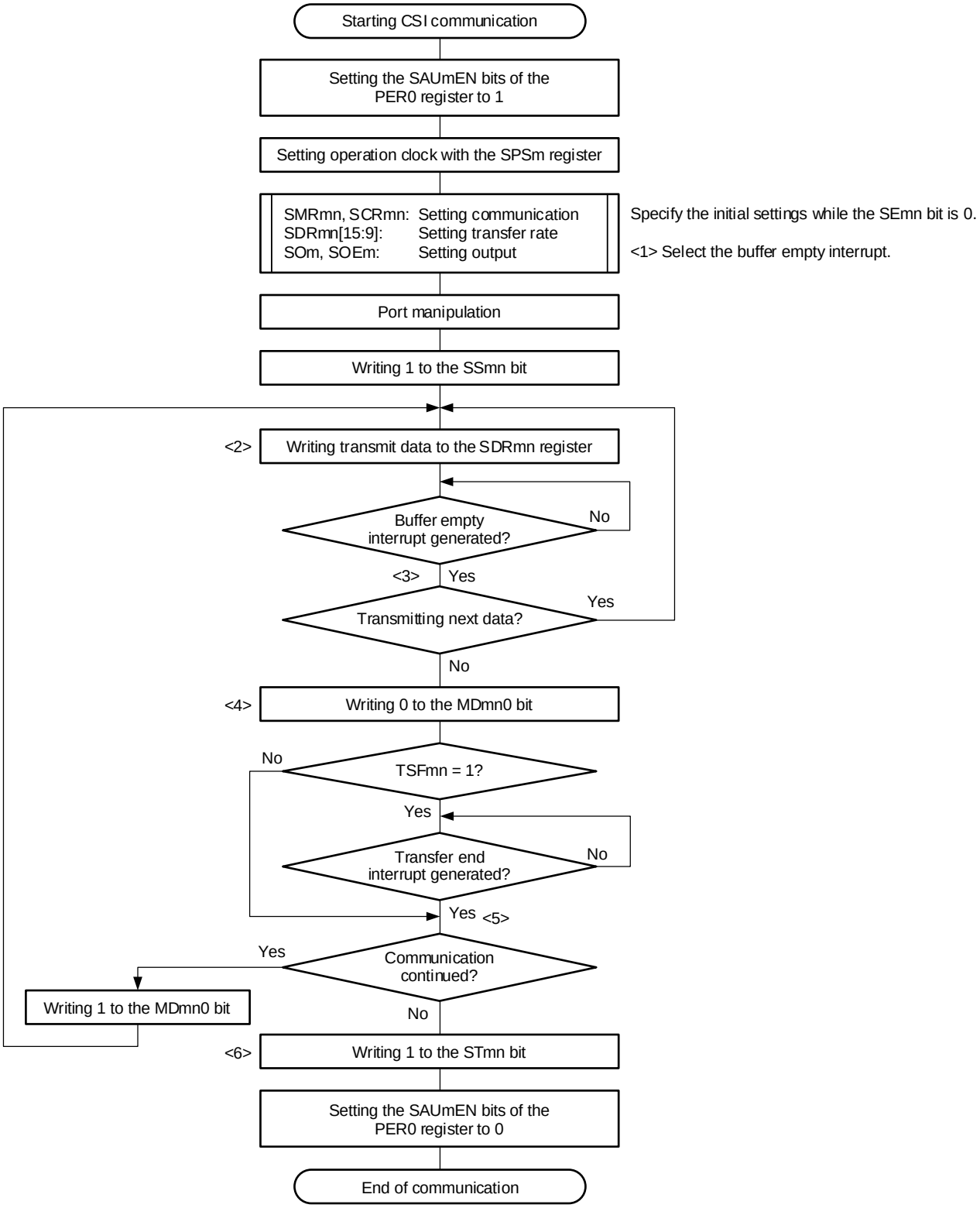
Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution 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.

Remarks 1. <1> to <6> in the above figure correspond to <1> to <6> in **Figure 16-33**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 16-33. Flowchart of Master Transmission (in Continuous Transmission Mode)



Remarks 1. <1> to <6> in the above figure correspond to <1> to <6> in **Figure 16-32**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.5.2 Master Reception

Master reception is an operation wherein this MCU outputs a transfer clock and receives data from the 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	SCK00, SI00	SCK01, SI01	SCK10, SI10	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 • DAPmn = 0: Data input starts from the start of the serial clock operation. • 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 • CKPmn = 0: Forward • 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)

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 0 (operation clock selected by CKSmn bit).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 00B (CSI mode).
The MDmn0 bit selects interrupt source of channel n (0: transfer end, 1: buffer empty).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 01B (reception only).
DAPmn and CKPmn bits select data and clock phase.
PTCmn1 and PTCmn0 bits set to 00B.
The DIRmn bit selects data transfer sequence (0: MSB first, 1: LSB first).
SLCmn1 and SLCmn0 bits are set to 00B.
DLSmn3 to DLSmn0 bits select data length (6H (7 bits) to FH (16 bits)).

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set baud rate to bits 15 to 9 of the SDRmn register.
When operation is in progress (SEmn = 1): Receive data (Write FFH as dummy data for 8-bit data transfer)

Note Lower 8 bit in the SDRmn register: SDRpL register.

(d) Serial output register m (SOm)

Set CKOmn bit of the target channel.
The CKOmn bit is set to 1 when CKPmn bit of the SCRmn register is 0 and 0 when CKPmn bit is 1 to start communication.

(e) Serial output enable register m (SOEm)

The register that not used in this mode.

(f) Serial channel start register m (SSm)

Set the SSmn bit of the target channel to 1 (operation start).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

(2) Operation Procedure

Figure 16-34. Initial Setting Procedure for Master Reception

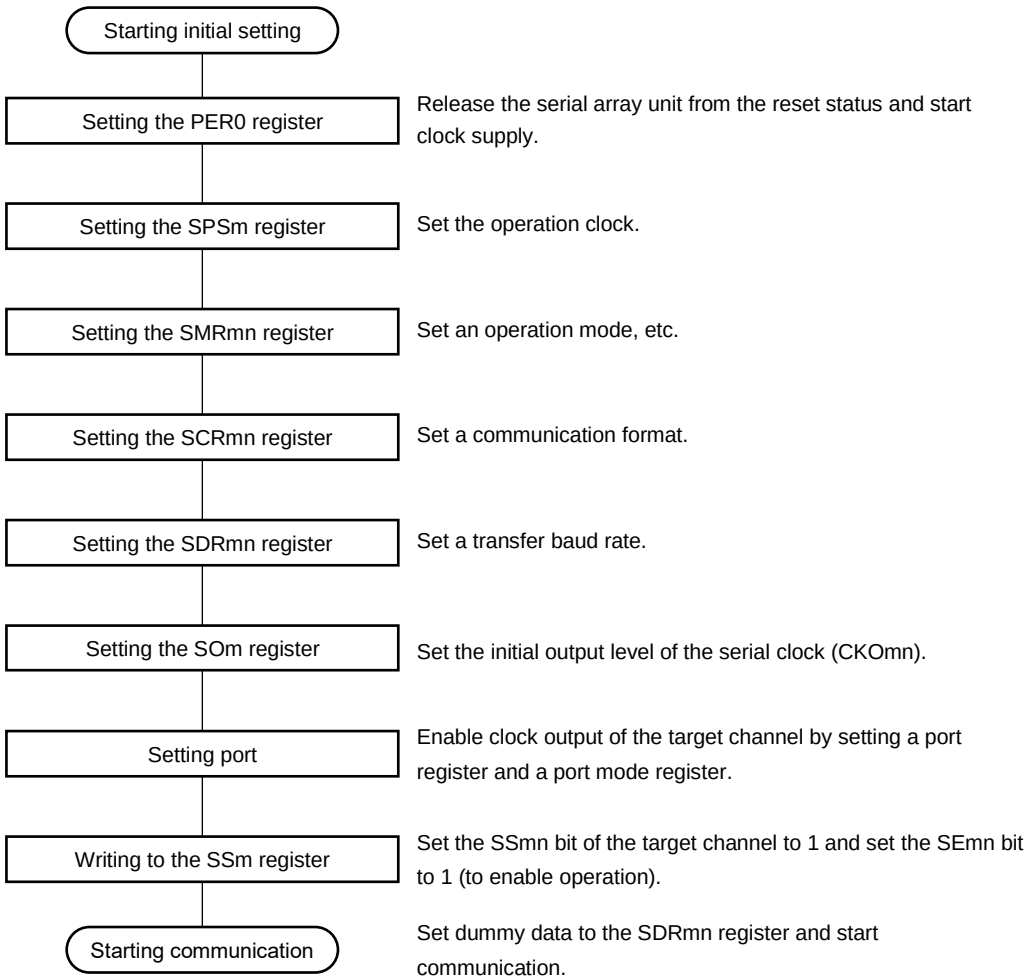
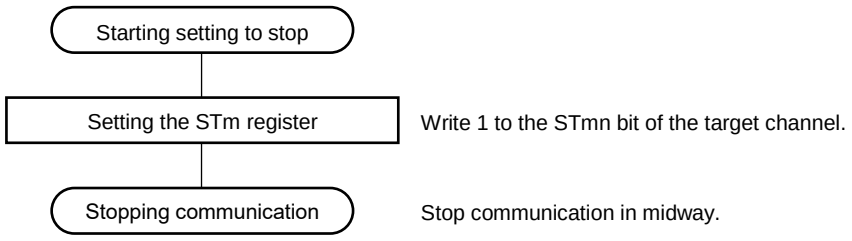


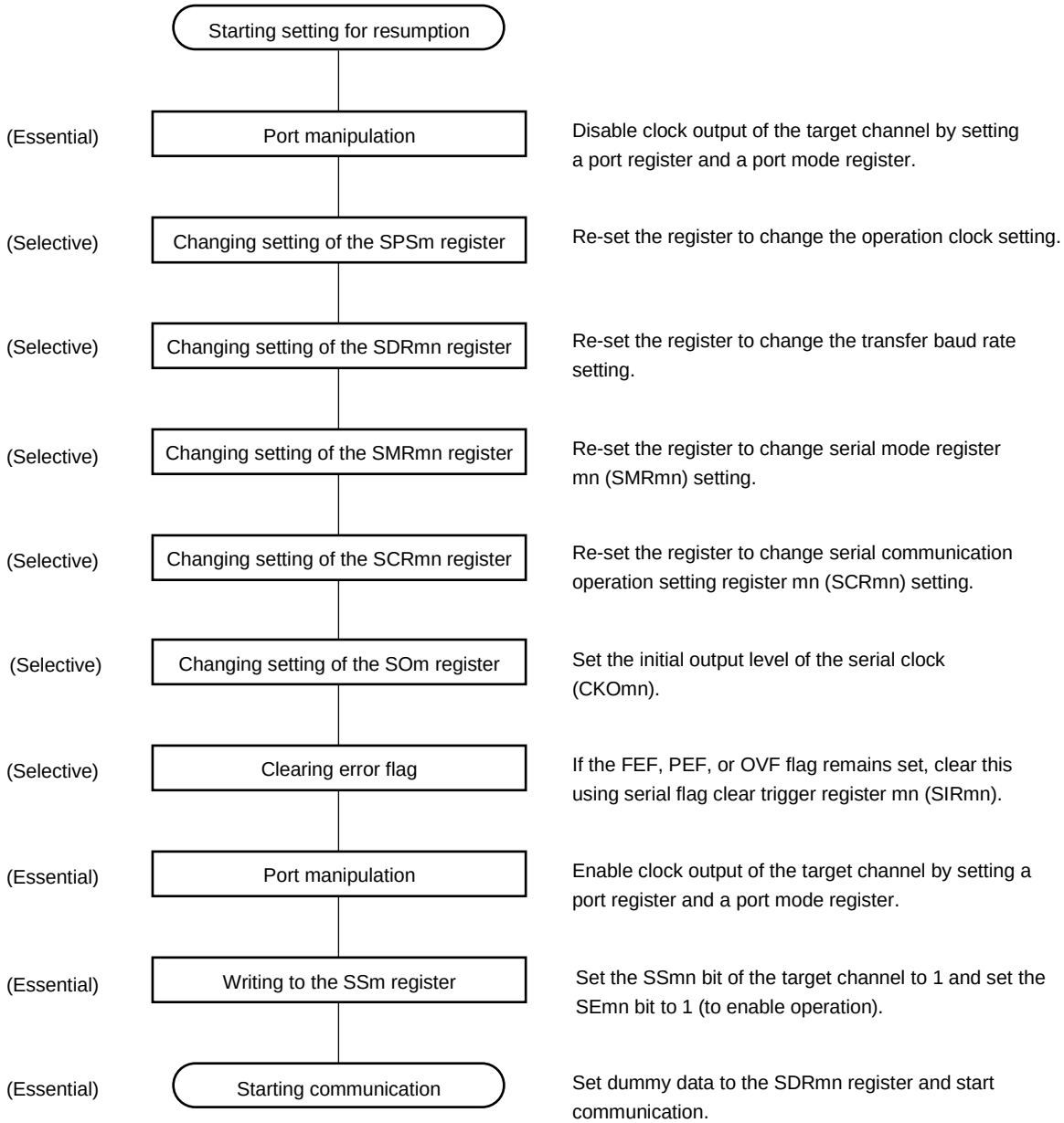
Figure 16-35. Procedure for Stopping Master Reception



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 16-36. Procedure for Resuming Master Reception**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

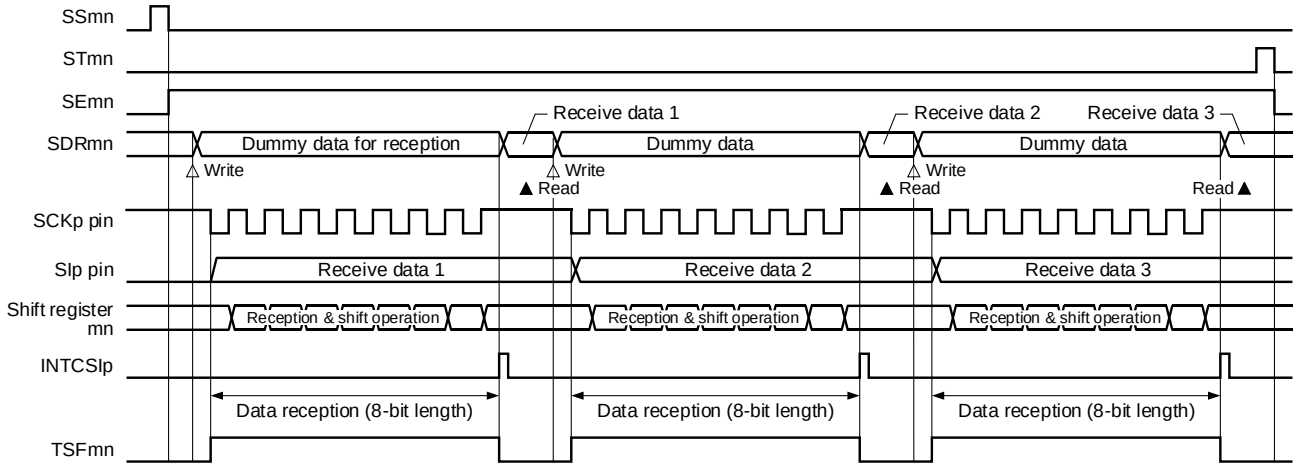
Figure 16-36. Procedure for Resuming Master Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

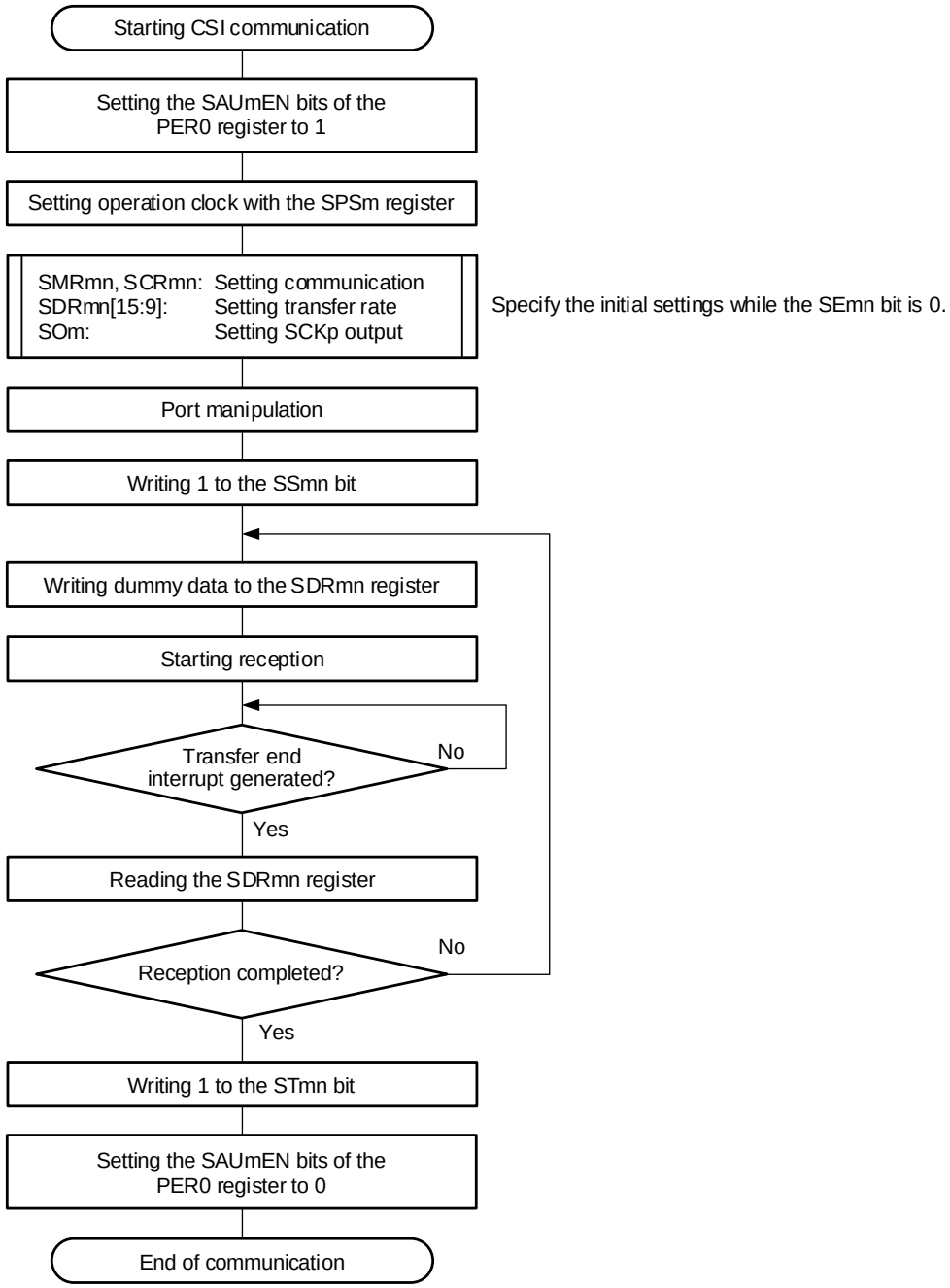
(3) Processing Flow (in Single-Reception Mode)

Figure 16-37. 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)

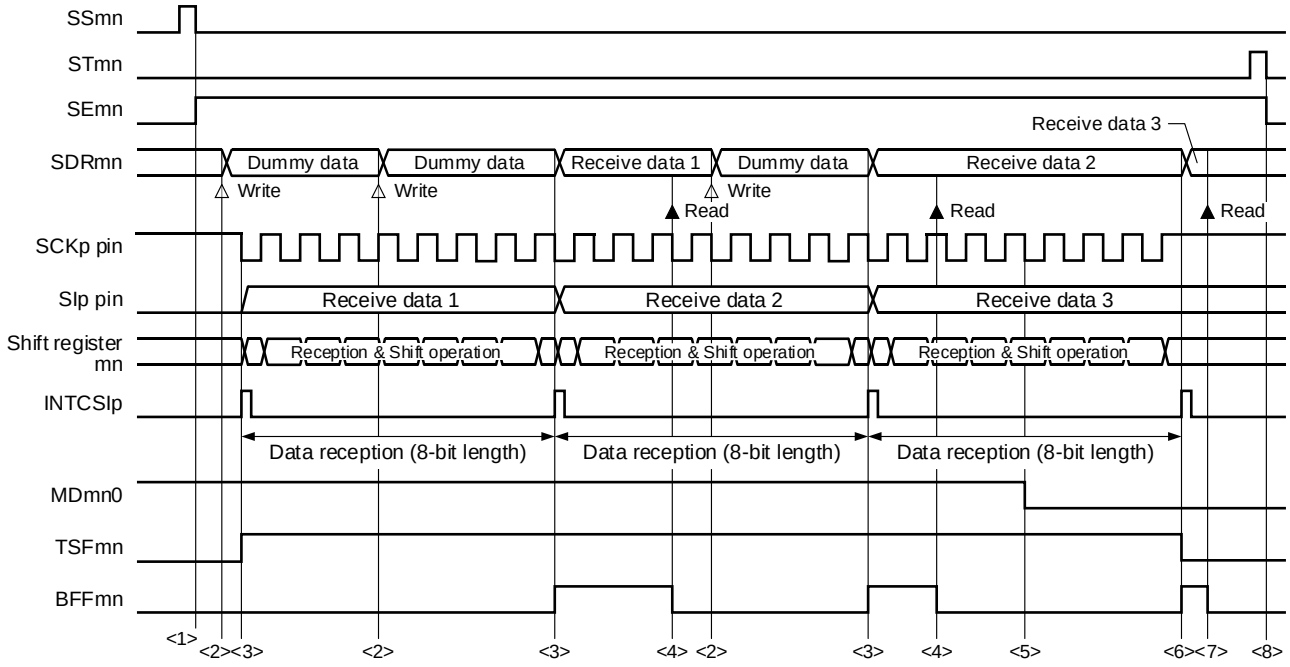
Figure 16-38. Flowchart of Master Reception (in Single-Reception Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

(4) Processing Flow (in Continuous Reception Mode)

Figure 16-39. Timing Chart of Master Reception (in Continuous Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

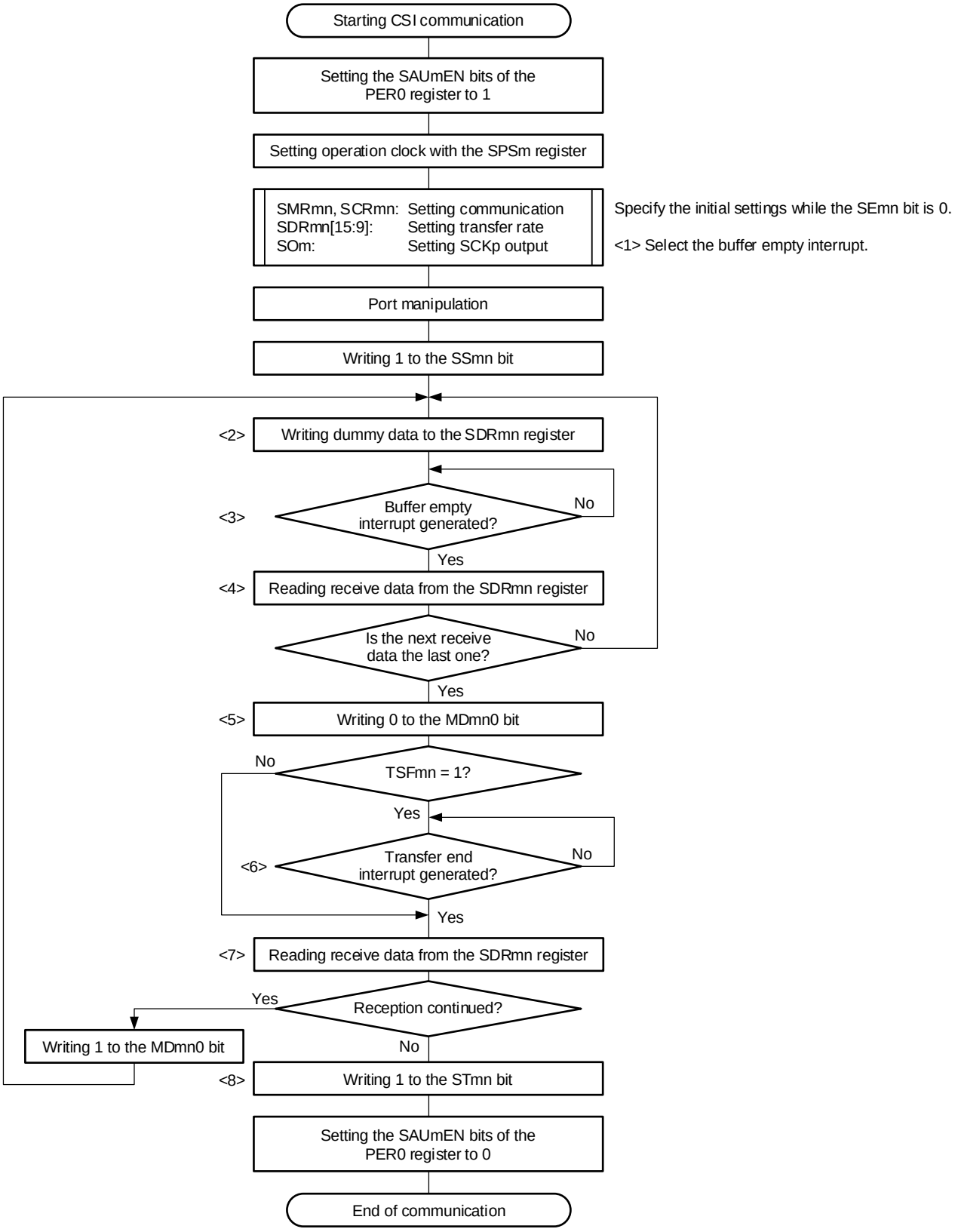


Caution MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the above figure correspond to <1> to <8> in **Figure 16-40**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 16-40. Flowchart of Master Reception (in Continuous Reception Mode)



Remarks 1. <1> to <8> in the above figure correspond to <1> to <8> in **Figure 16-39.**

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

16.5.3 Master Transmission/Reception

Master transmission/reception is an operation wherein this MCU outputs a transfer clock and transmits/receives data to/from the 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	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	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 • DAPmn = 0: Data I/O starts at the start of the serial clock operation. • 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 • CKPmn = 0: Forward • 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)

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 0 (operation clock selected by CKSmn bit).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 00B (CSI mode).
The MDmn0 bit selects interrupt source of channel n (0: transfer end, 1: buffer empty).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 11B (transmission/reception).
DAPmn and CKPmn bits select data and clock phase.
PTCmn1 and PTCmn0 bits set to 00B.
The DIRmn bit selects data transfer sequence (0: MSB first, 1: LSB first).
SLCmn1 and SLCmn0 bits are set to 00B.
DLSmn3 to DLSmn0 bits select data length (6H (7 bits) to FH (16 bits)).

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set baud rate to bits 15 to 9 of the SDRmn register.
When operation is in progress (SEmn = 1): Transmit data setting/receive data in the SDRmn register.

Note Lower 8 bit in the SDRmn register: SDRpL register.

(d) Serial output register m (SOm)

Set CKOmn bit and SOmn bit of the target channel.
The CKOmn bit is set to 1 when CKPmn bit of the SCRmn register is 0 and 0 when CKPmn bit is 1 to start communication.

(e) Serial output enable register m (SOEm)

Set the SOEmn bit of the target channel to 1 (serial output enable).

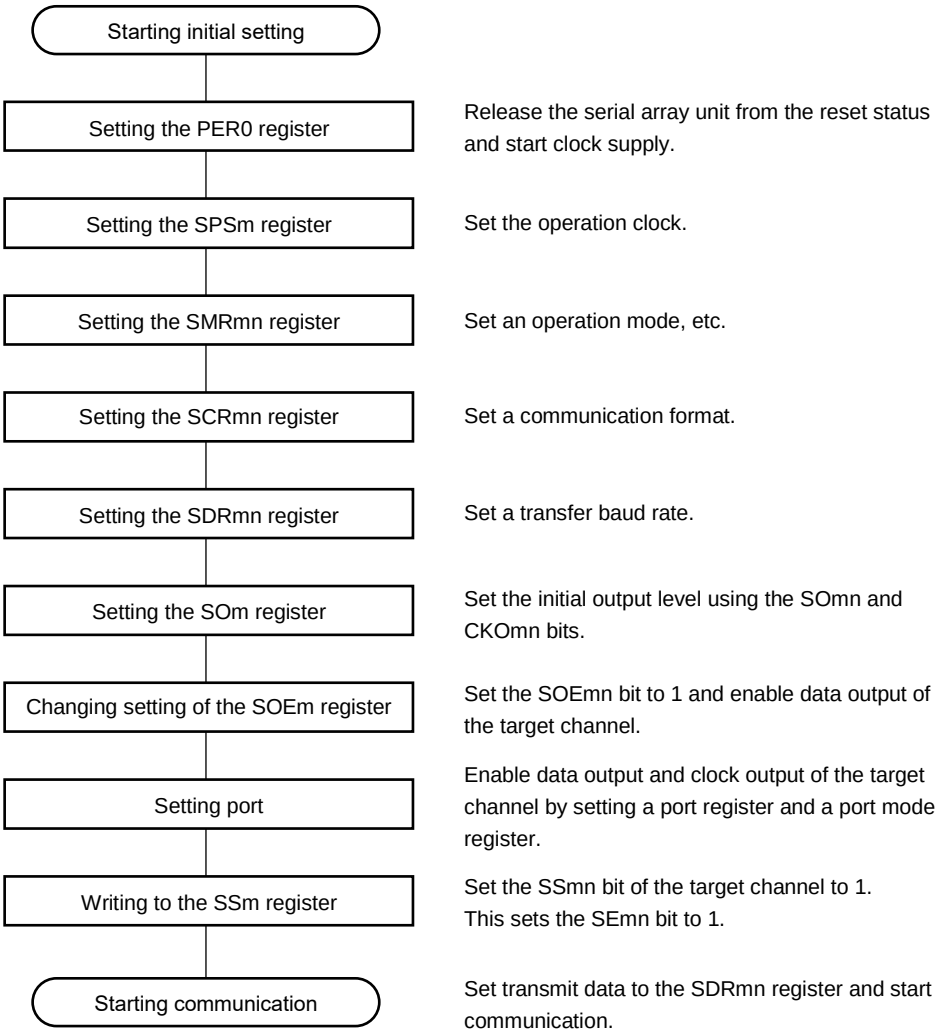
(f) Serial channel start register m (SSm)

Set the SSmn bit of the target channel to 1 (operation start).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

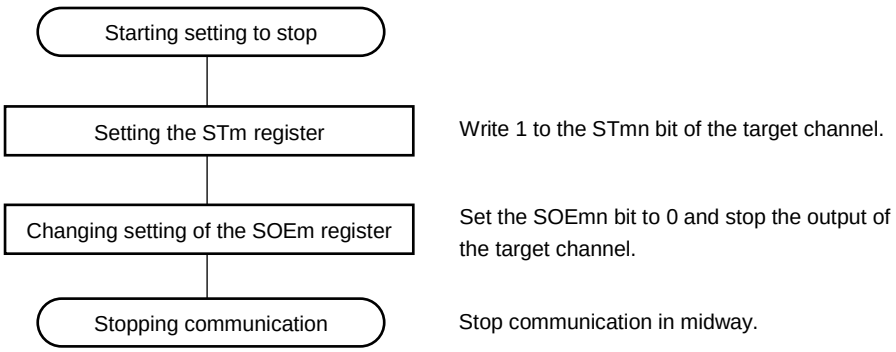
(2) Operation Procedure

Figure 16-41. Initial Setting Procedure for Master Transmission/Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

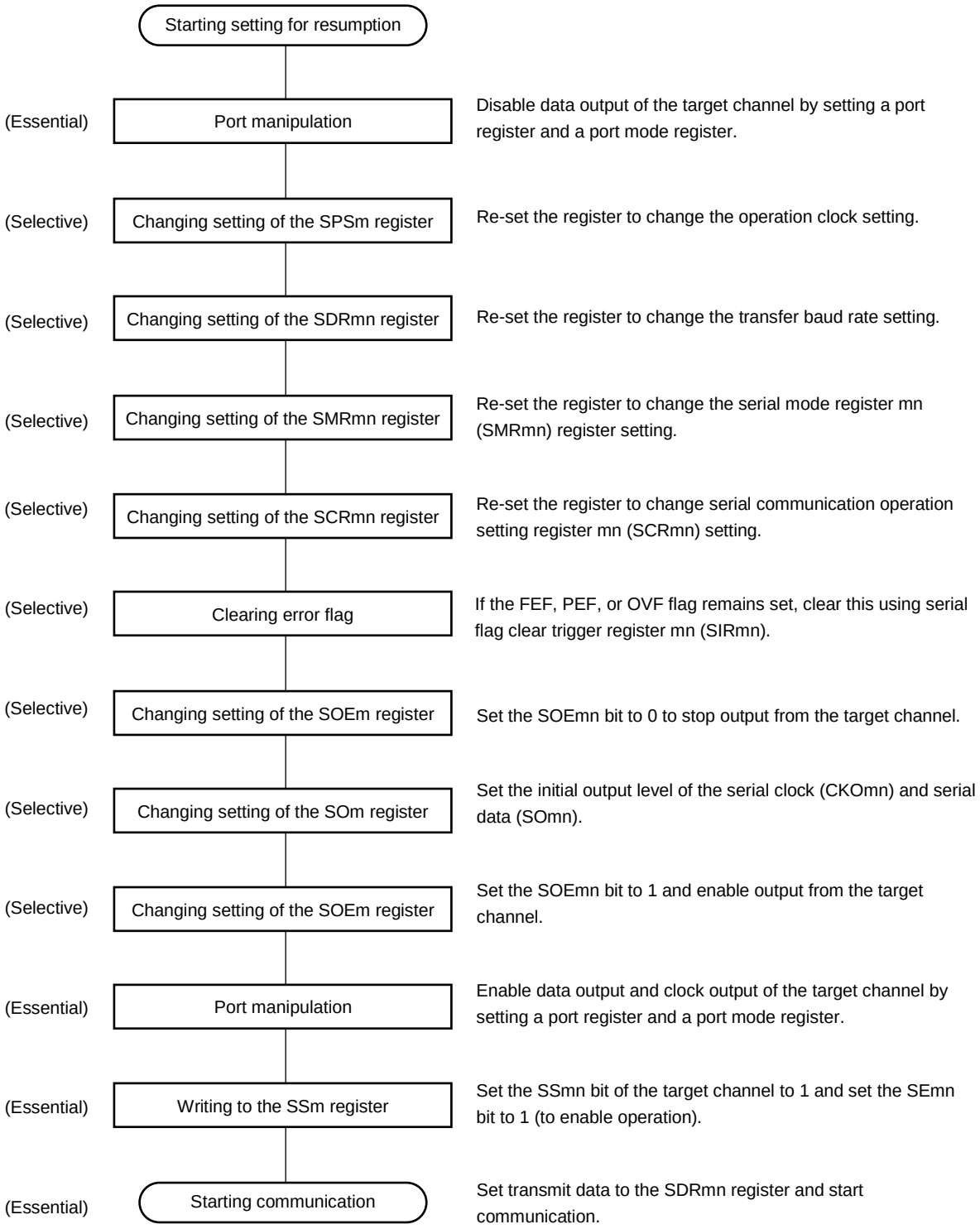
Figure 16-42. Procedure for Stopping Master Transmission/Reception



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 16-43. Procedure for Resuming Master Transmission/Reception**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

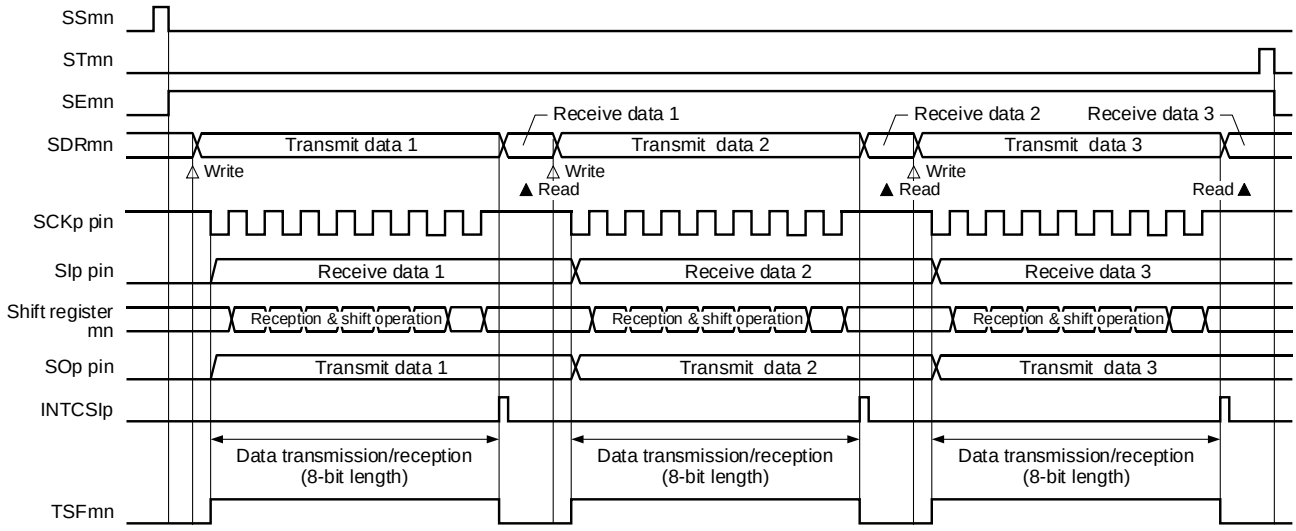
Figure 16-43. Procedure for Resuming Master Transmission/Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

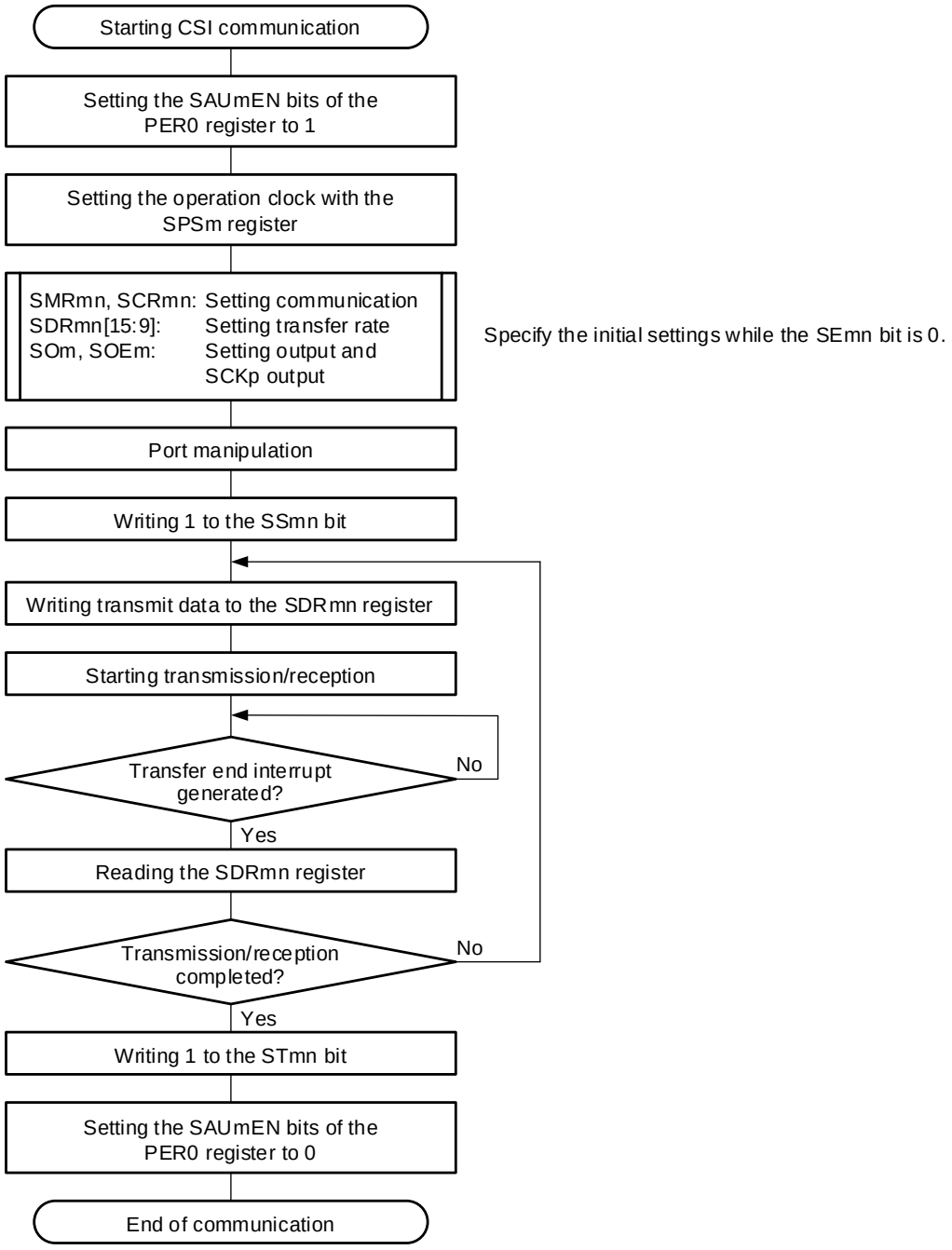
(3) Processing Flow (in Single-Transmission/Reception Mode)

Figure 16-44. 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)

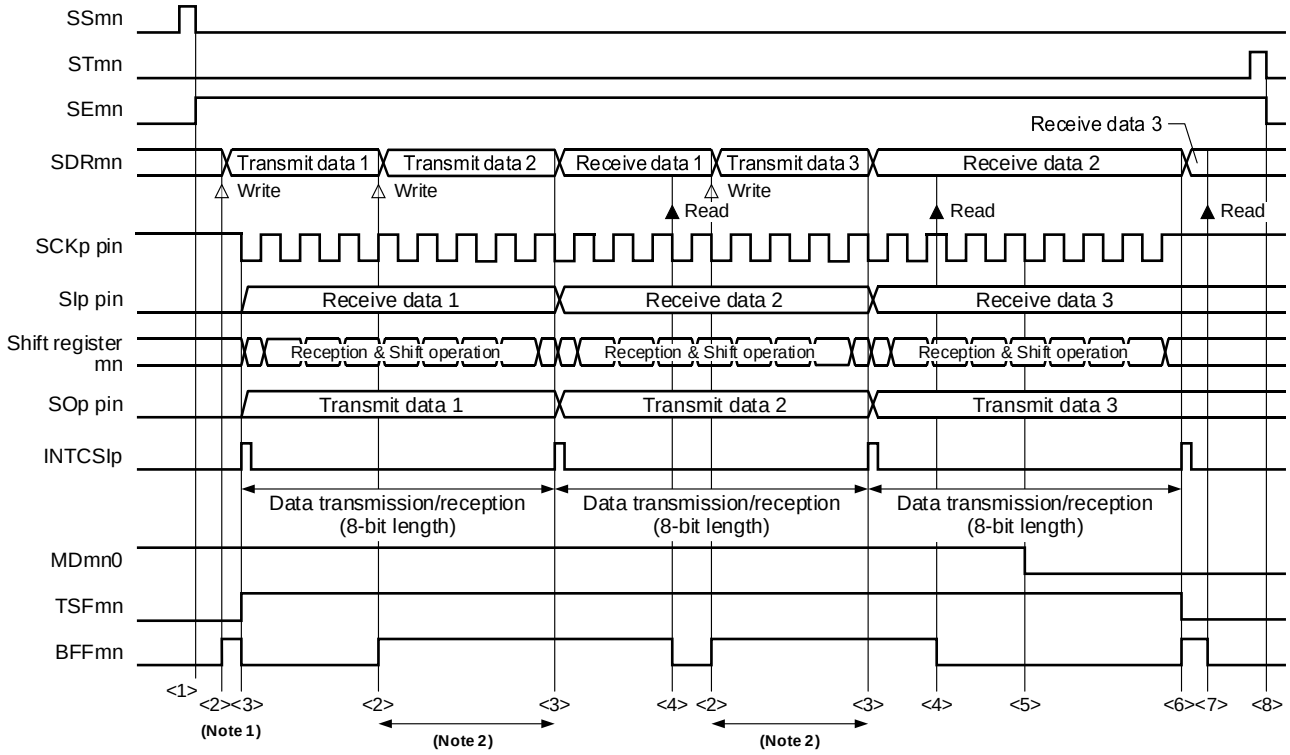
Figure 16-45. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

(4) Processing Flow (in Continuous Transmission/Reception Mode)

Figure 16-46. 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.

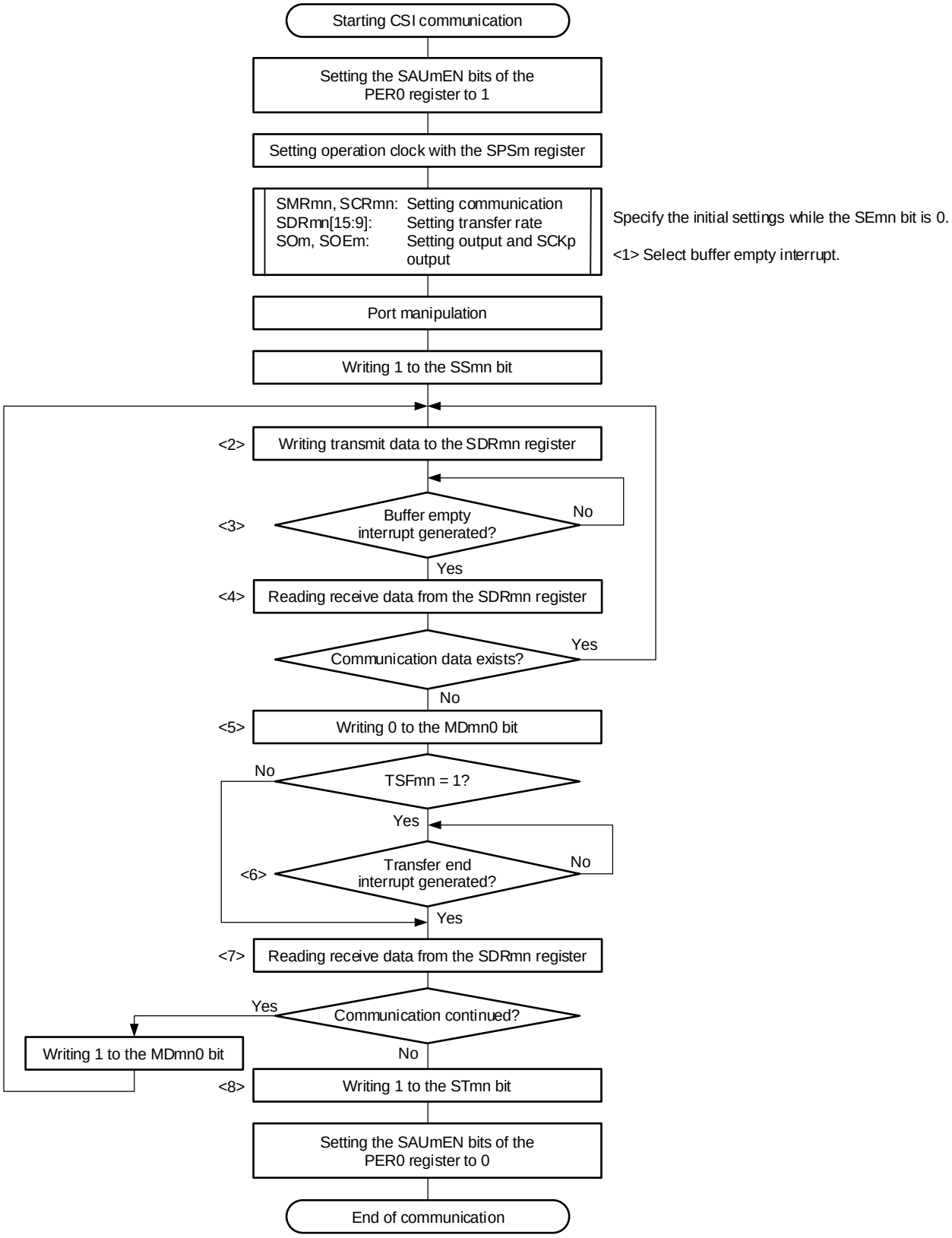
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution 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 above figure correspond to <1> to <8> in **Figure 16-47**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 16-47. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remarks 1. <1> to <8> in the above figure correspond to <1> to <8> in **Figure 16-46**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

16.5.4 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	SCK00, SO00	SCK01, SO01	SCK10, SO10	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 • DAPmn = 0: Data output starts from the start of the serial clock operation. • 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 • CKPmn = 0: Forward • CKPmn = 1: Reverse			
Data direction	MSB or LSB first			

Notes 1. Because the external serial clock input to the SCK00, SCK01, SCK10, and 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)

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 1 (clock input from the SCKp pin).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 00B (CSI mode).
The MDmn0 bit selects interrupt source of channel n (0: transfer end, 1: buffer empty).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 10B (transmission only).
DAPmn and CKPmn bits select data and clock phase.
PTCmn1 and PTCmn0 bits set to 00B.
The DIRmn bit selects data transfer sequence (0: MSB first, 1: LSB first).
SLCmn1 and SLCmn0 bits are set to 00B.
DLSmn3 to DLSmn0 bits select data length (6H (7 bits) to FH (16 bits)).

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set bits 15 to 9 of the SDRmn register to 0000000B.
When operation is in progress (SEmn = 1): Set the transmit data in the SDRmn register.

Note Lower 8 bit in the SDRmn register: SDRpL register.

(d) Serial output register m (SOm)

Set the SOmn bit of the target channel.

(e) Serial output enable register m (SOEm)

Set the SOEmn bit of the target channel to 1 (serial output enable).

(f) Serial channel start register m (SSm)

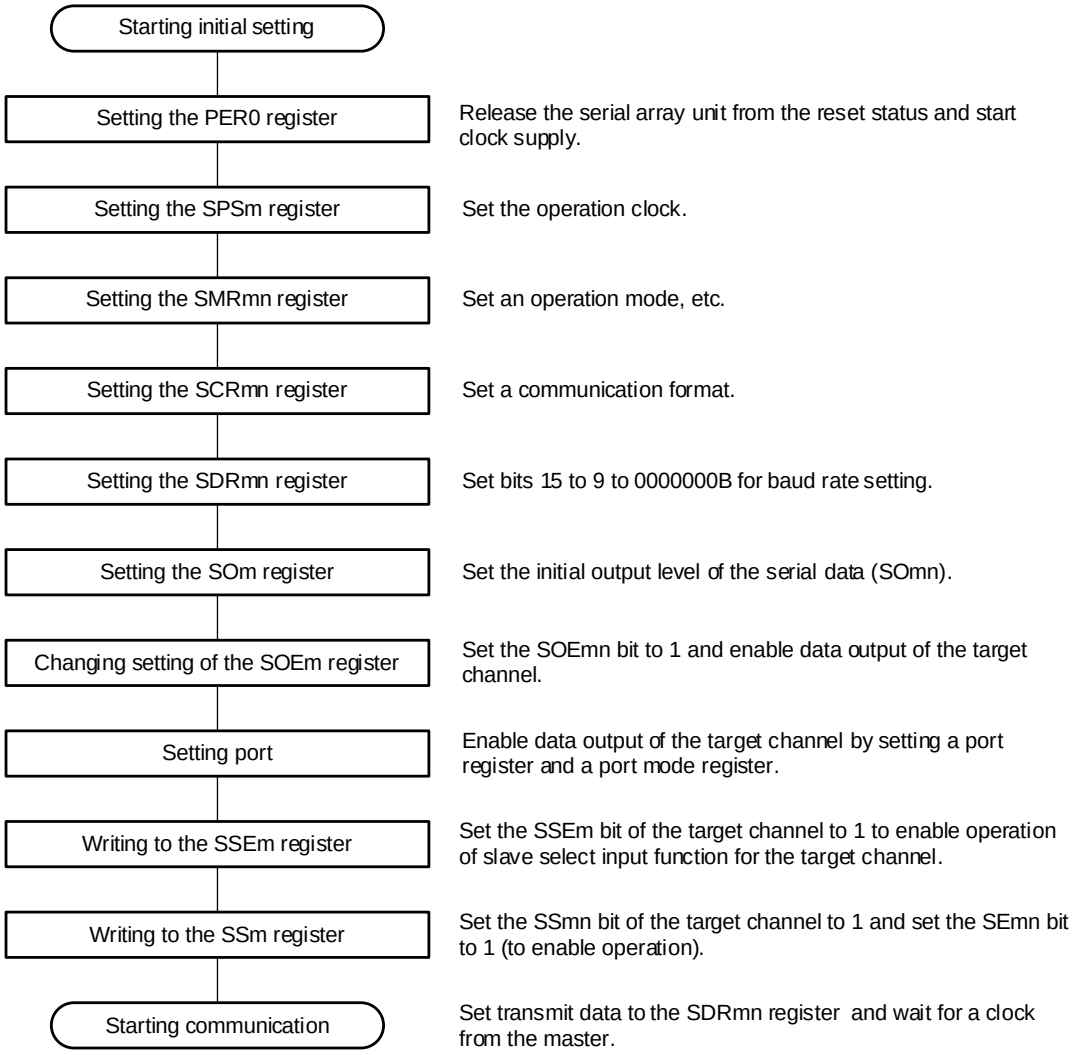
Set the SSmn bit of the target channel to 1 (operation start).

Caution Be sure to set the transmit data to SDRmn register before the serial clock is input from the master.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

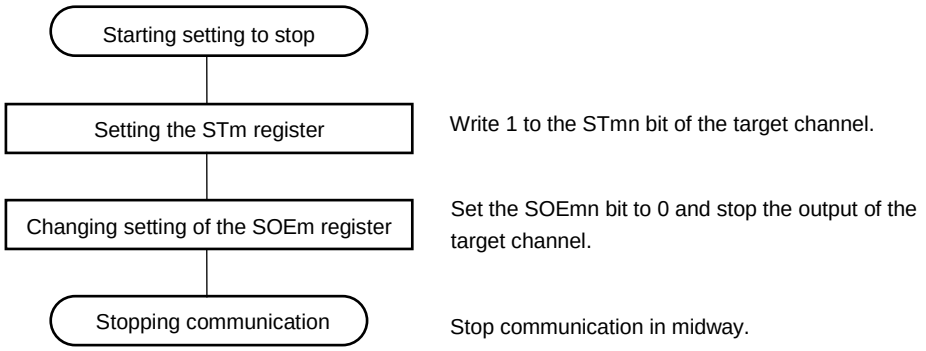
(2) Operation Procedure

Figure 16-48. Initial Setting Procedure for Slave Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

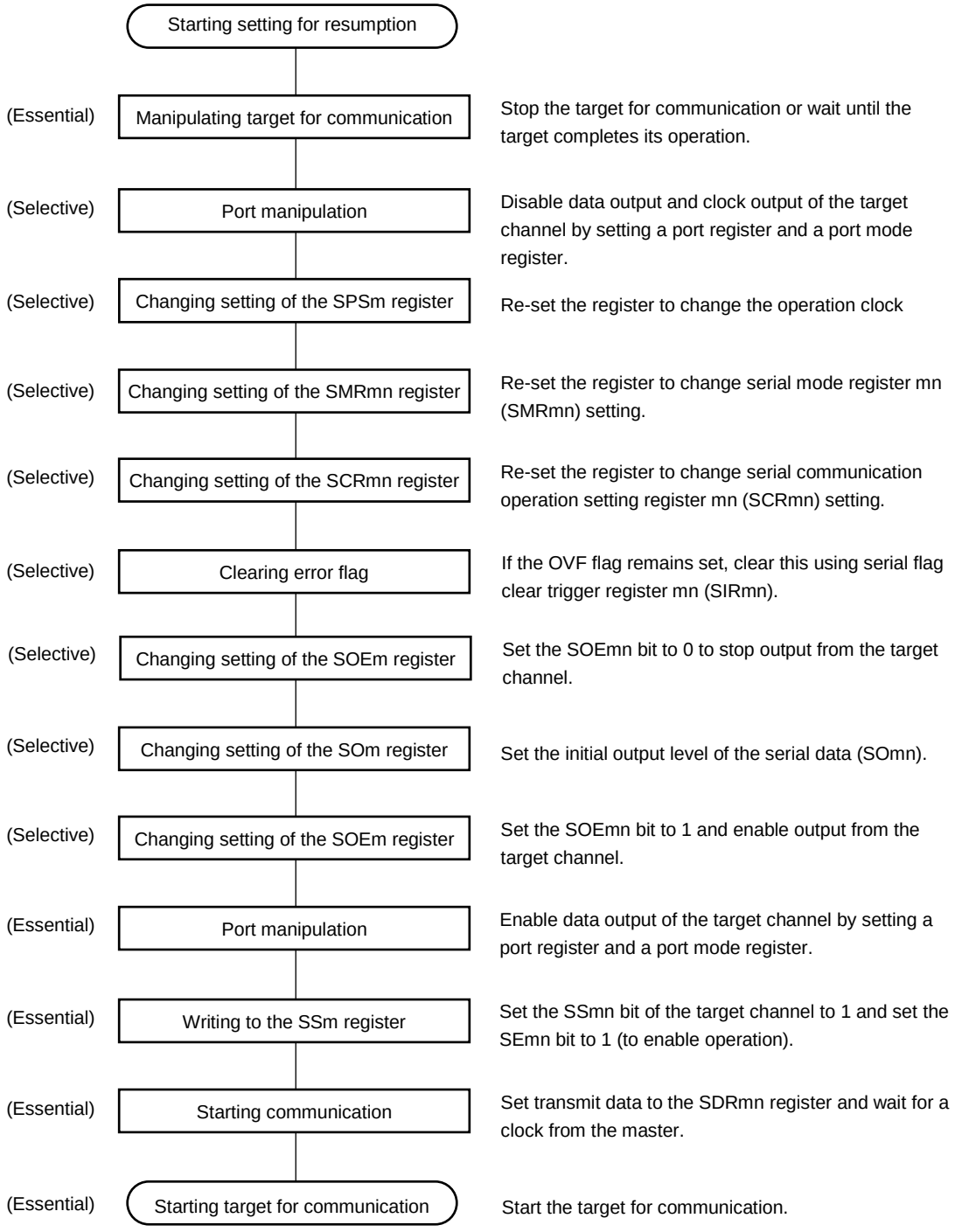
Figure 16-49. Procedure for Stopping Slave Transmission



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 16-50. Procedure for Resuming Slave Transmission**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

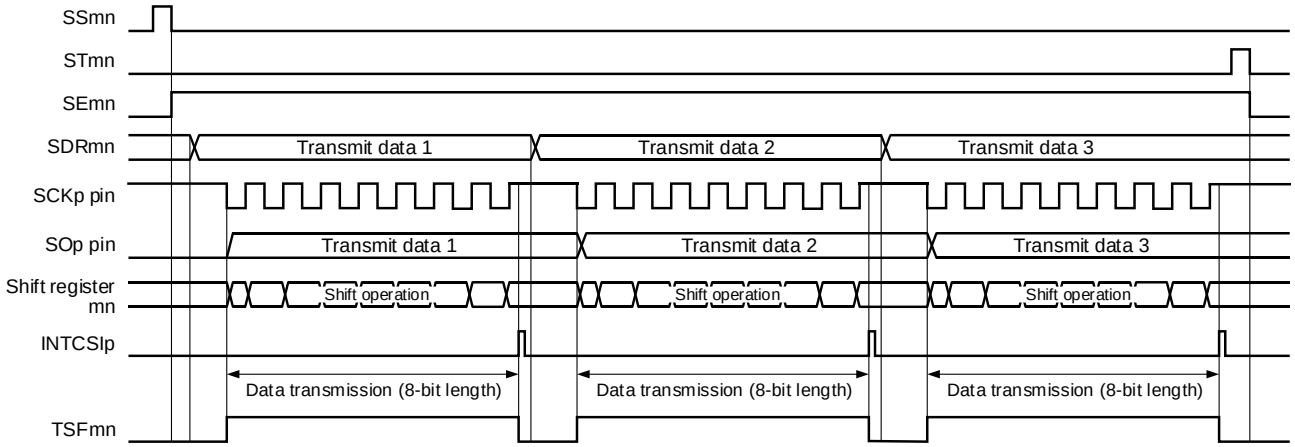
Figure 16-50. Procedure for Resuming Slave Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

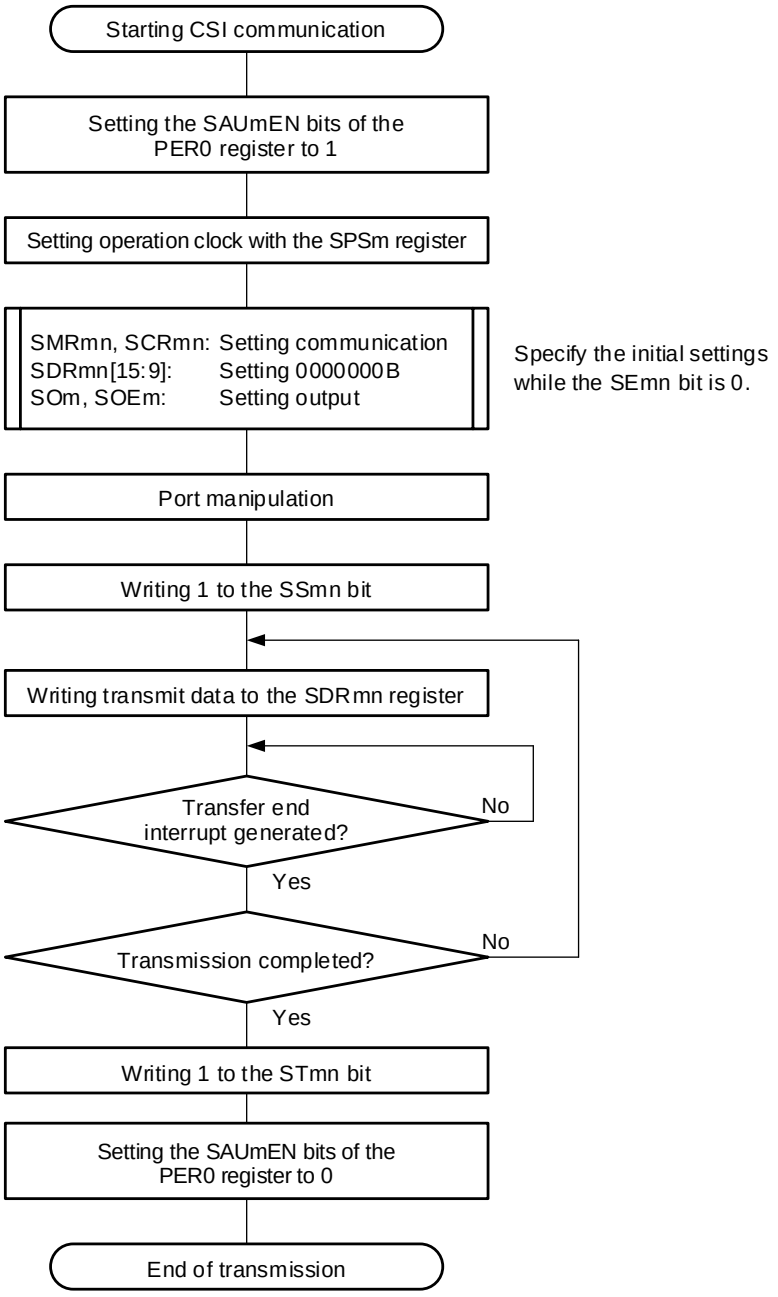
(3) Processing Flow (in Single-Transmission Mode)

Figure 16-51. 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)

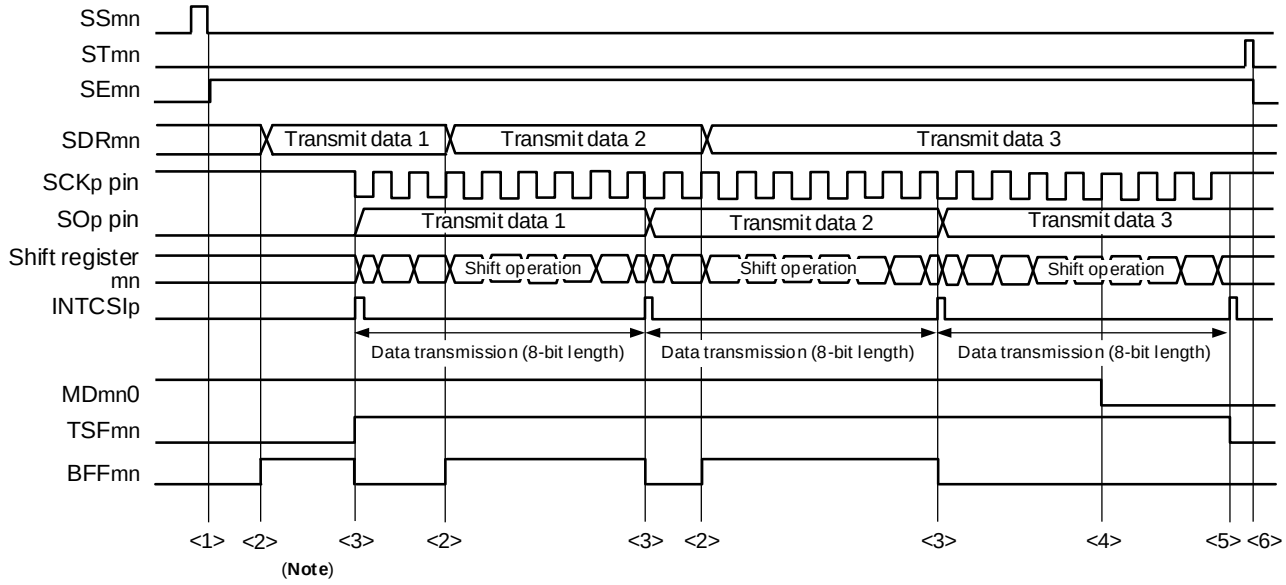
Figure 16-52. Flowchart of Slave Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(4) Processing Flow (in Continuous Transmission Mode)

Figure 16-53. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



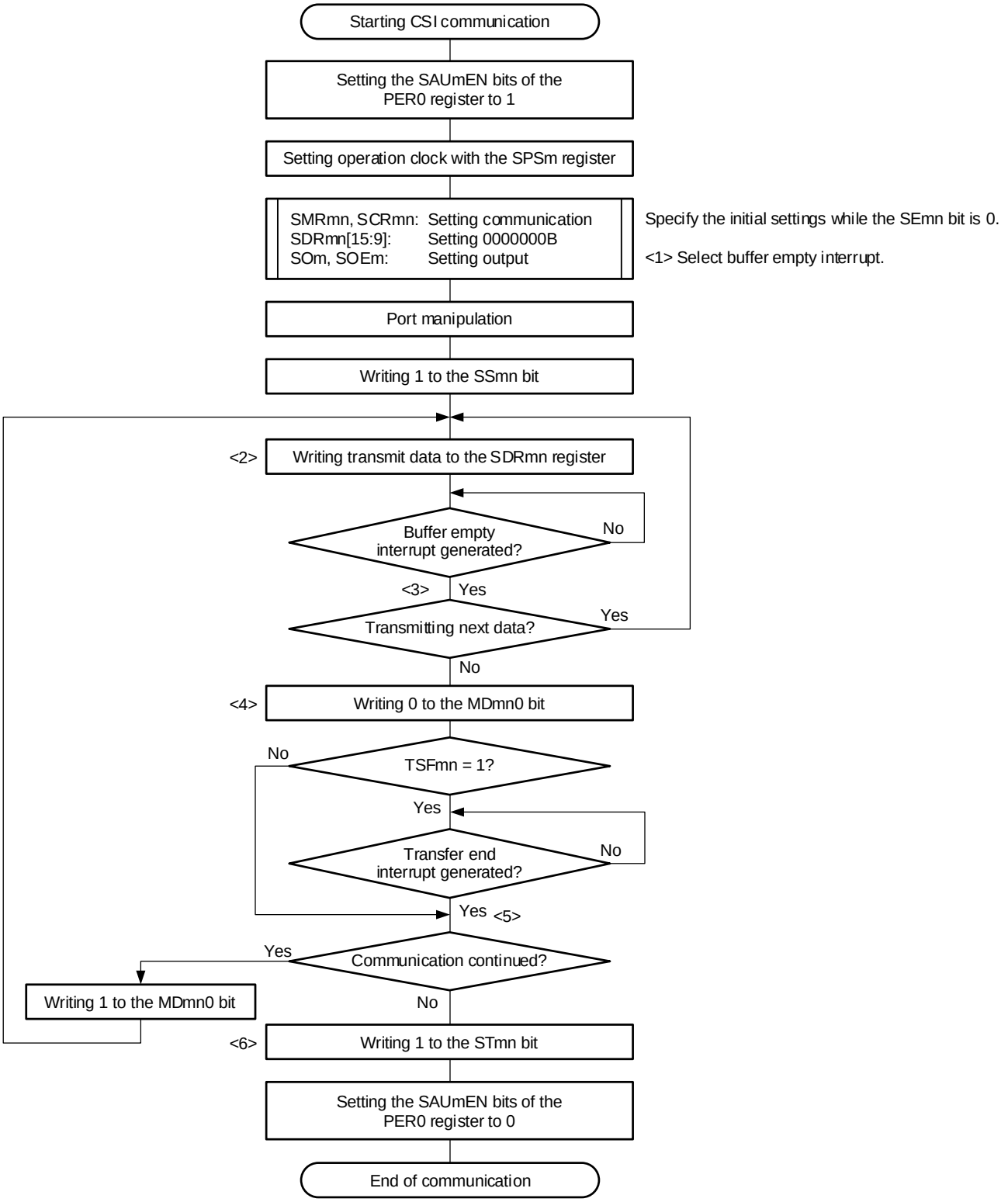
Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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.

Remarks 1. <1> to <6> in the above figure correspond to <1> to <6> in **Figure 16-54**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 16-54. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remarks 1. <1> to <6> in the above figure correspond to <1> to <6> in **Figure 16-53**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.5.5 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	SCK00, SI00	SCK01, SI01	SCK10, SI10	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 • DAPmn = 0: Data input starts from the start of the serial clock operation. • 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 • CKPmn = 0: Forward • CKPmn = 1: Reverse			
Data direction	MSB or LSB first			

Notes 1. Because the external serial clock input to the SCK00, SCK01, SCK10, and 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)

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 1 (clock input from the SCKp pin).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 00B (CSI mode).
The MDmn0 bit is set to 0 (interrupt source is transfer end).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 01B (reception only).
DAPmn and CKPmn bits select data and clock phase.
PTCmn1 and PTCmn0 bits set to 00B.
The DIRmn bit selects data transfer sequence (0: MSB first, 1: LSB first).
SLCmn1 and SLCmn0 bits are set to 00B.
DLSmn3 to DLSmn0 bits select data length (6H (7 bits) to FH (16 bits)).

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set bits 15 to 9 of the SDRmn register to 0000000B.
When operation is in progress (SEmn = 1): Receive data

Note Lower 8 bit in the SDRmn register: SDRpL register.

(d) Serial output register m (SOm)

The register that not used in this mode.

(e) Serial output enable register m (SOEm)

The register that not used in this mode.

(f) Serial channel start register m (SSm)

Set the SSmn bit of the target channel to 1 (operation start).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

(2) Operation Procedure

Figure 16-55. Initial Setting Procedure for Slave Reception

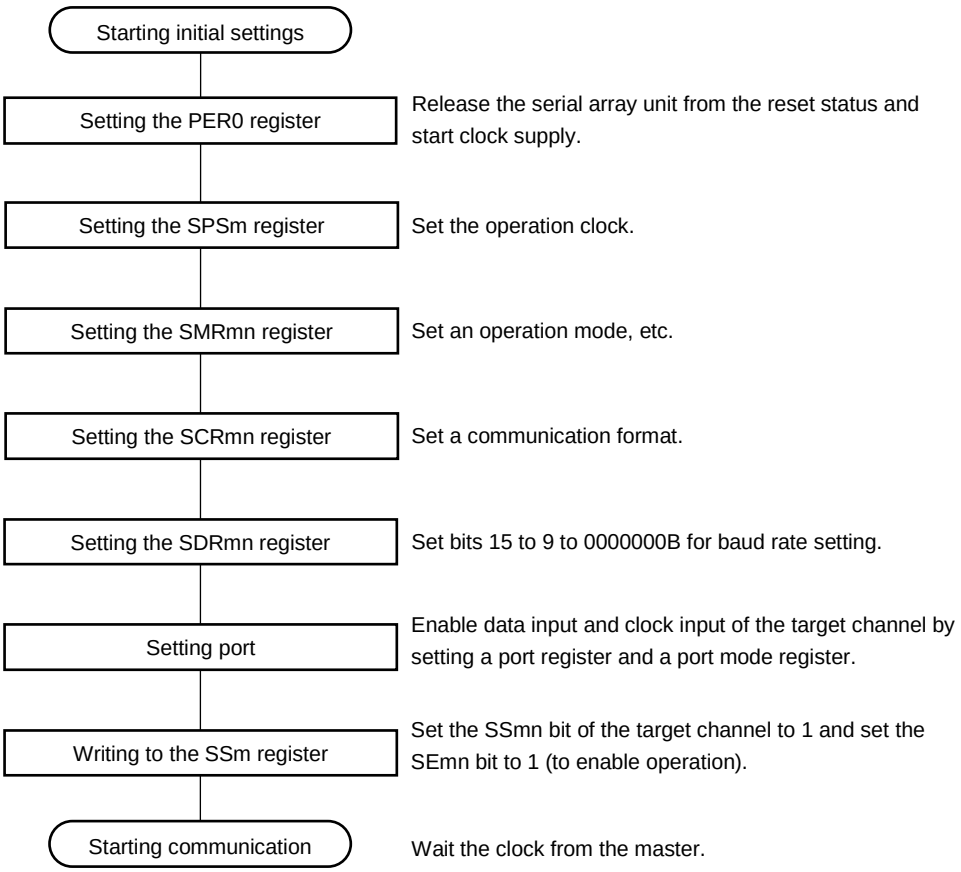
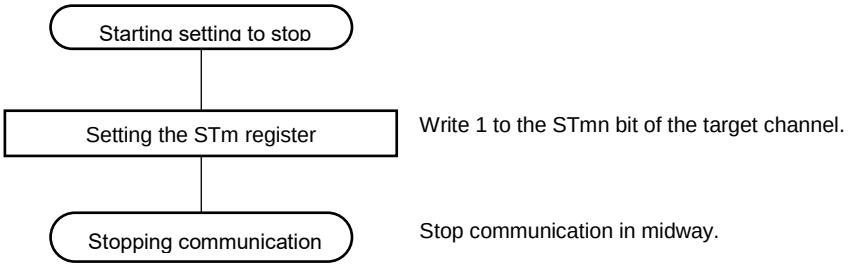
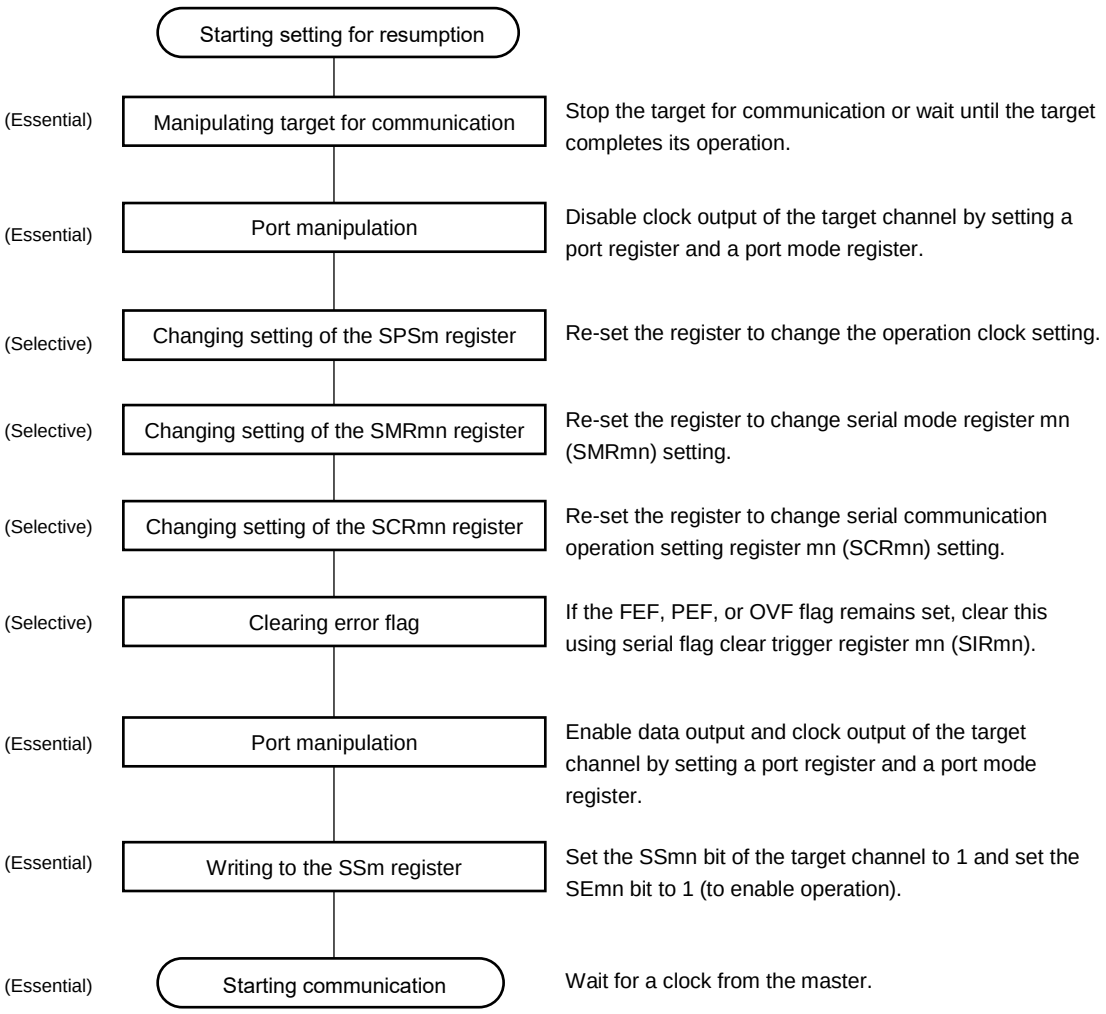


Figure 16-56. Procedure for Stopping Slave Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

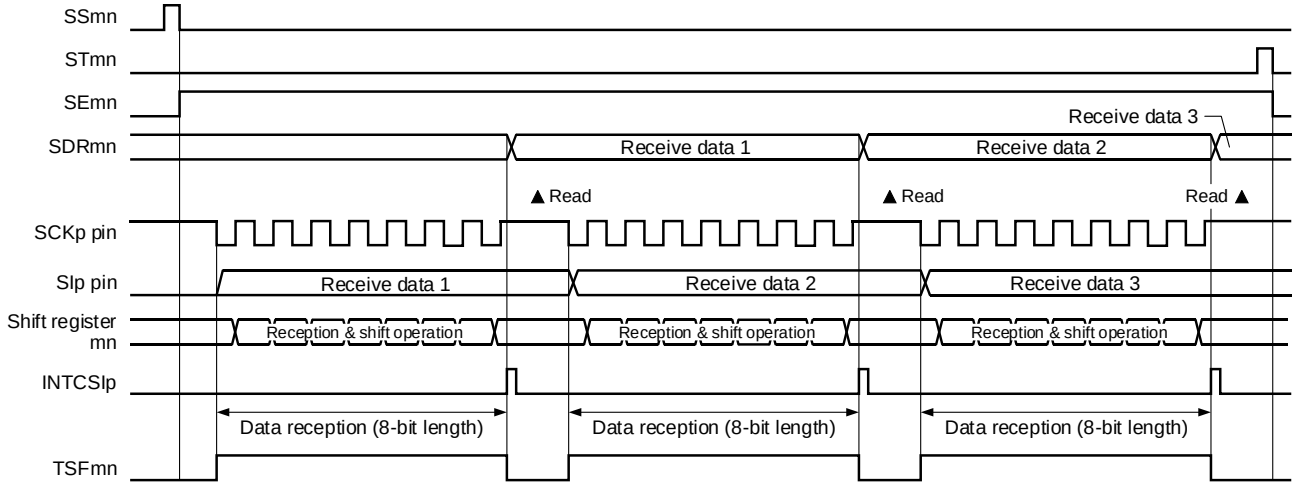
Figure 16-57. Procedure for Resuming Slave Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

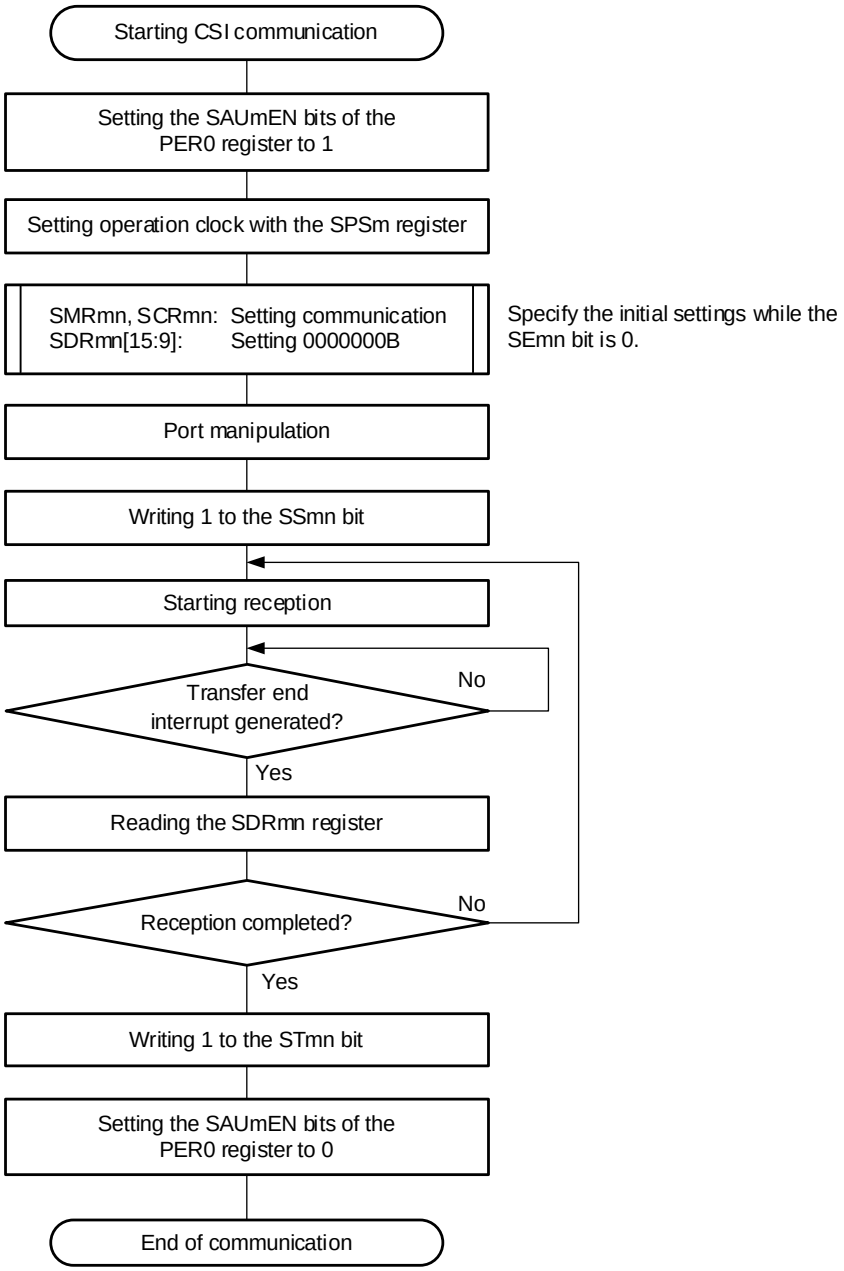
(3) Processing Flow (in Single-Reception Mode)

Figure 16-58. 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)

Figure 16-59. Flowchart of Slave Reception (in Single-Reception Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.5.6 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	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	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 • DAPmn = 0: Data I/O starts from the start of the serial clock operation. • 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 • CKPmn = 0: Forward • CKPmn = 1: Reverse			
Data direction	MSB or LSB first			

Notes 1. Because the external serial clock input to the SCK00, SCK01, SCK10, and 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)

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 1 (clock input from the SCKp pin).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 00B (CSI mode).
The MDmn0 bit selects interrupt source of channel n (0: transfer end, 1: buffer empty).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 11B (transmission/reception).
DAPmn and CKPmn bits select data and clock phase.
PTCmn1 and PTCmn0 bits set to 00B.
The DIRmn bit selects data transfer sequence (0: MSB first, 1: LSB first).
SLCmn1 and SLCmn0 bits are set to 00B.
DLSmn3 to DLSmn0 bits select data length (6H (7 bits) to FH (16 bits)).

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set bits 15 to 9 of the SDRmn register to 0000000B.
When operation is in progress (SEmn = 1): Transmit data setting/receive data in the SDRmn register.

Note Lower 8 bit in the SDRmn register: SDRpL register.

(d) Serial output register m (SOm)

Set the SOmn bit of the target channel.

(e) Serial output enable register m (SOEm)

Set the SOEmn bit of the target channel to 1 (serial output enable).

(f) Serial channel start register m (SSm)

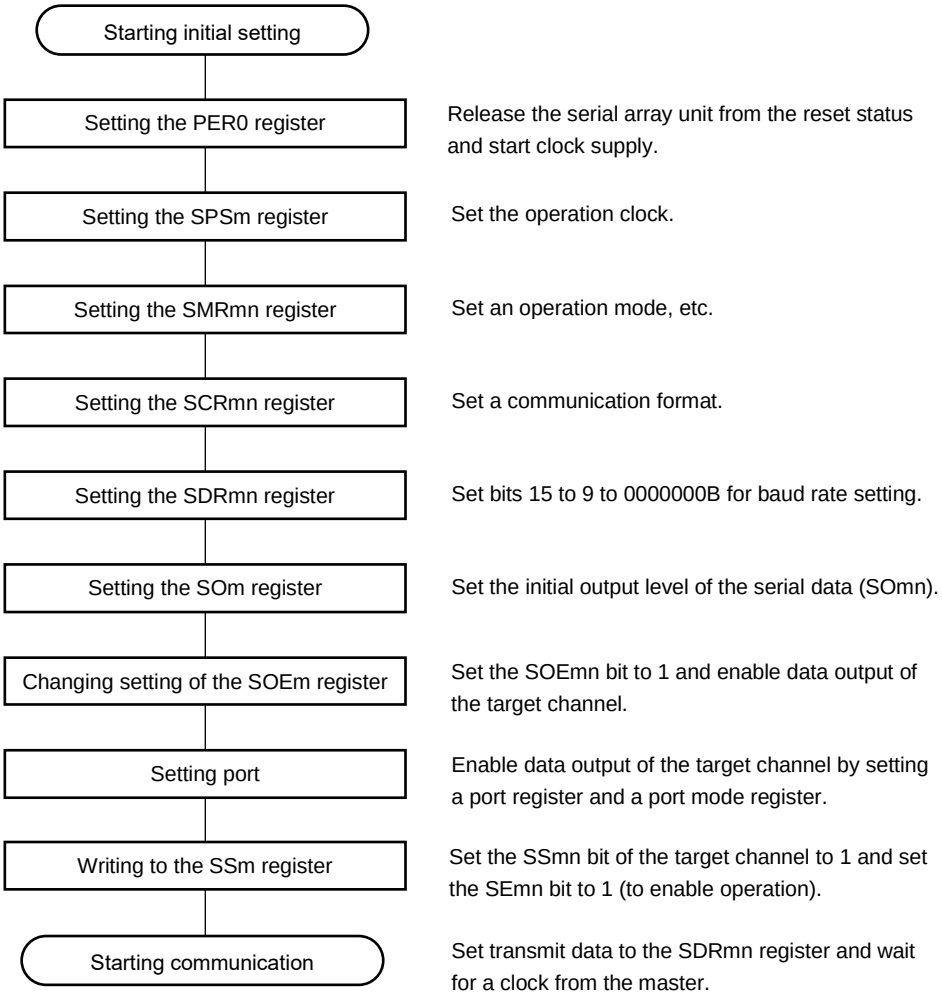
Set the SSmn bit of the target channel to 1 (operation start).

Caution Be sure to set the transmit data to SDRmn register before the serial clock is input from the master.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

(2) Operation Procedure

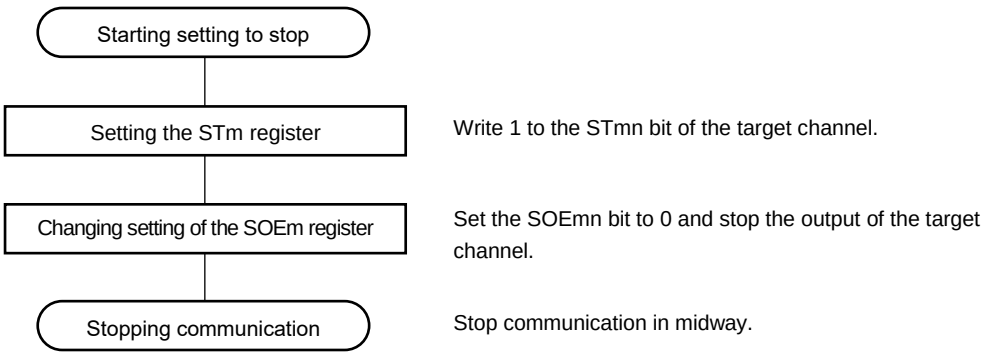
Figure 16-60. Initial Setting Procedure for Slave Transmission/Reception



Caution Be sure to set the transmit data to SDRmn register before the serial clock is input from the master.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

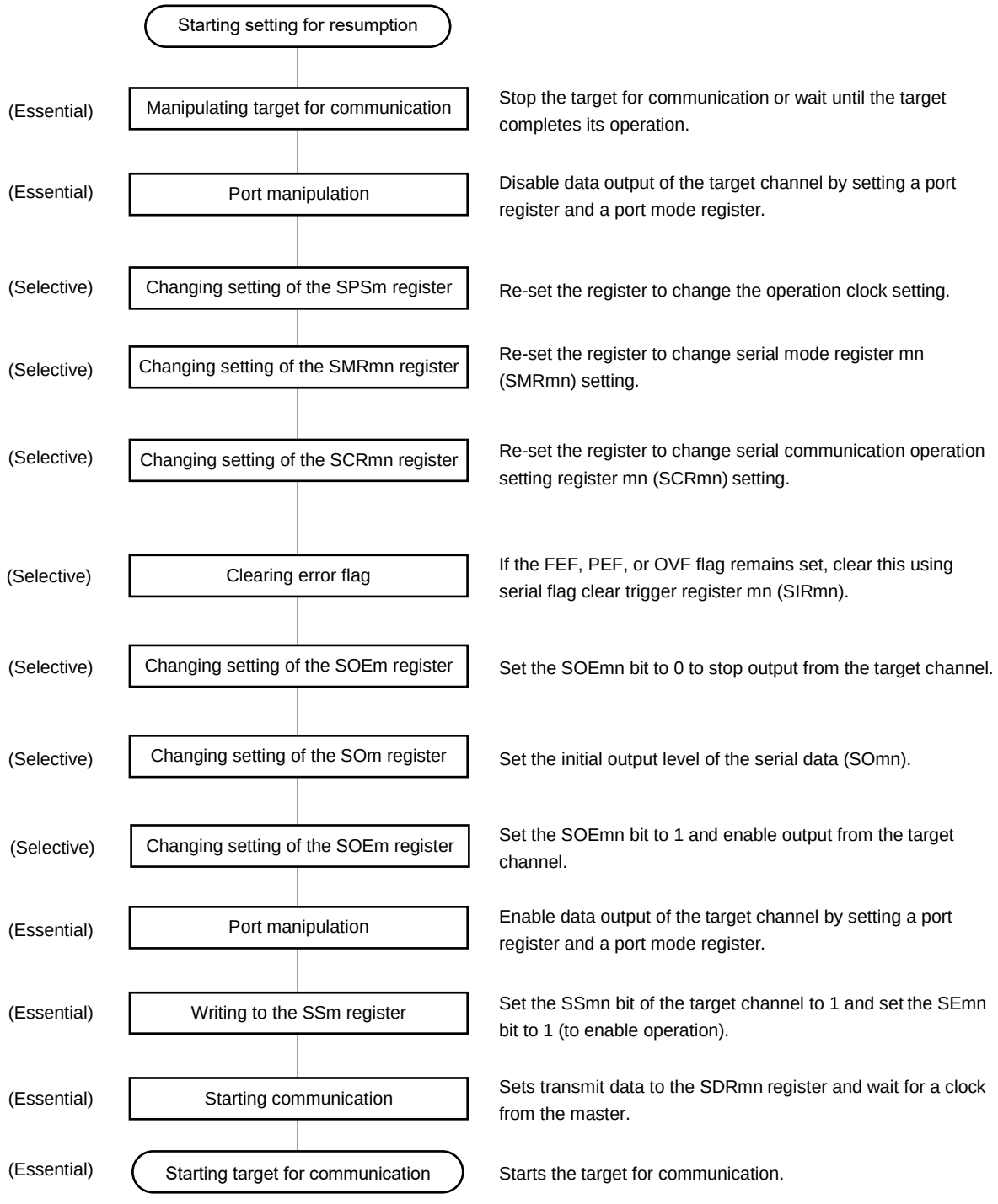
Figure 16-61. Procedure for Stopping Slave Transmission/Reception



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 16-62. Procedure for Resuming Slave Transmission/Reception**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

Figure 16-62. Procedure for Resuming Slave Transmission/Reception

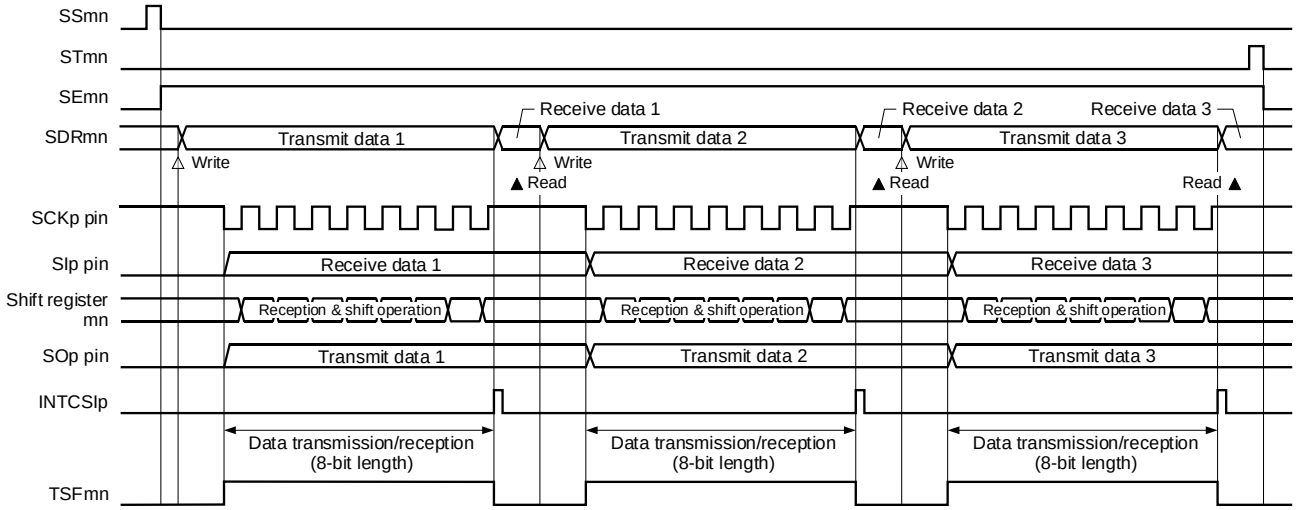


Caution Be sure to set the transmit data to SDRmn register before the serial clock is input from the master.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

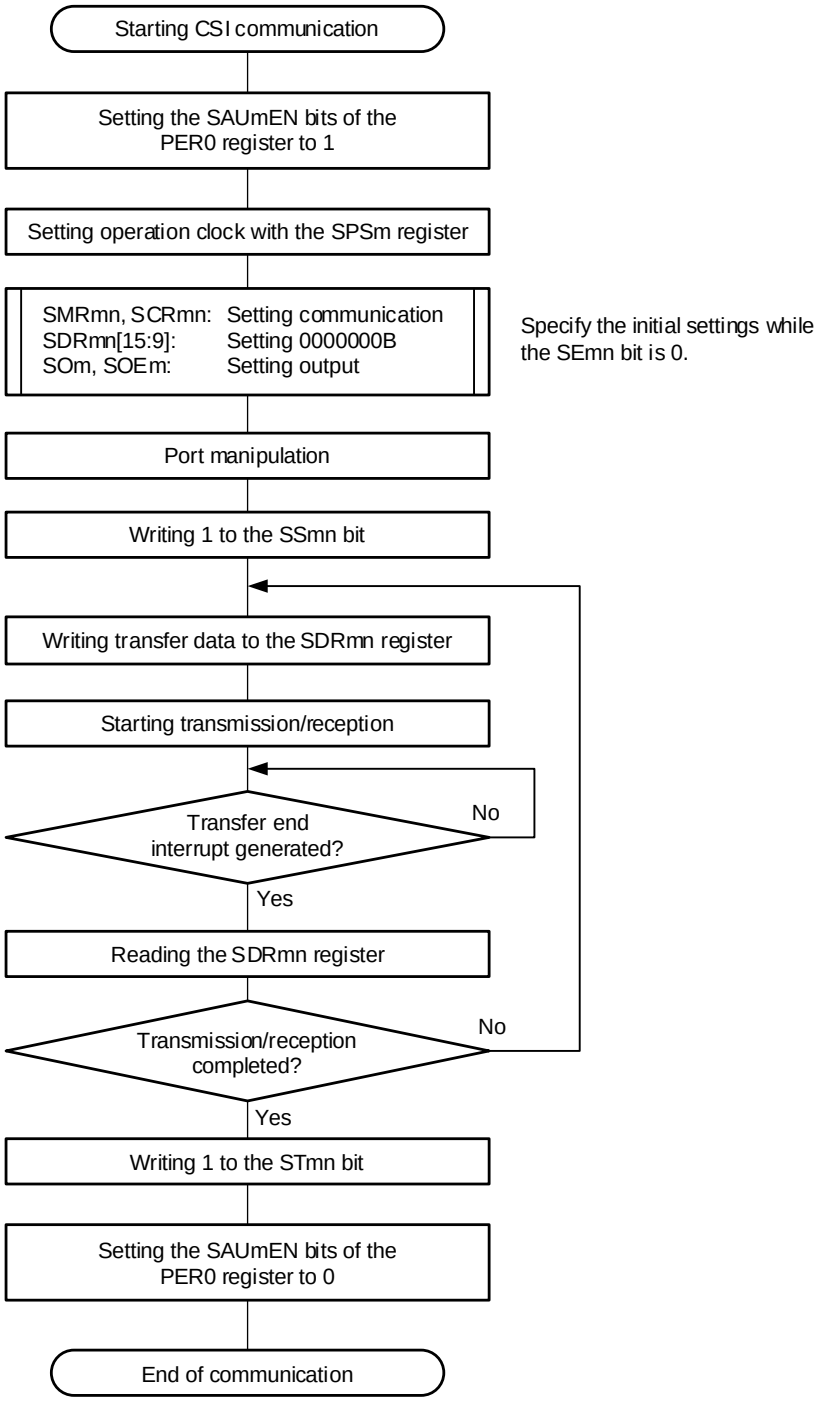
(3) Processing Flow (in Single-Transmission/Reception Mode)

Figure 16-63. 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)

Figure 16-64. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)

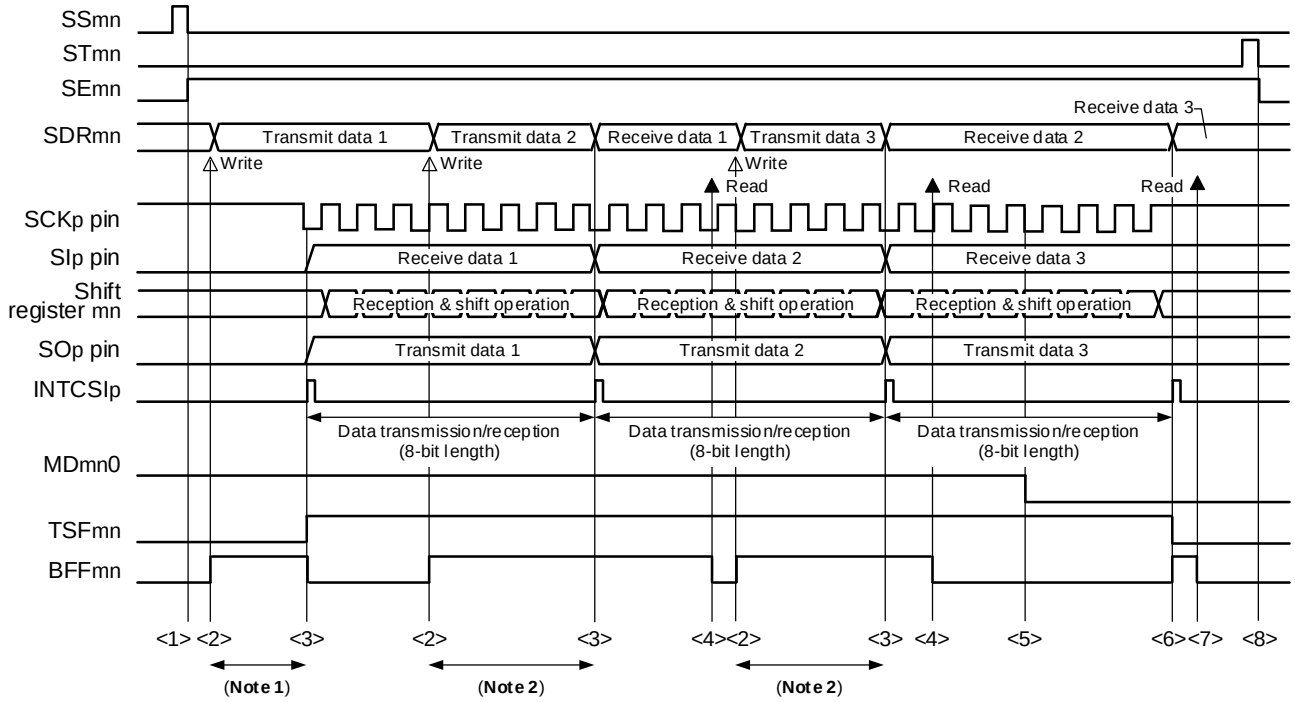


Caution Be sure to set the transmit data to SDRmn register before the serial clock is input from the master.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(4) Processing Flow (in Continuous Transmission/Reception Mode)

Figure 16-65. 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.

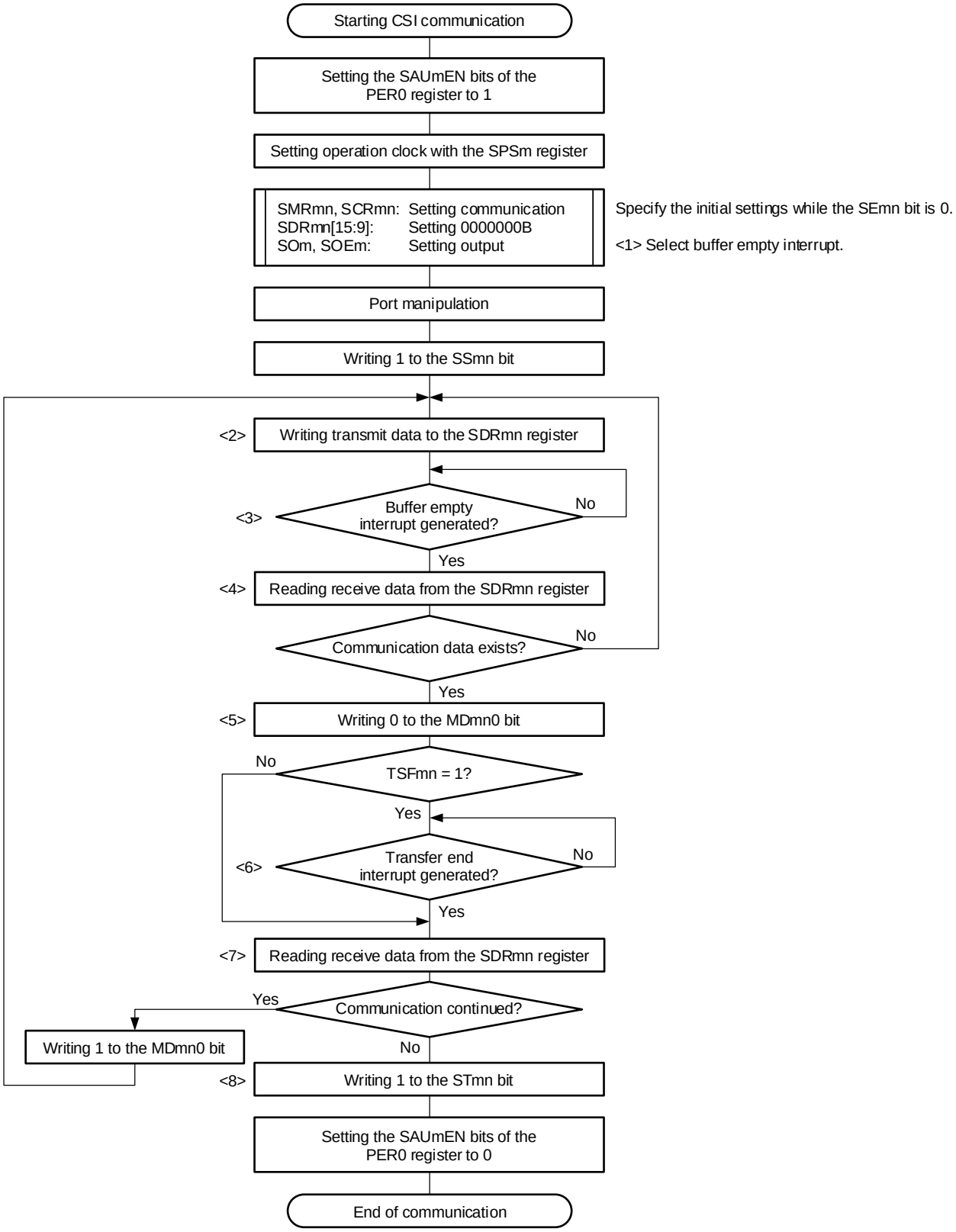
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution 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 above figure correspond to <1> to <8> in **Figure 16-66**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 16-66. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution Be sure to set the transmit data to SDRmn register before the serial clock is input from the master.

Remarks 1. <1> to <8> in the above figure correspond to <1> to <8> in **Figure 16-65**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.5.7 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 16-3. 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	f_{CLK} = 40 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	32 MHz	40 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16 MHz	20 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8 MHz	10 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4 MHz	5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2 MHz	2.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1 MHz	1.25 MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500 kHz	625 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250 kHz	312.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125 kHz	156.25 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz	78.125 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz	39.0625 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz	19.53125 kHz
1	0	0	0	0	X	X	X	X	f_{CLK}	32 MHz	40 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16 MHz	20 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8 MHz	10 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4 MHz	5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2 MHz	2.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1 MHz	1.25 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500 kHz	625 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250 kHz	312.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125 kHz	156.25 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5 kHz	78.125 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25 kHz	39.0625 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63 kHz	19.53125 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)

16.5.8 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 16-67**.

Figure 16-67. Processing Procedure in Case of Overrun Error

Software Manipulation		Hardware Status	Remarks
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)

16.6 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

- RL78/F25 Products and RL78/F22 48-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C	Simplified I^2S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function)		IIC01	–
1	0	CSI10 (supporting SPI function) Note 1	UART1	IIC10	Supported
	1	CSI11 (supporting SPI function)		IIC11	–

- RL78/F22 32-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C	Simplified I^2S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function)		IIC01	–
1	0	CSI10 (supporting SPI function) Note 1	UART1	IIC10	Supported
	1	–		–	–

- RL78/F22 24-pin Products.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C	Simplified I^2S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function) Note 2		IIC01	–
1	0	–	UART1	–	Supported
	1	–		–	–

Notes 1. 48-pin and 32-pin products do not have SSI10 pin.

2. 24-pin products do not have SSI01 pin.

SPI function performs the following six types of communication operations.

- Master transmission (See **16.6.1 Master Transmission**.)
- Master reception (See **16.6.2 Master Reception**.)
- Master transmission/reception (See **16.6.3 Master Transmission/Reception**.)
- Slave transmission (See **16.6.4 Slave Transmission**.)
- Slave reception (See **16.6.5 Slave Reception**.)
- Slave transmission/reception (See **16.6.6 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. 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 16-68. Example of Slave Select Input Function Configuration

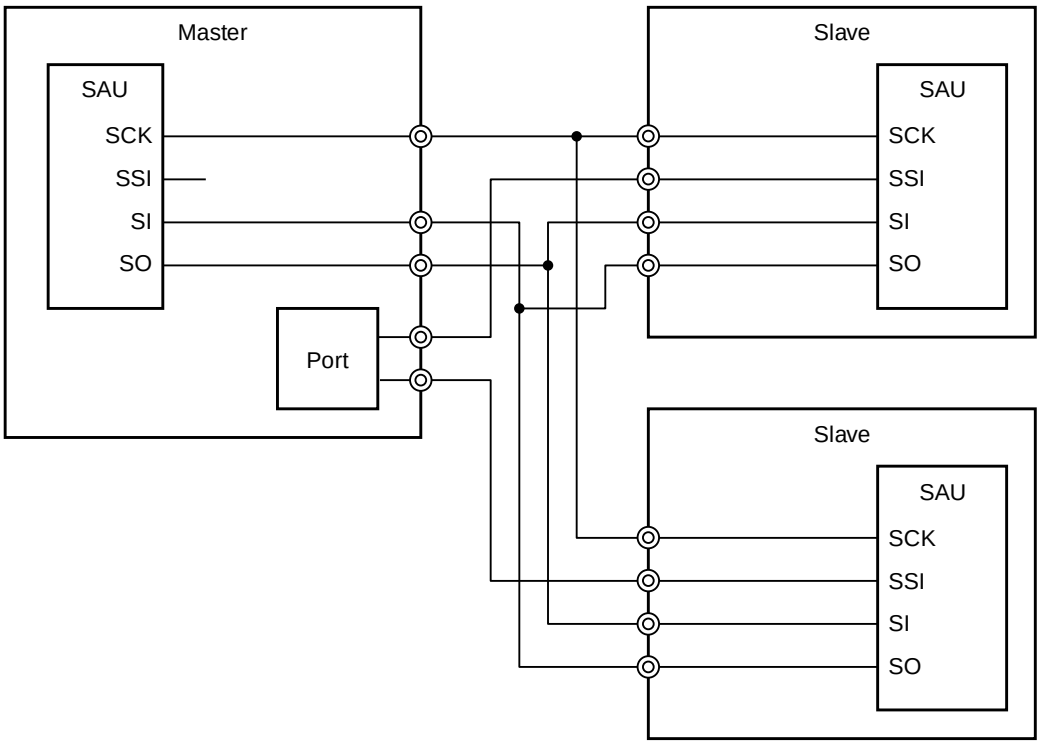
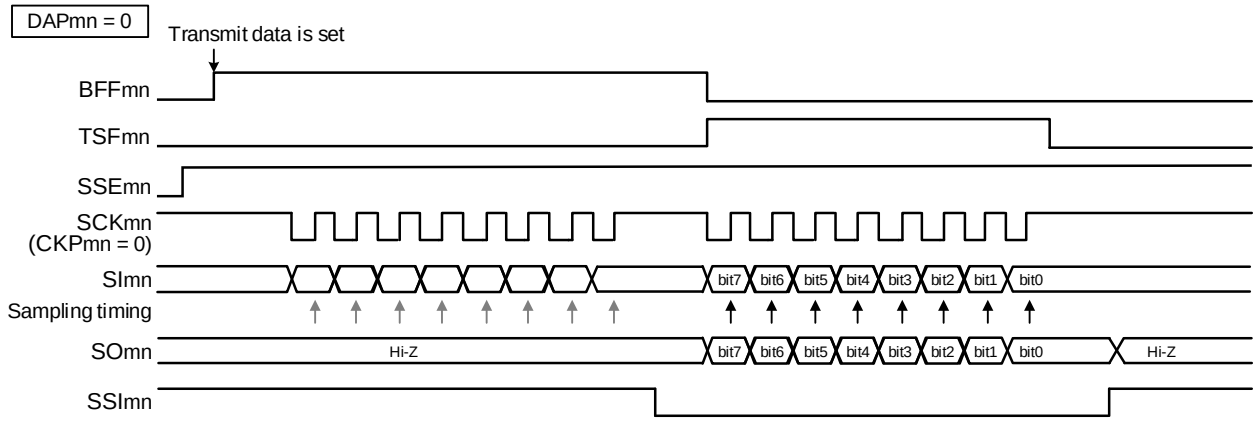
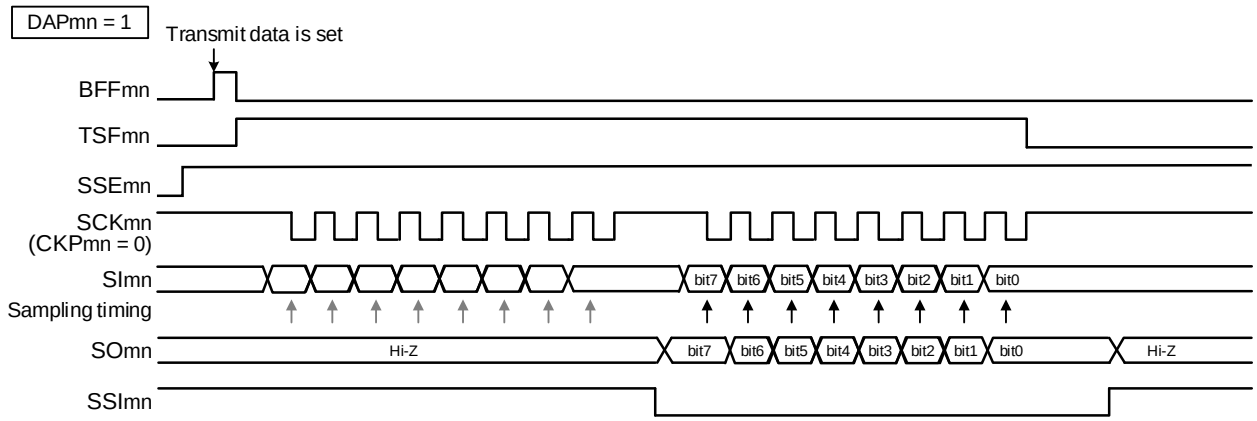


Figure 16-69. Slave Select Input Function Timing Diagram



While SSImn is at high level, transmission is not performed even if the SCKmn (serial clock) arrives, and neither is the receive data sampled in synchronization with the rising edge.
When 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.



While SSImn is at high level, transmission is not performed even if the SCKmn (serial clock) arrives, and neither is the receive data sampled in synchronization with the falling edge.
When SSImn goes to low level, data is output (shifted) in synchronization with the rising edge of the serial clock and a reception operation is performed in synchronization with the falling edge.

Caution This timing diagram is an example of slave transmission/reception. For master transmission / reception, output the slave select signal by port manipulation.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.6.1 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	SCK00, SO00	SCK01, SO01	SCK10, SO10	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 • DAPmn = 0: Data output starts from the start of the serial clock operation. • 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 • CKPmn = 0: Forward • 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.

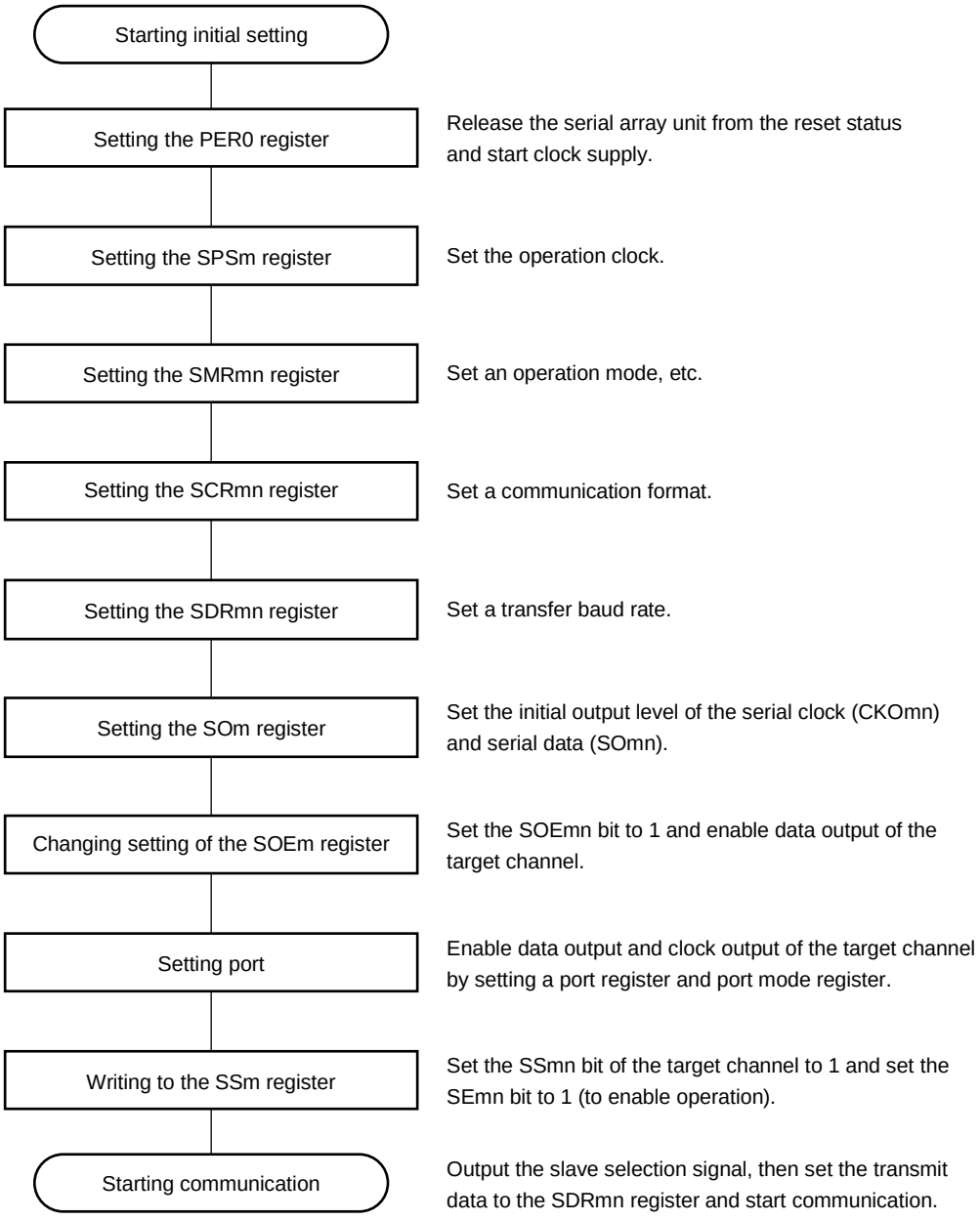
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(1) Register Setting

Master transmission is the same as 3-wire serial I/O function. See **16.5.1 Master Transmission** of 3-wire serial I/O function.

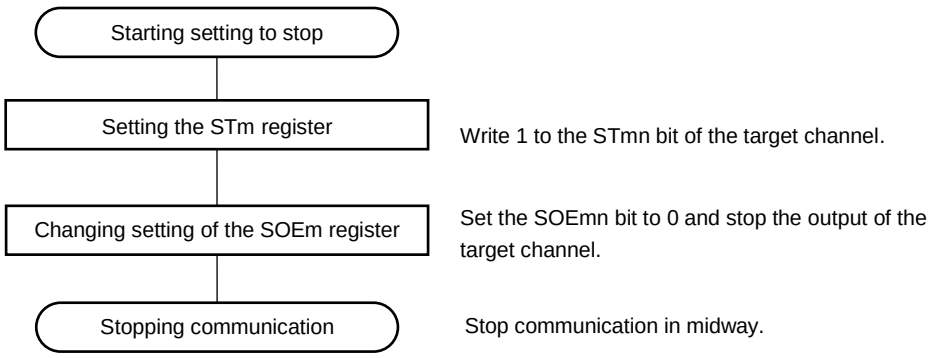
(2) Operation Procedure

Figure 16-70. Initial Setting Procedure for Master Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

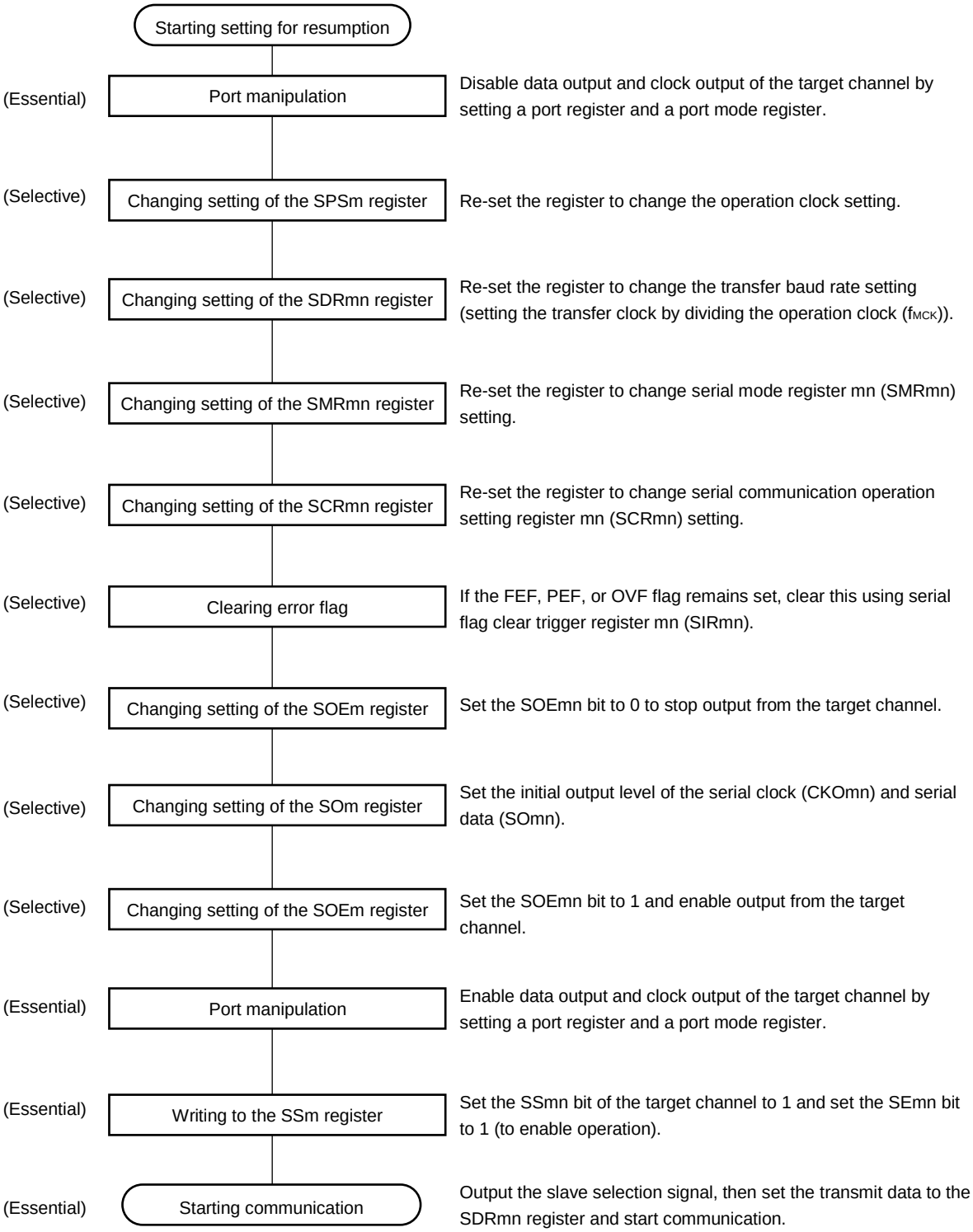
Figure 16-71. Procedure for Stopping Master Transmission



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 16-72. Procedure for Resuming Master Transmission**). Also, control the slave selection signal by port manipulation.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

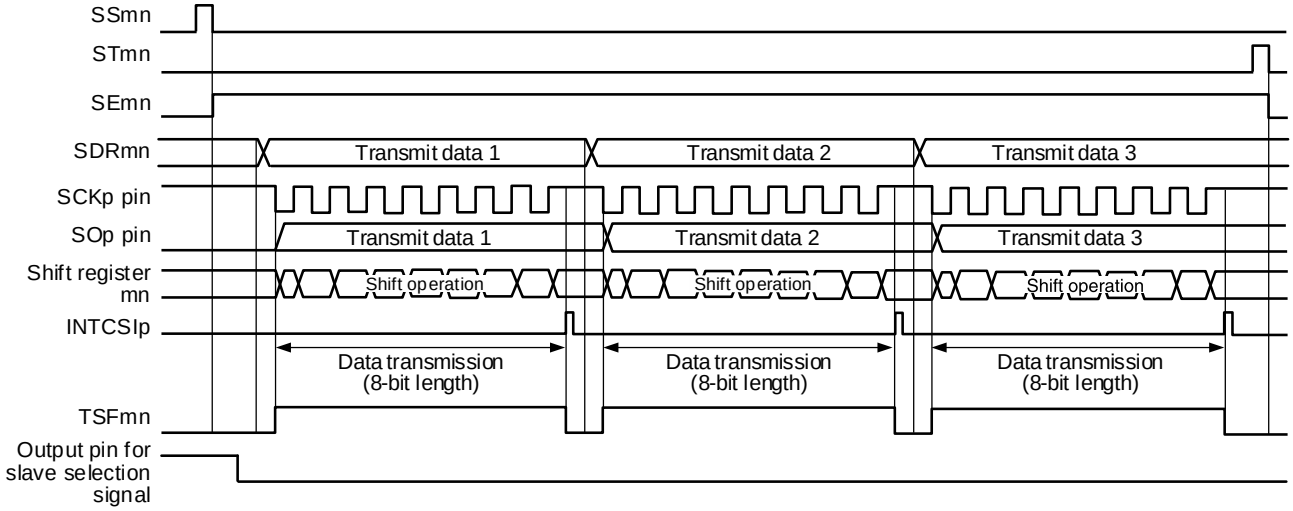
Figure 16-72. Procedure for Resuming Master Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1),

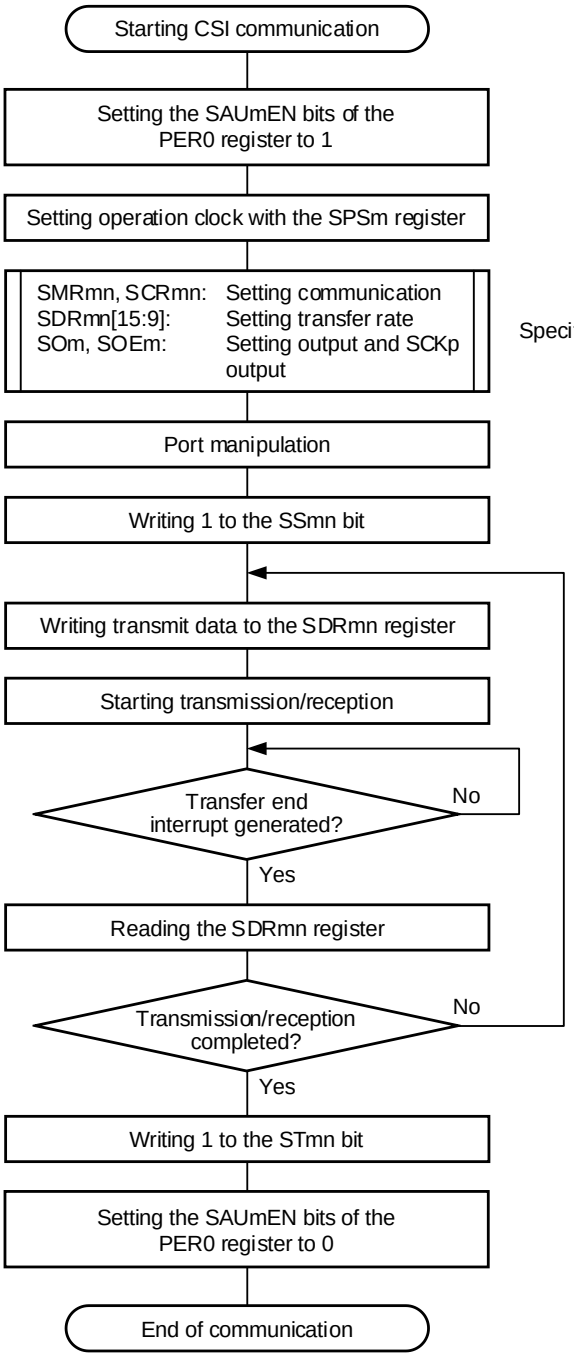
(3) Processing Flow (in Single-Transmission Mode)

Figure 16-73. 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)

Figure 16-74. Flowchart of Master Transmission (in Single-Transmission Mode)

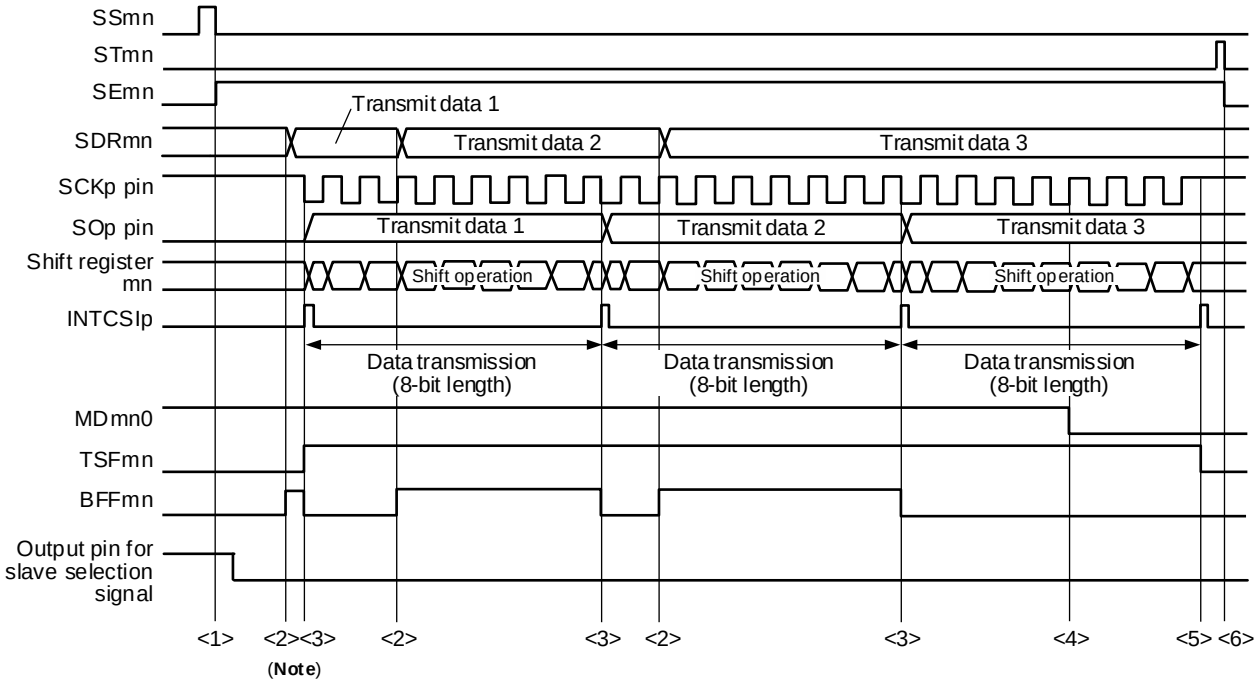


Caution Control the output of the slave selection signal by port manipulation.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

(4) Processing Flow (in Continuous Transmission Mode)

Figure 16-75. Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



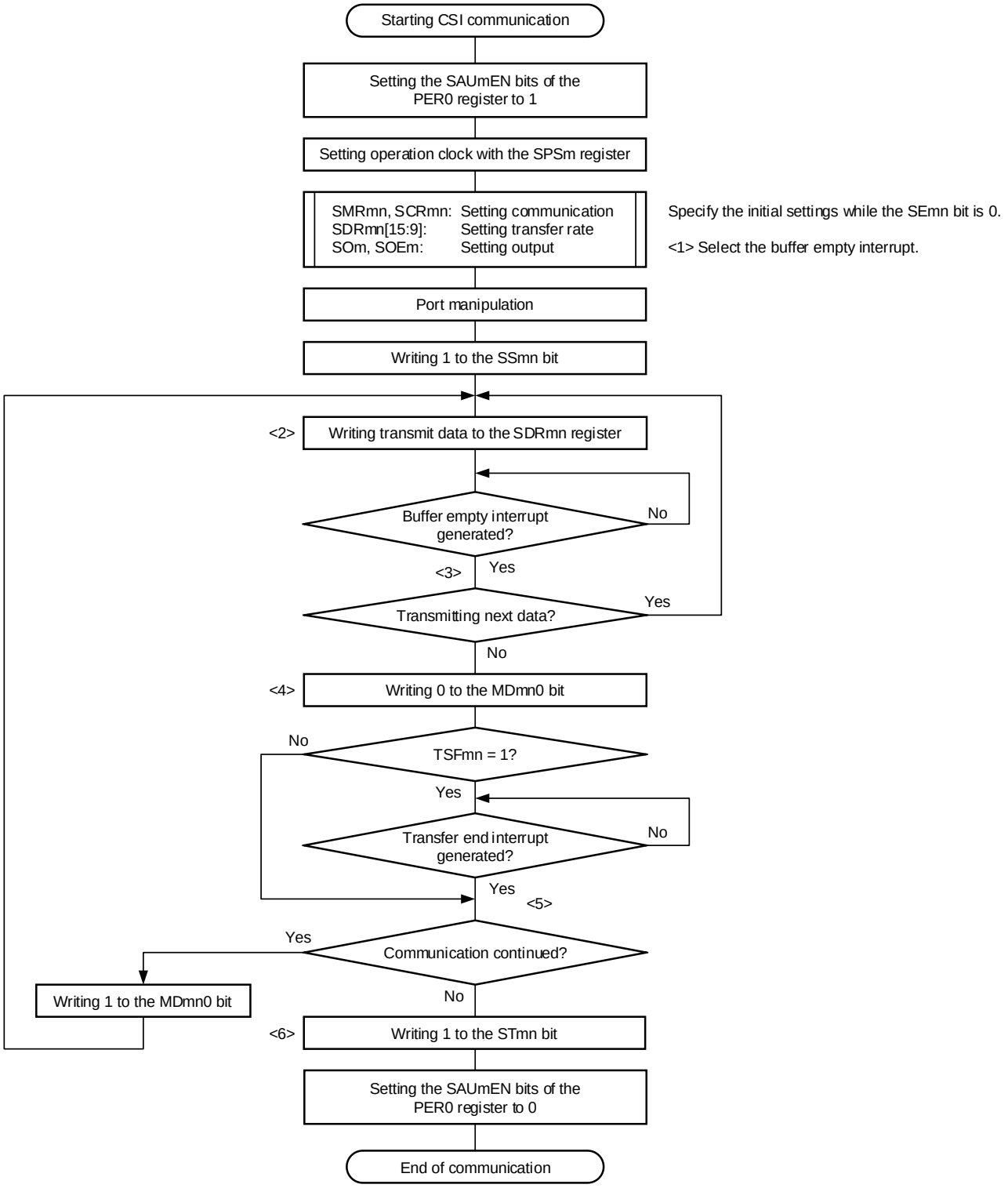
Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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.

Remarks 1. <1> to <6> in the above figure correspond to <1> to <6> in **Figure 16-76**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 16-76. Flowchart of Master Transmission (in Continuous Transmission Mode)



Caution Control the output of the slave selection signal by port manipulation.

Remarks 1. <1> to <6> in the above figure correspond to <1> to <6> in **Figure 16-75**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.6.2 Master Reception

Master reception is an operation wherein this MCU outputs a transfer clock and receives data from the 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	SCK00, SI00	SCK01, SI01	SCK10, SI10	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 • DAPmn = 0: Data input starts from the start of the serial clock operation. • 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 • CKPmn = 0: Forward • 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)

(1) Register Setting

Master reception is the same as 3-wire serial I/O function. See **16.5.2 Master Reception** of 3-wire serial I/O function.

(2) Operation Procedure

Figure 16-77. Initial Setting Procedure for Master Reception

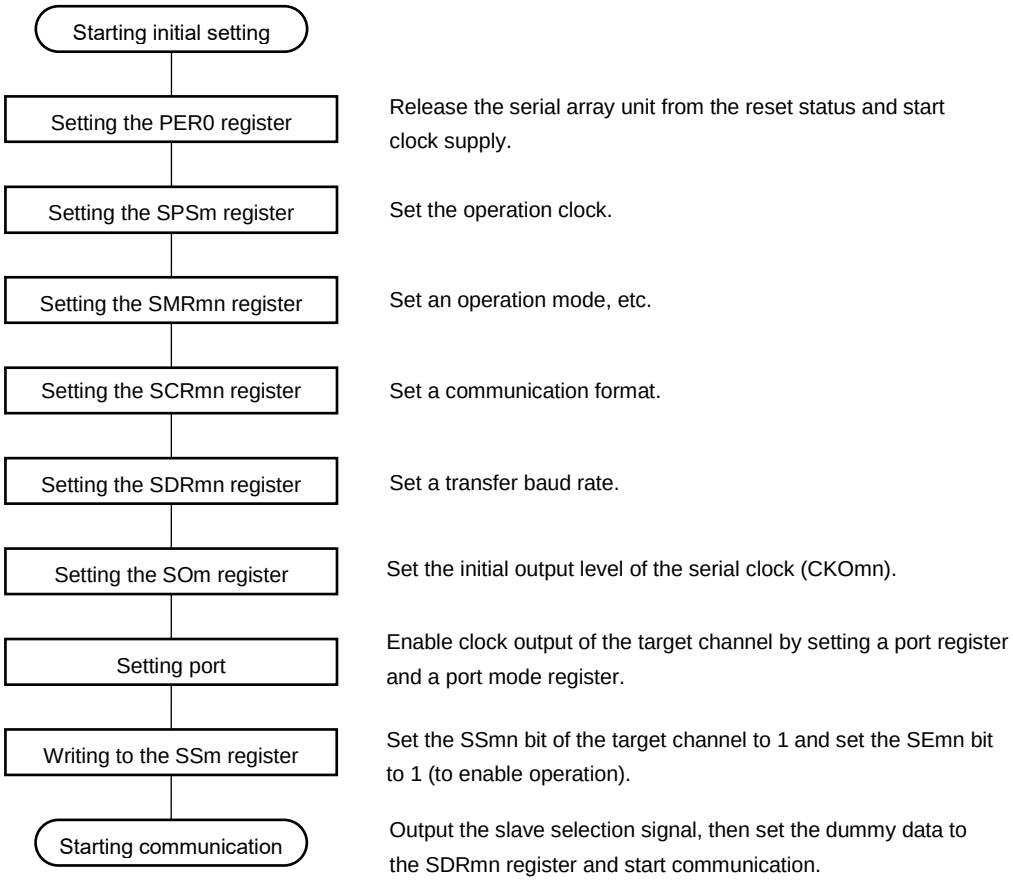
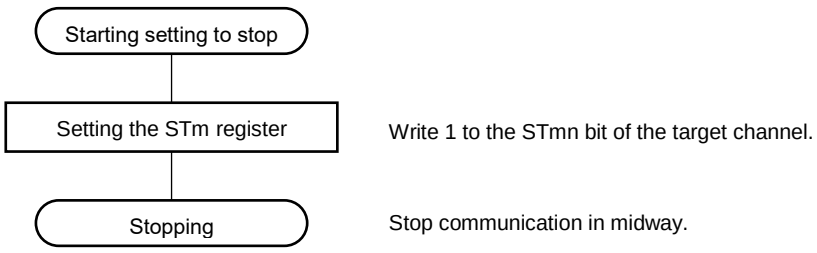


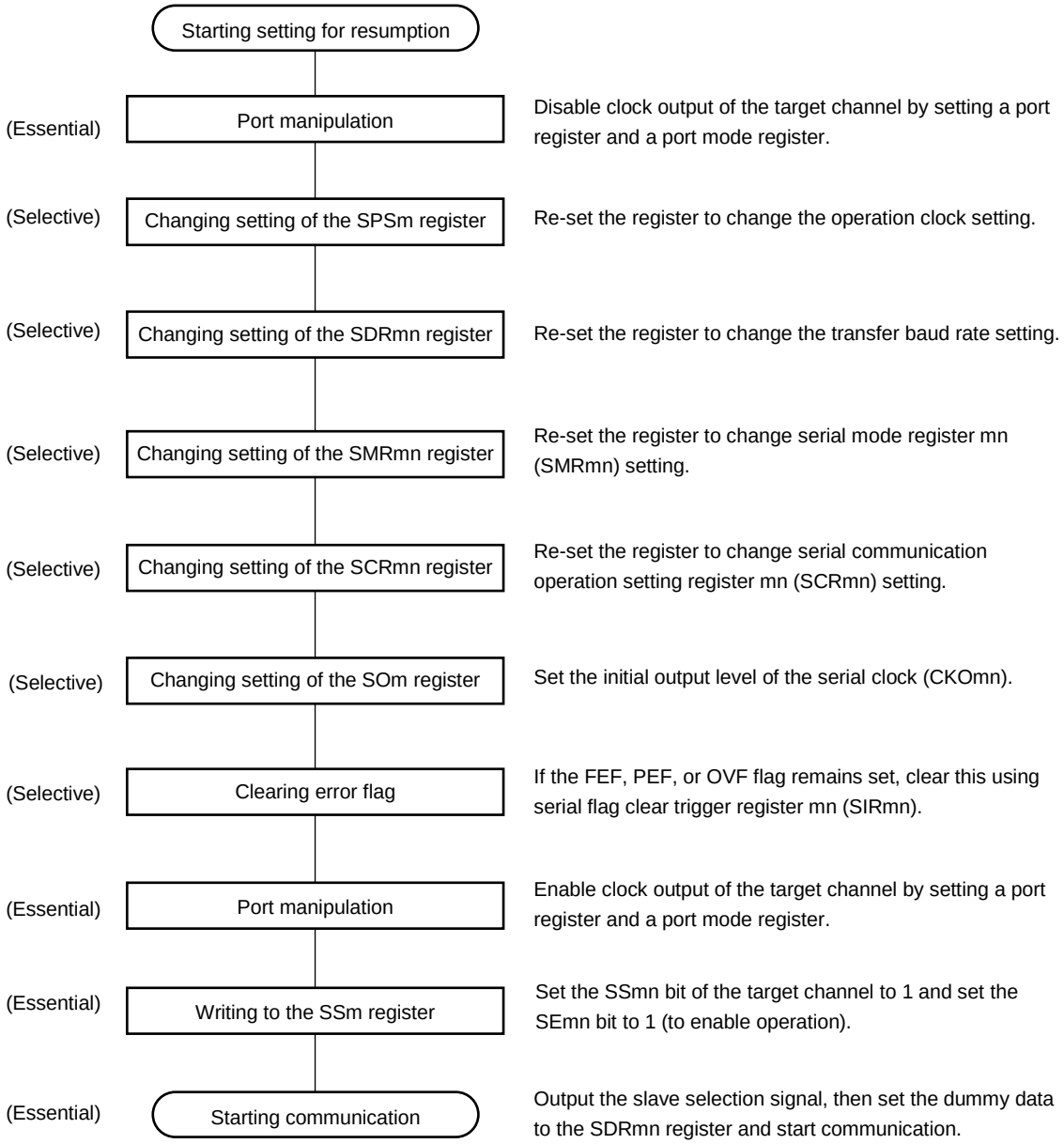
Figure 16-78. Procedure for Stopping Master Reception



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 16-79. Procedure for Resuming Master Reception**). Also, control the slave selection signal by port manipulation.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

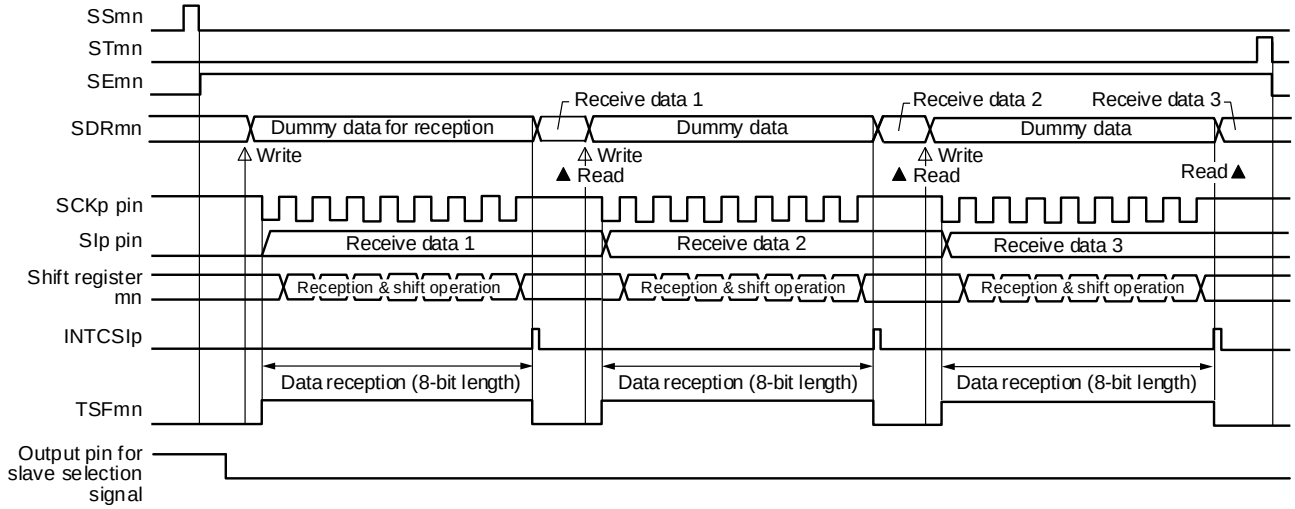
Figure 16-79. Procedure for Resuming Master Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

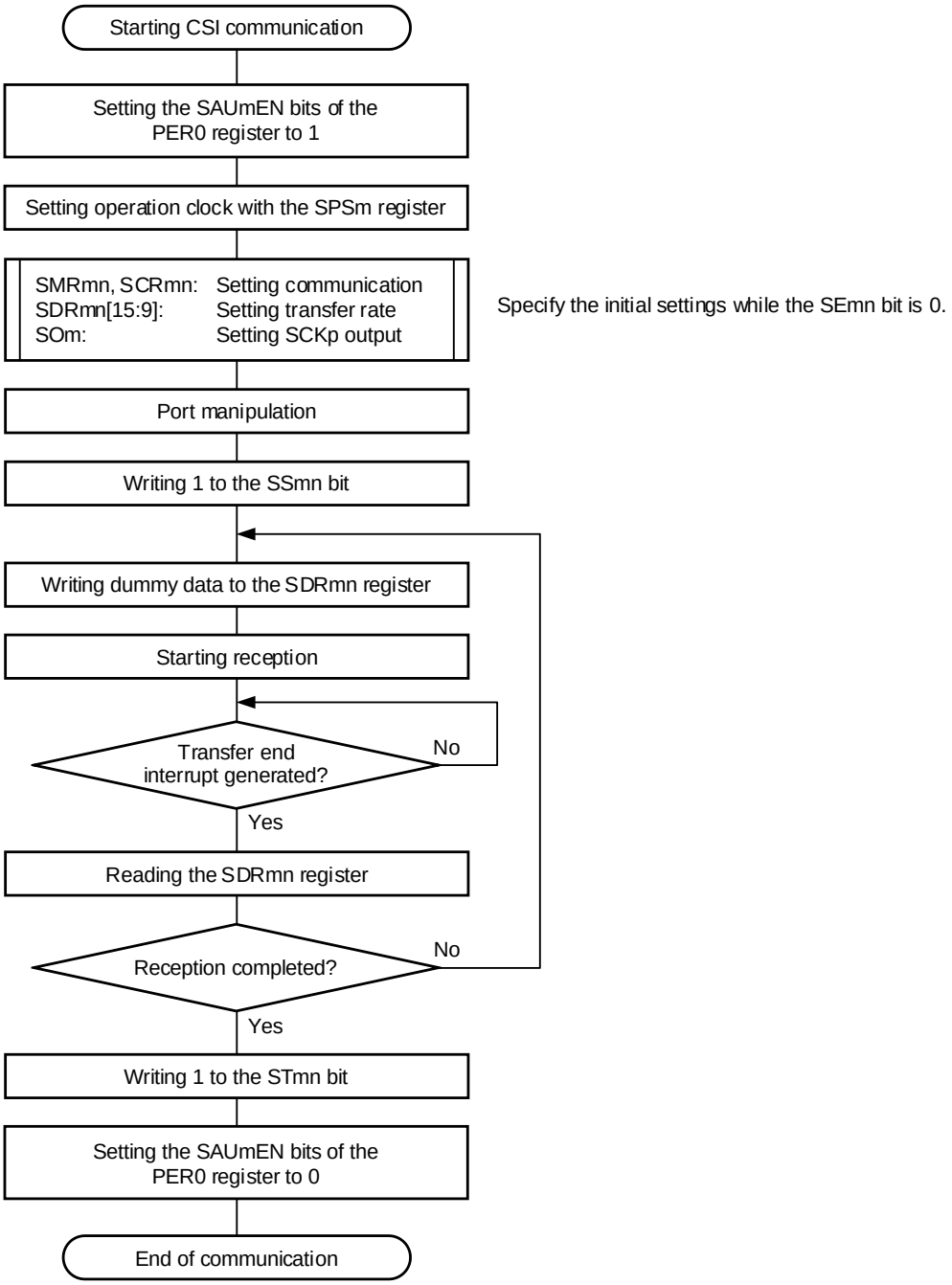
(3) Processing Flow (in Single-Reception Mode)

Figure 16-80. 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)

Figure 16-81. Flowchart of Master Reception (in Single-Reception Mode)

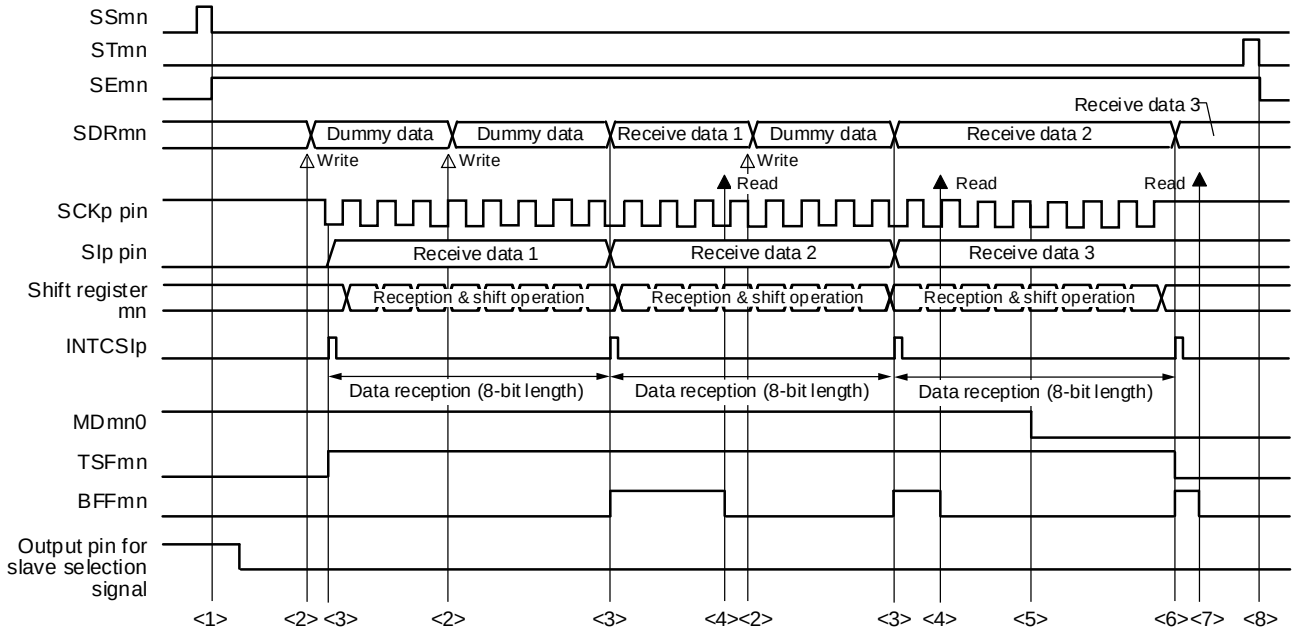


Caution Control the output of the slave selection signal by port manipulation.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

(4) Processing Flow (in Continuous Reception Mode)

Figure 16-82. Timing Chart of Master Reception (in Continuous Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

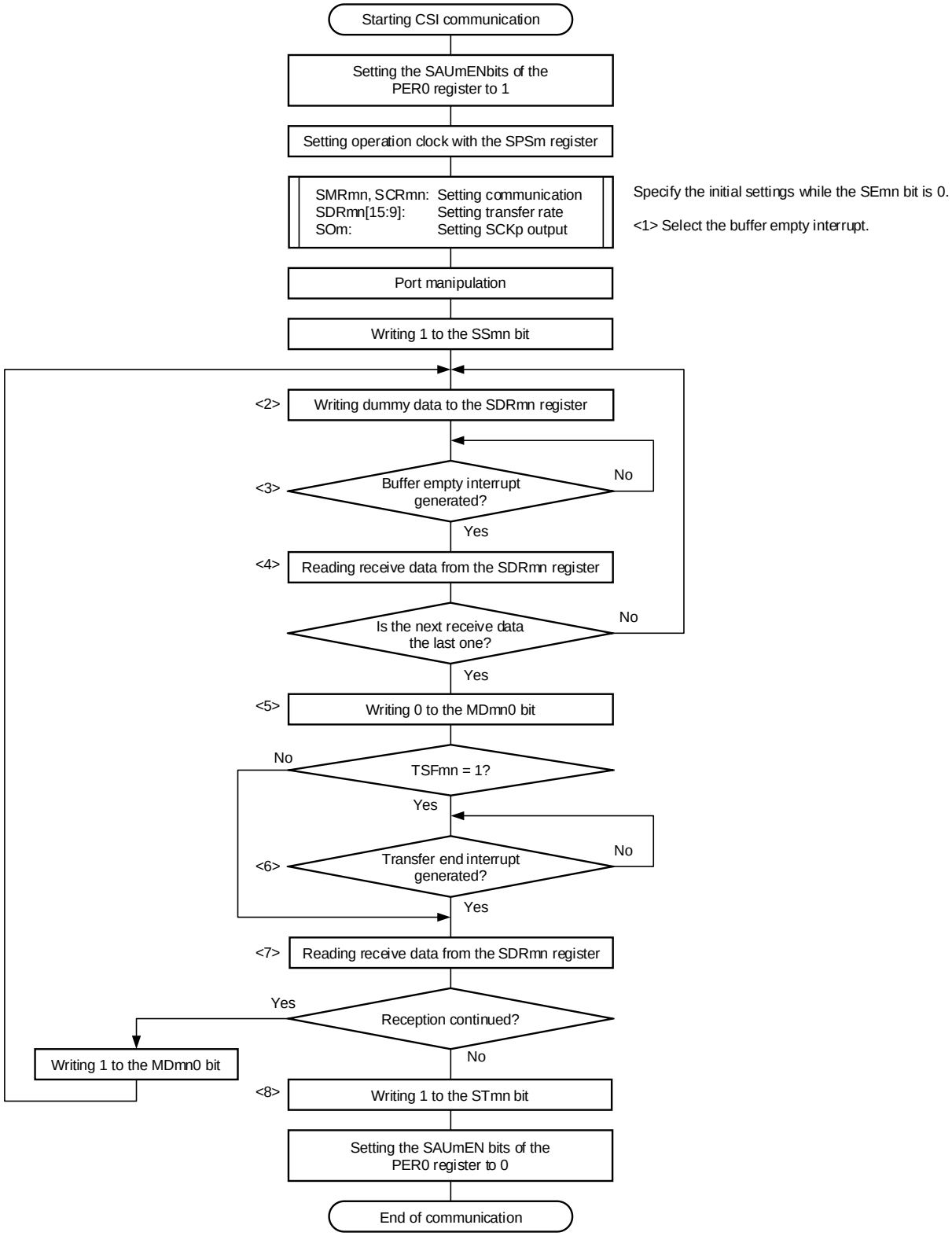


Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the above figure correspond to <1> to <8> in **Figure 16-83**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 16-83. Flowchart of Master Reception (in Continuous Reception Mode)



Caution Control the output of the slave selection signal by port manipulation.

Remarks 1. <1> to <8> in the above figure correspond to <1> to <8> in **Figure 16-82**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

16.6.3 Master Transmission/Reception

Master transmission/reception is an operation wherein this MCU outputs a transfer clock and transmits/receives data to/from the 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	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	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 • DAPmn = 0: Data I/O starts at the start of the serial clock operation. • 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 • CKPmn = 0: Forward • 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.

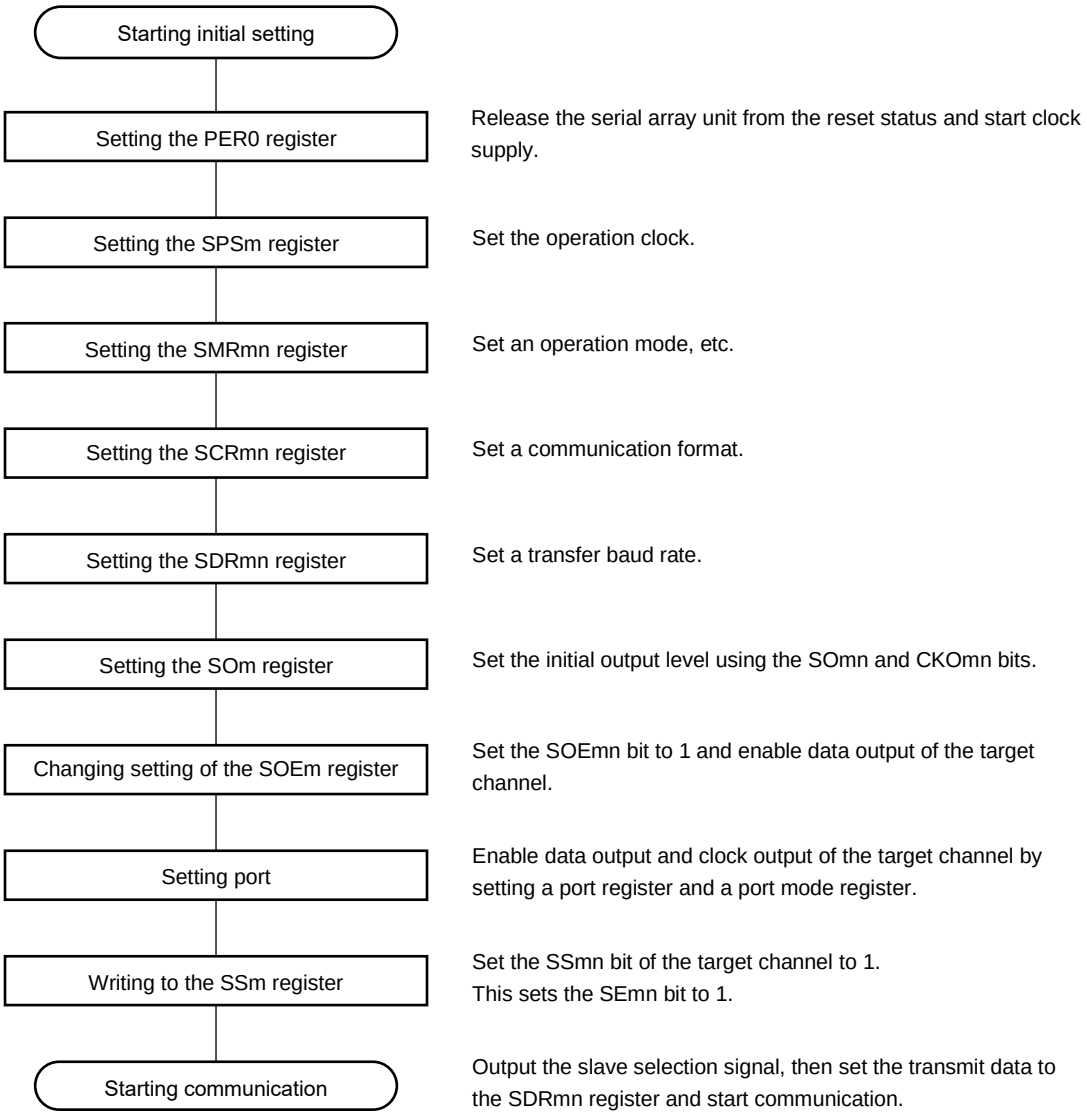
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(1) Register Setting

Master transmission/reception is the same as 3-wire serial I/O function. See **16.5.3 Master Transmission/Reception** of 3-wire serial I/O function.

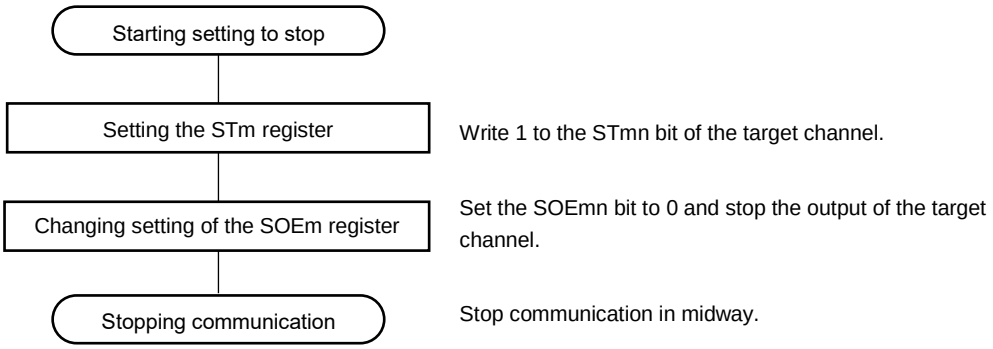
(2) Operation Procedure

Figure 16-84. Initial Setting Procedure for Master Transmission/Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

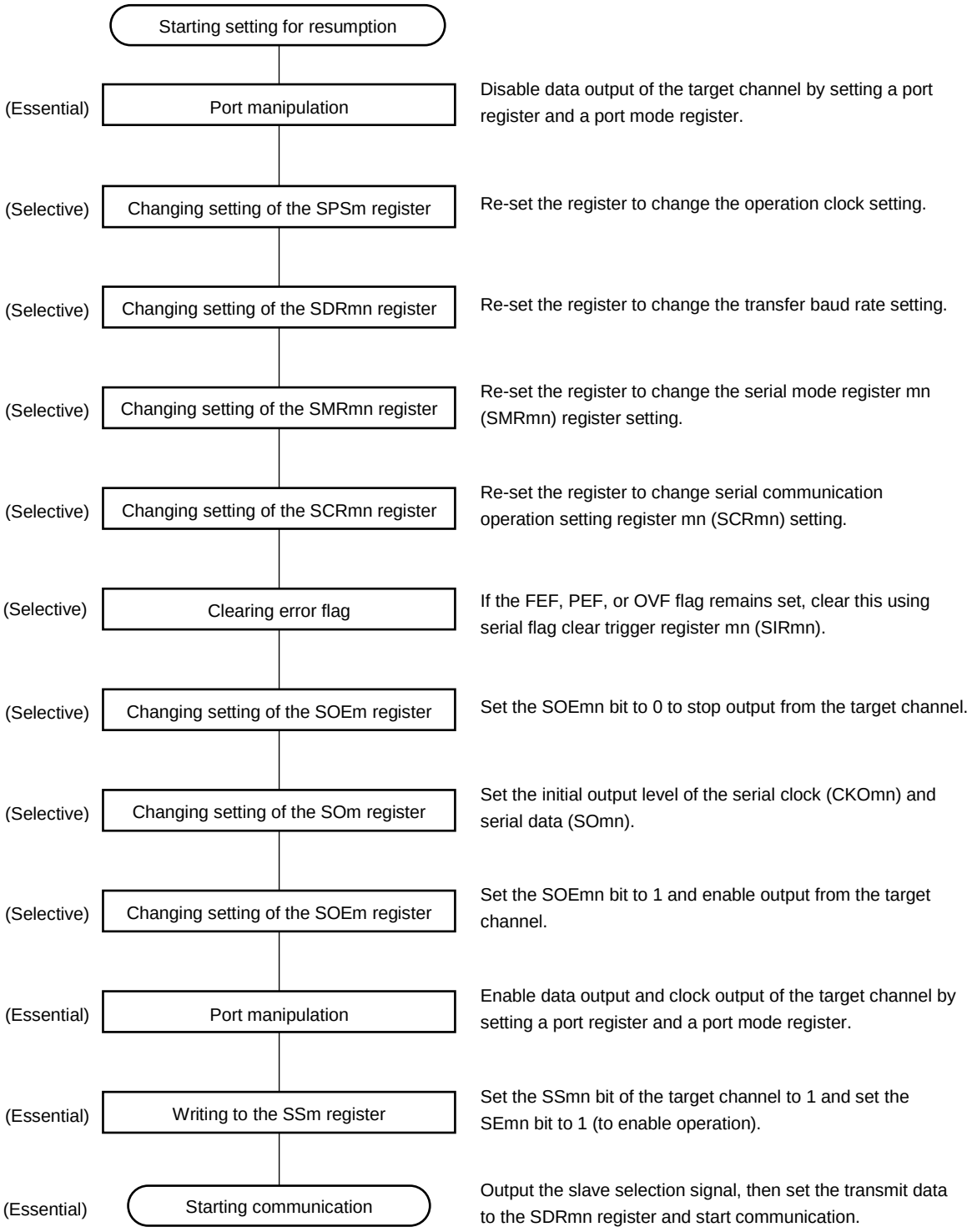
Figure 16-85. Procedure for Stopping Master Transmission/Reception



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set SOm register (see **Figure 16-86. Procedure for Resuming Master Transmission/Reception**).
Also, control the slave selection signal by port manipulation.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

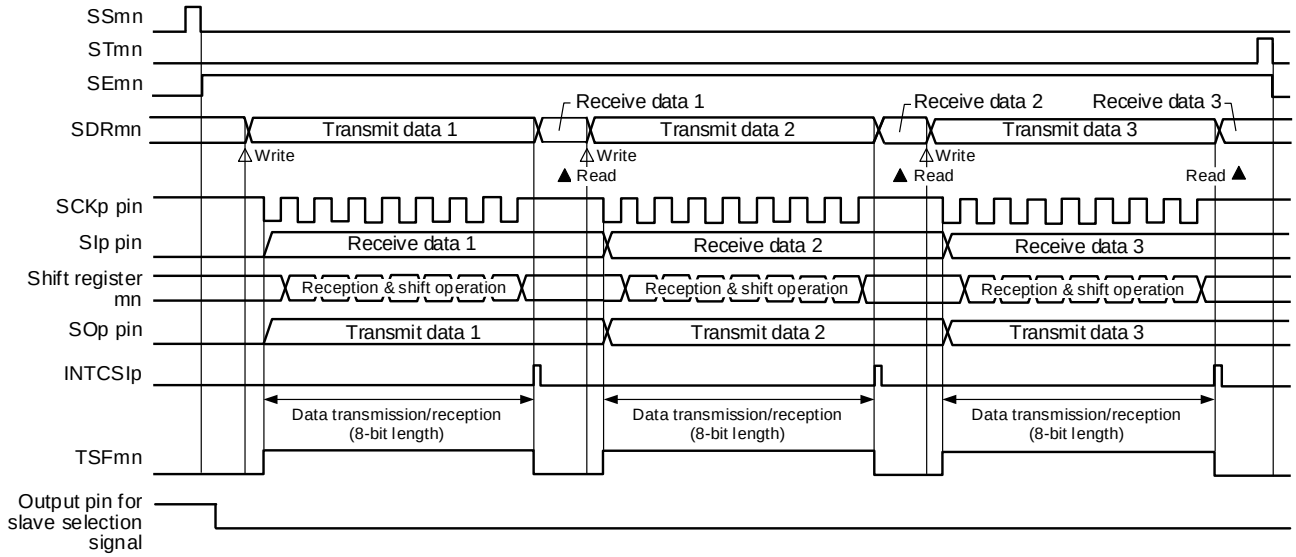
Figure 16-86. Procedure for Resuming Master Transmission/Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

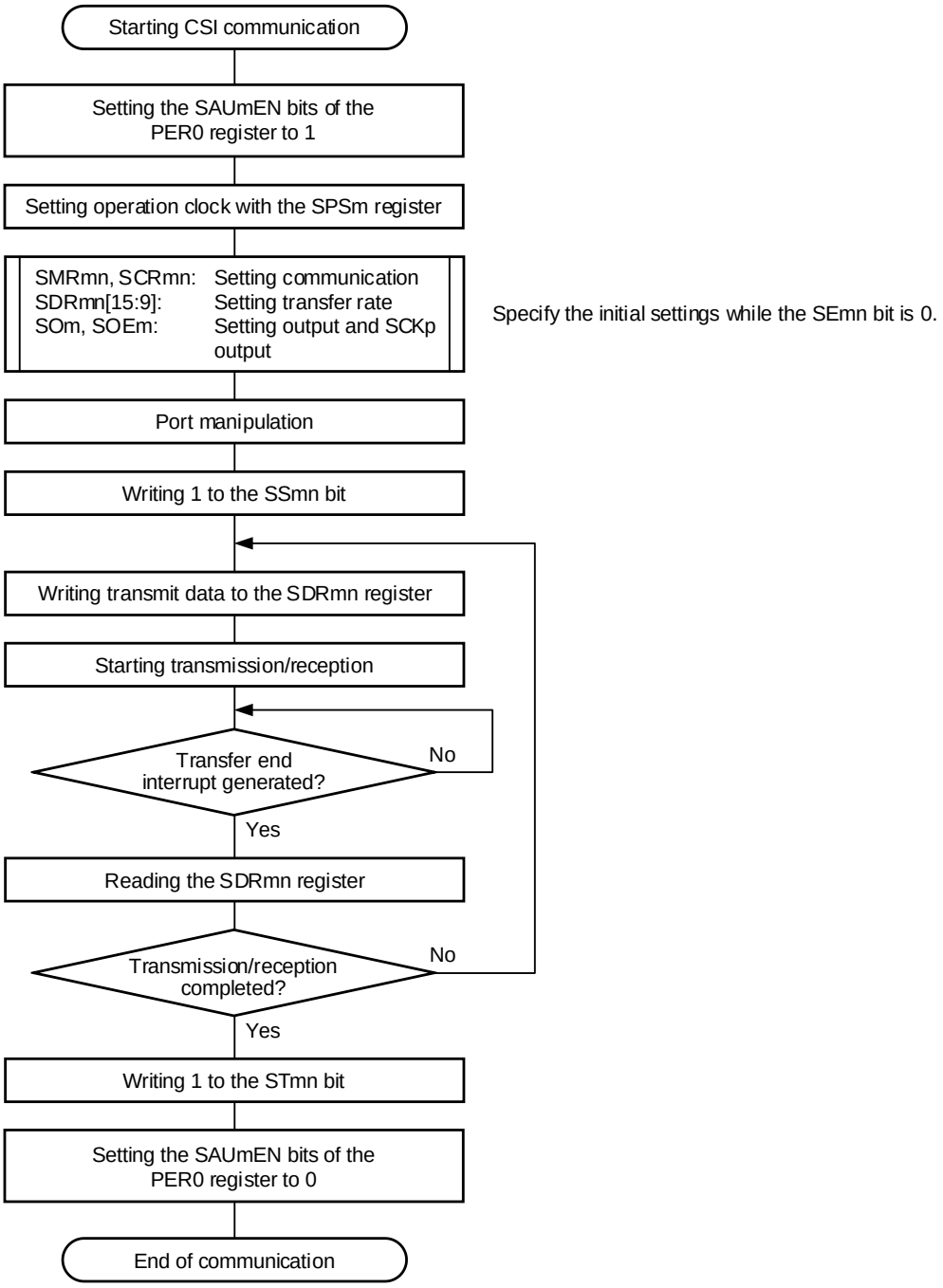
(3) Processing Flow (in Single-Transmission/Reception Mode)

Figure 16-87. 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)

Figure 16-88. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)

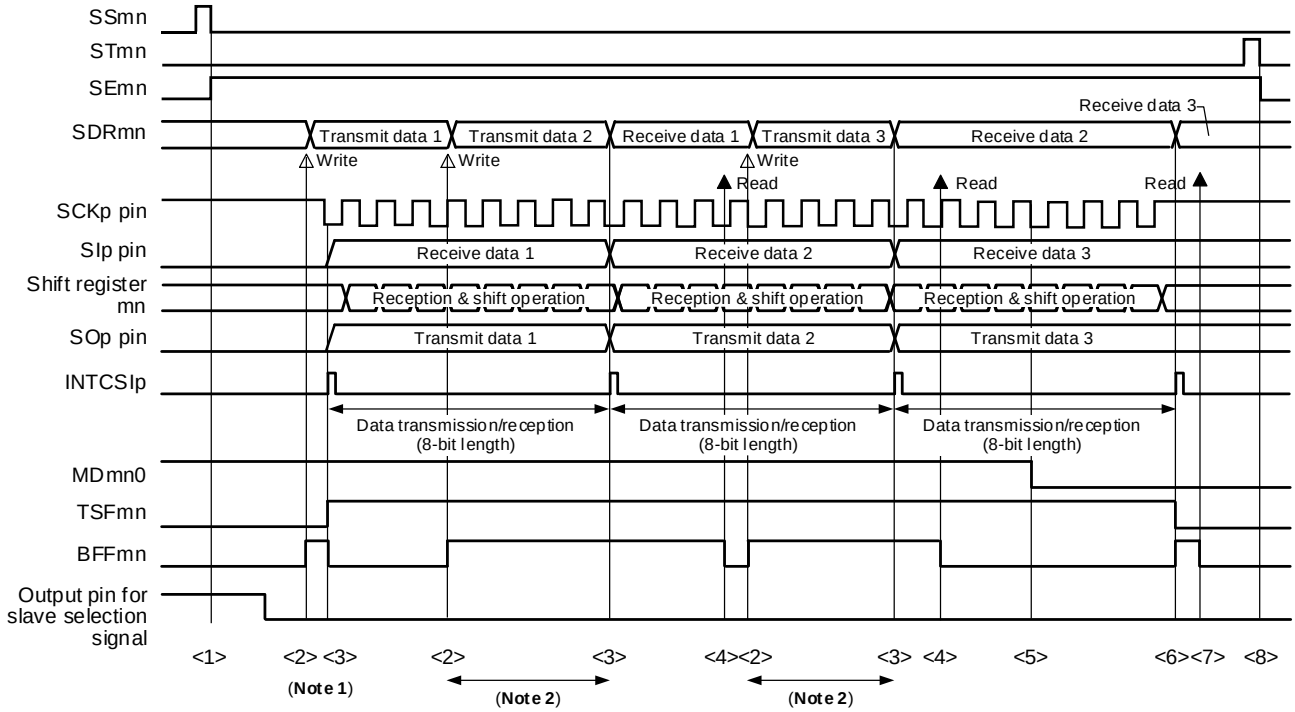


Caution Control the output of the slave selection signal by port manipulation.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

(4) Processing Flow (in Continuous Transmission/Reception Mode)

Figure 16-89. 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.

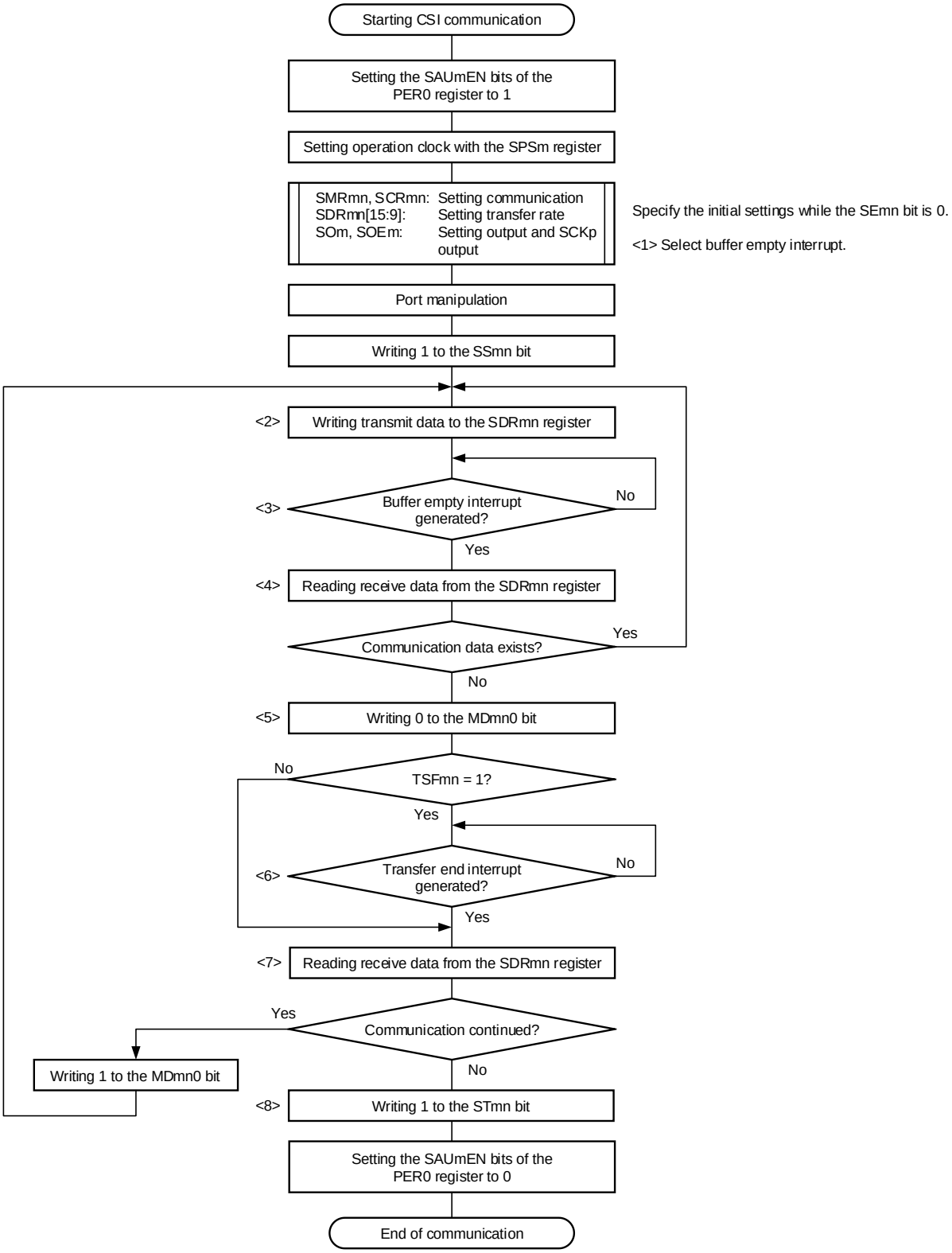
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 above figure correspond to <1> to <8> in **Figure 16-90**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 16-90. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution Control the output of the slave selection signal by port manipulation.

Remarks 1. <1> to <8> in the above figure correspond to <1> to <8> in **Figure 16-89**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

16.6.4 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	SCK00, SO00, SSI00	SCK01, SO01, SSI01	SCK10, SO10, SSI10	SCK11, SO11, 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 • DAPmn = 0: Data output starts from the start of the serial clock operation. • 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 • CKPmn = 0: Forward • 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 SCK00, SCK01, SCK10, and 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)

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 1 (clock input from the SCKp pin).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 00B (CSI mode).
The MDmn0 bit selects interrupt source of channel n (0: transfer end, 1: buffer empty).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 10B (transmission only).
DAPmn and CKPmn bits select data and clock phase.
PTCmn1 and PTCmn0 bits set to 00B.
The DIRmn bit selects data transfer sequence (0: MSB first, 1: LSB first).
SLCmn1 and SLCmn0 bits are set to 00B.
DLSmn3 to DLSmn0 bits select data length (6H (7 bits) to FH (16 bits)).

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set bits 15 to 9 of the SDRmn register to 0000000B.
When operation is in progress (SEmn = 1): Set the transmit data in the SDRmn register.

Note Lower 8 bit in the SDRmn register: SDRpL register

(d) Serial output register m (SOm)

Set the SOmn bit of the target channel.

(e) Serial output enable register m (SOEm)

Set the SOEmn bit of the target channel to 1 (serial output enable).

(f) Serial slave select enable register m (SSEm)

Set the SSEmn bit of the target channel to 1 (enables SSIp pin input).

(g) Serial channel start register m (SSm)

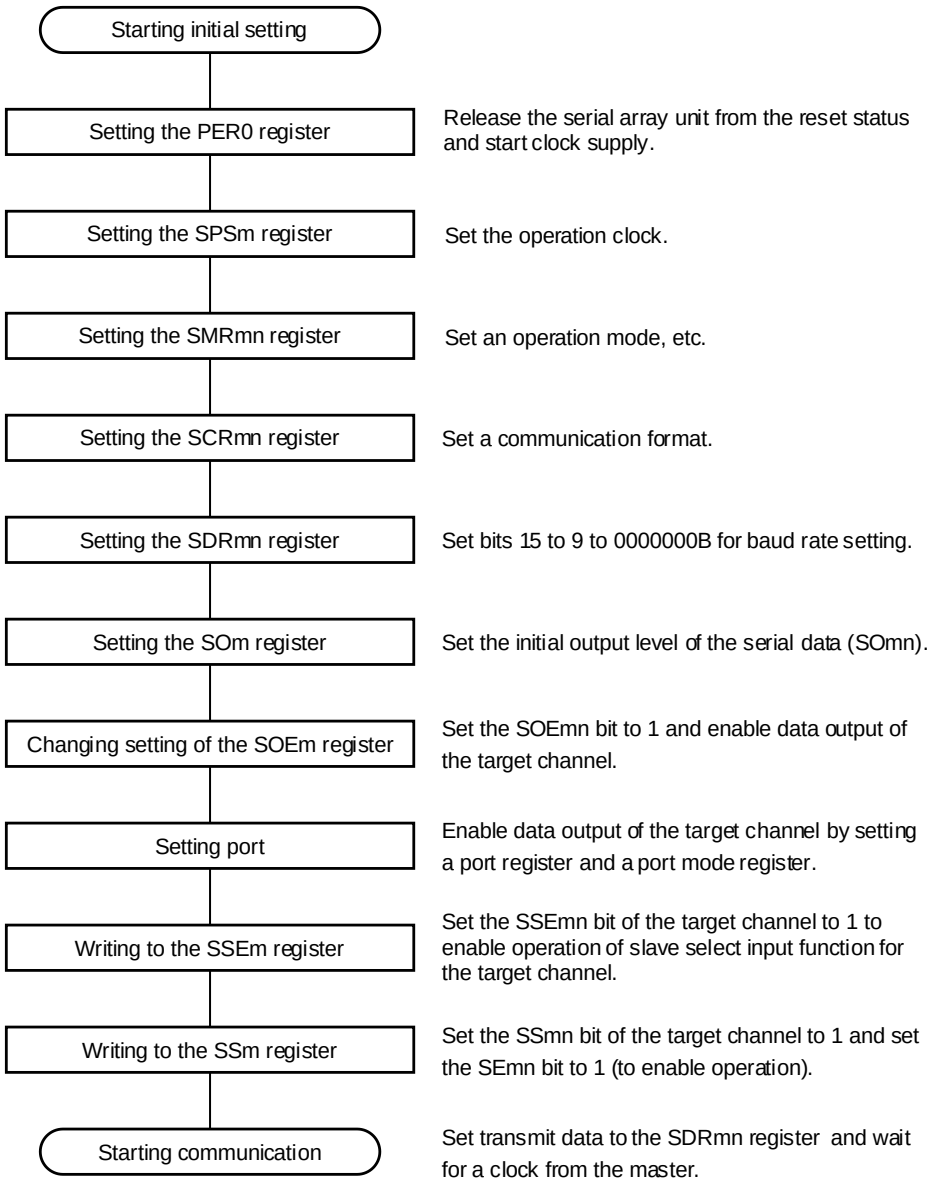
Set the SSmn bit of the target channel to 1 (operation start).

Caution Be sure to set the transmit data to SDRmn register before the serial clock is input from the master.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

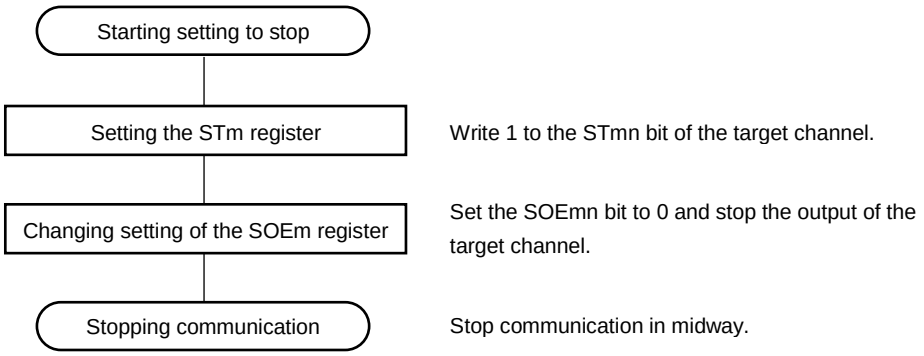
(2) Operation Procedure

Figure 16-91. Initial Setting Procedure for Slave Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

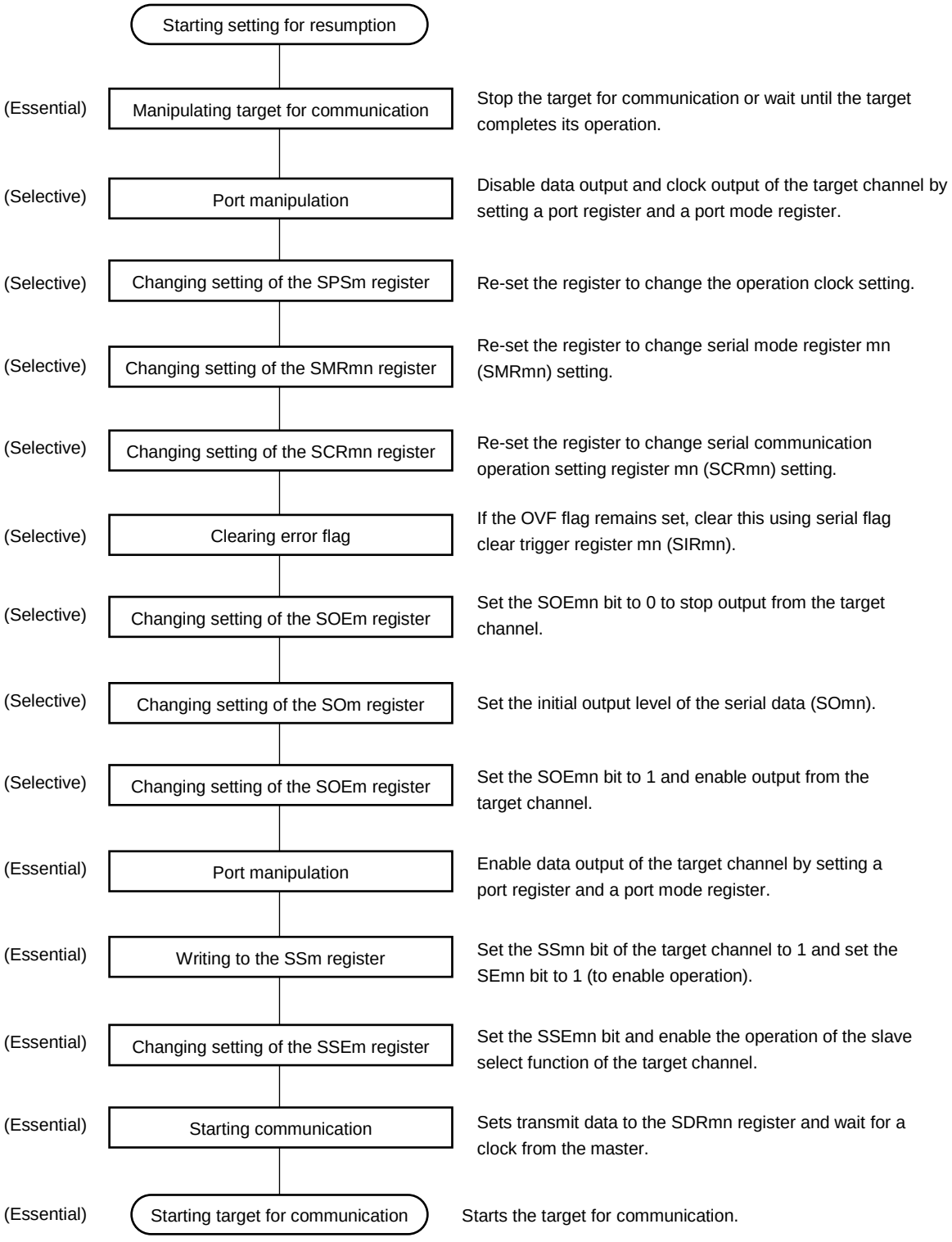
Figure 16-92. Procedure for Stopping Slave Transmission



Remarks 1. When the SSImn input signal is low, the pin level is retained even after communication is stopped. To resume the operation, re-set SOm register (see **Figure 16-93. Procedure for Resuming Slave Transmission**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

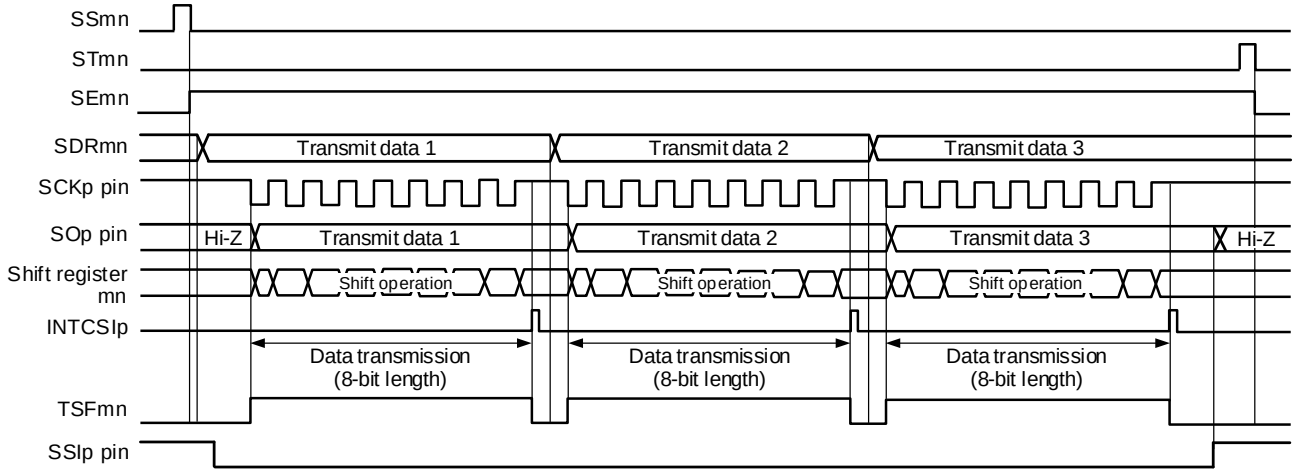
Figure 16-93. Procedure for Resuming Slave Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

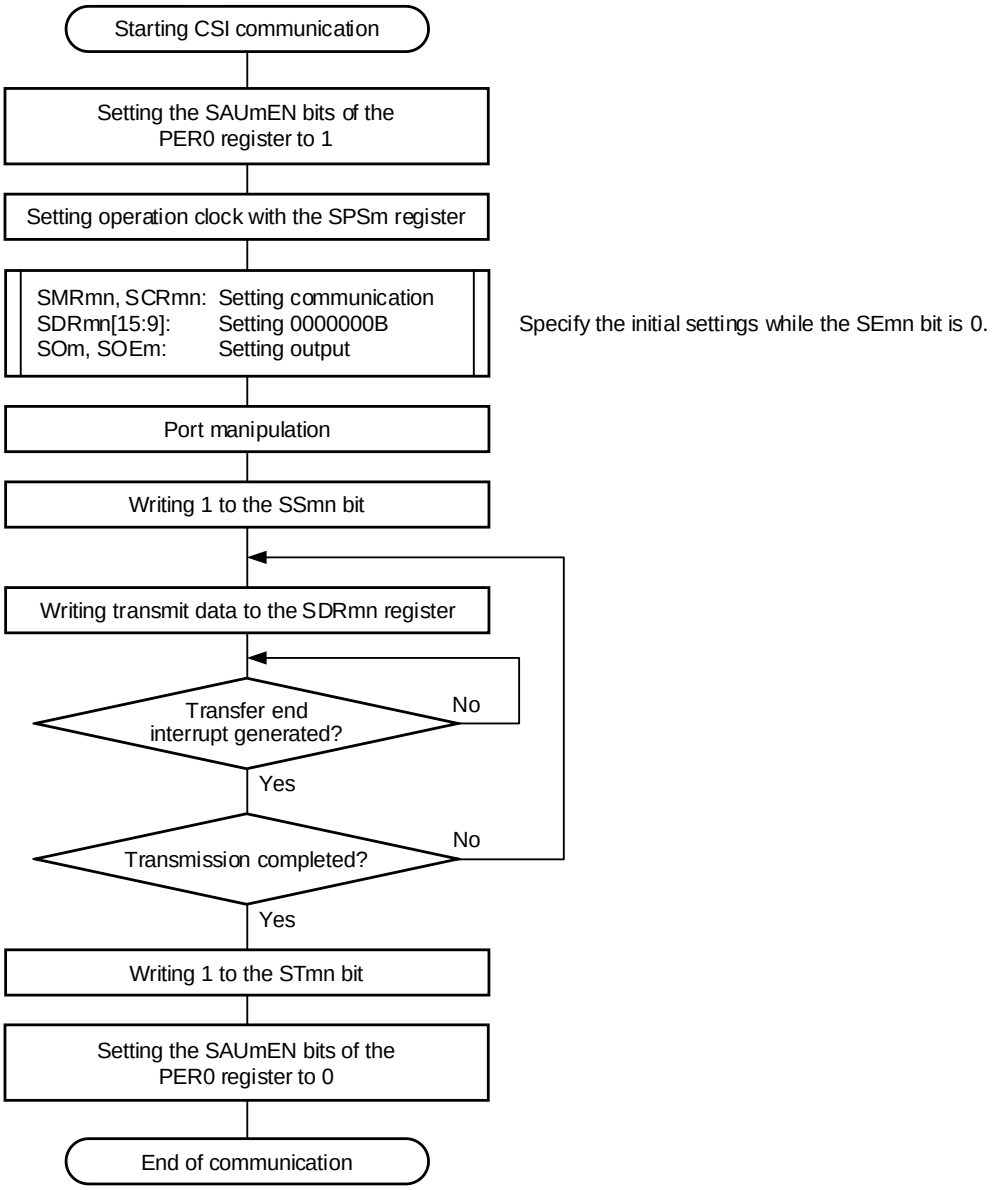
(3) Processing Flow (in Single-Transmission Mode)

Figure 16-94. 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)

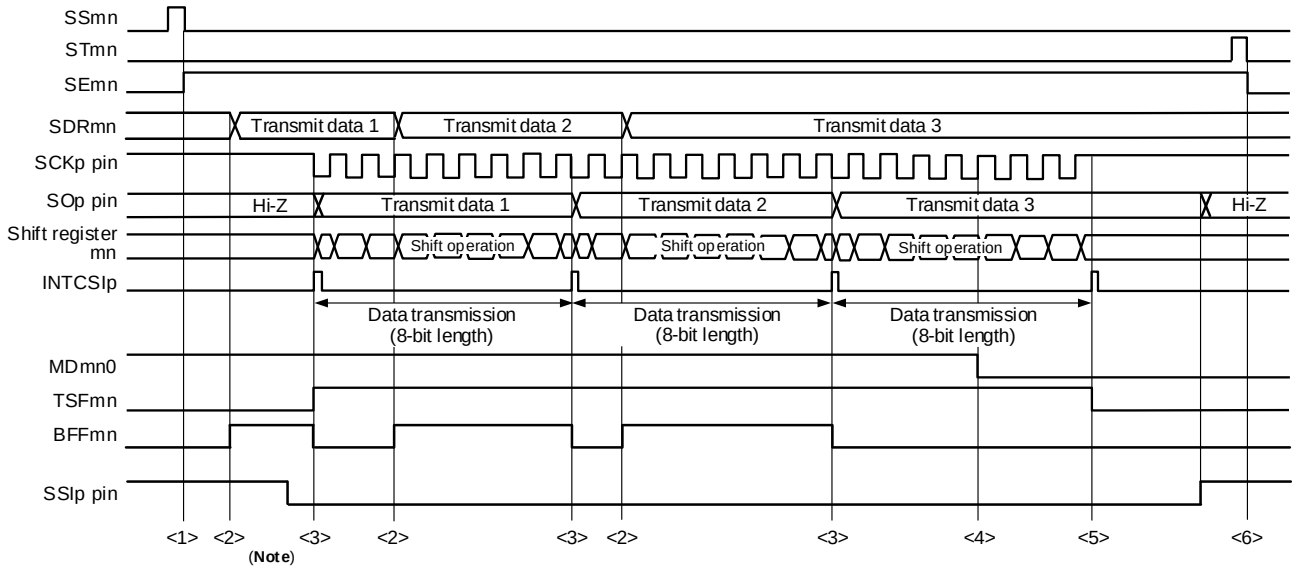
Figure 16-95. Flowchart of Slave Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(4) Processing Flow (in Continuous Transmission Mode)

Figure 16-96. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



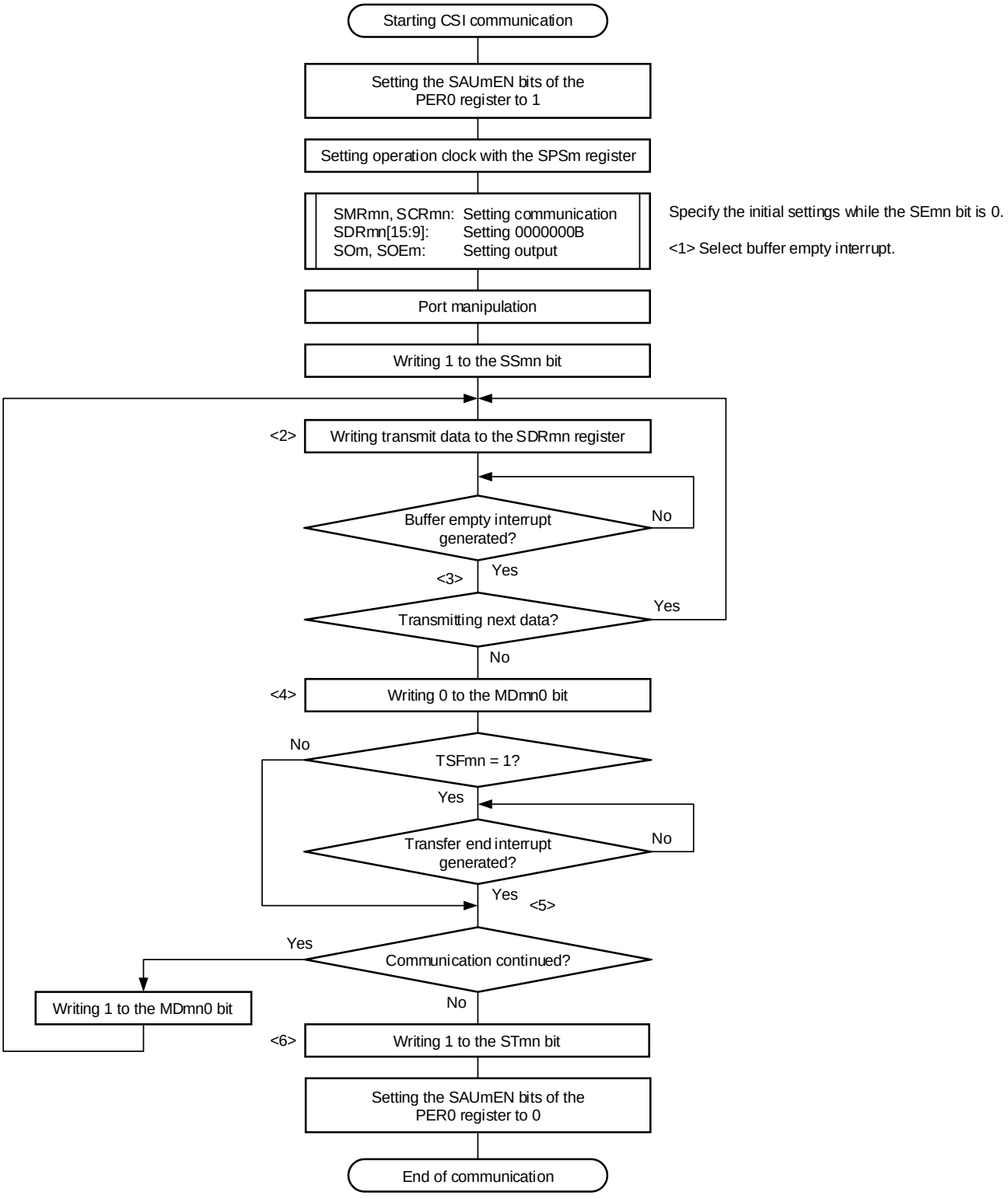
Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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.

Remarks 1. <1> to <6> in the above figure correspond to <1> to <6> in **Figure 16-97**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 16-97. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remarks 1. <1> to <6> in the above figure correspond to <1> to <6> in **Figure 16-96**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

16.6.5 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 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00, SSI00	SCK01, SI01, SSI01	SCK10, SI10, SSI10	SCK11, SI11, SSI11
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 • DAPmn = 0: Data input starts from the start of the serial clock operation. • 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 • CKPmn = 0: Forward • 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 SCK00, SCK01, SCK10, and 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)

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 1 (clock input from the SCKp pin).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 00B (CSI mode).
The MDmn0 bit is set to 0 (interrupt source is transfer end).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 01B (reception only).
DAPmn and CKPmn bits select data and clock phase.
PTCmn1 and PTCmn0 bits set to 00B.
The DIRmn bit selects data transfer sequence (0: MSB first, 1: LSB first).
SLCmn1 and SLCmn0 bits are set to 00B.
DLSmn3 to DLSmn0 bits select data length (6H (7 bits) to FH (16 bits)).

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set bits 15 to 9 of the SDRmn register to 0000000B.
When operation is in progress (SEmn = 1): Receive data

Note Lower 8 bit in the SDRmn register: SDRpL register

(d) Serial output register m (SOm)

The register that not used in this mode.

(e) Serial output enable register m (SOEm)

The register that not used in this mode.

(f) Serial slave select enable register m (SSEm)

Set the SSEmn bit of the target channel to 1 (enables SSIp pin input).

(g) Serial channel start register m (SSm)

Set the SSmn bit of the target channel to 1 (operation start).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

(2) Operation Procedure

Figure 16-98. Initial Setting Procedure for Slave Reception

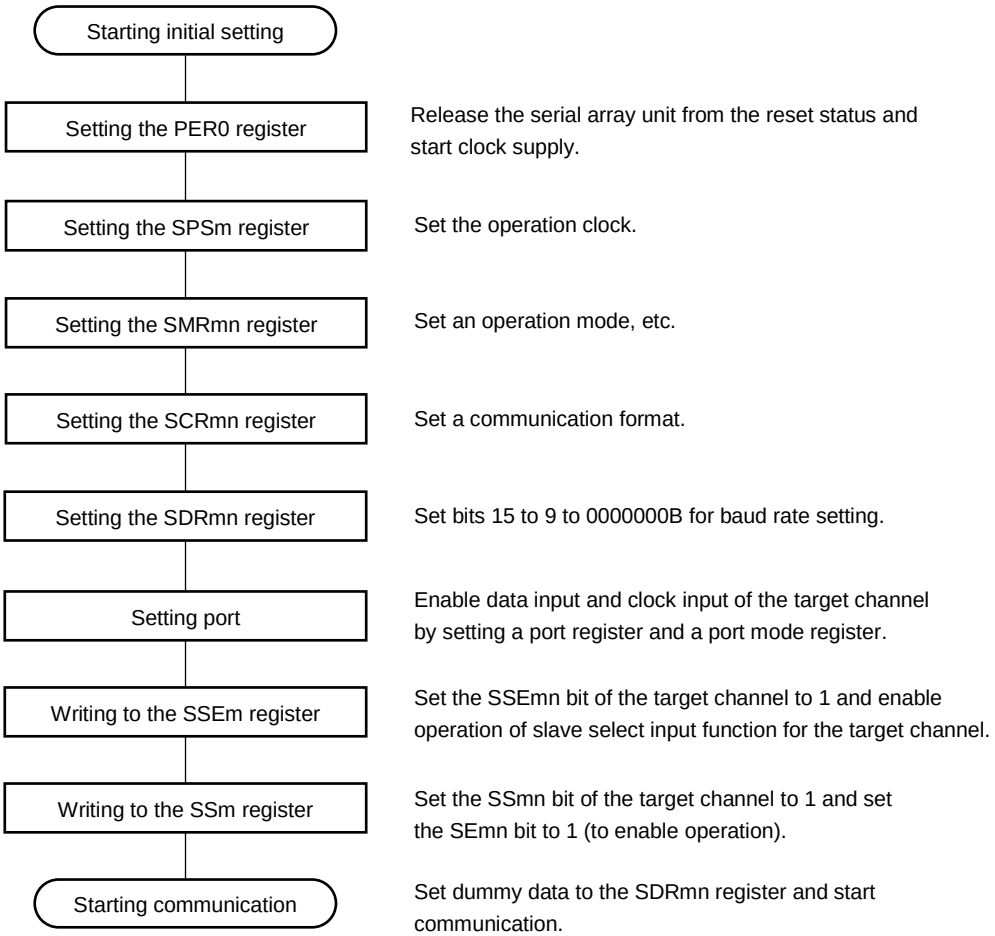
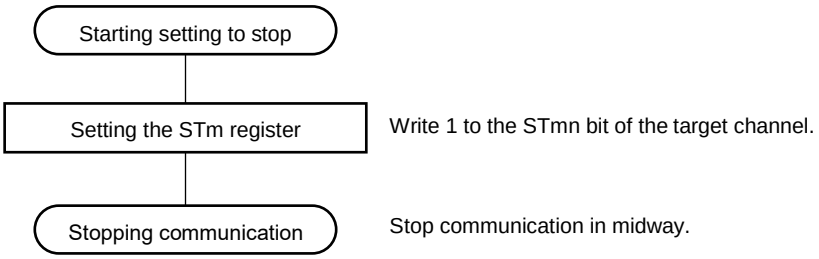
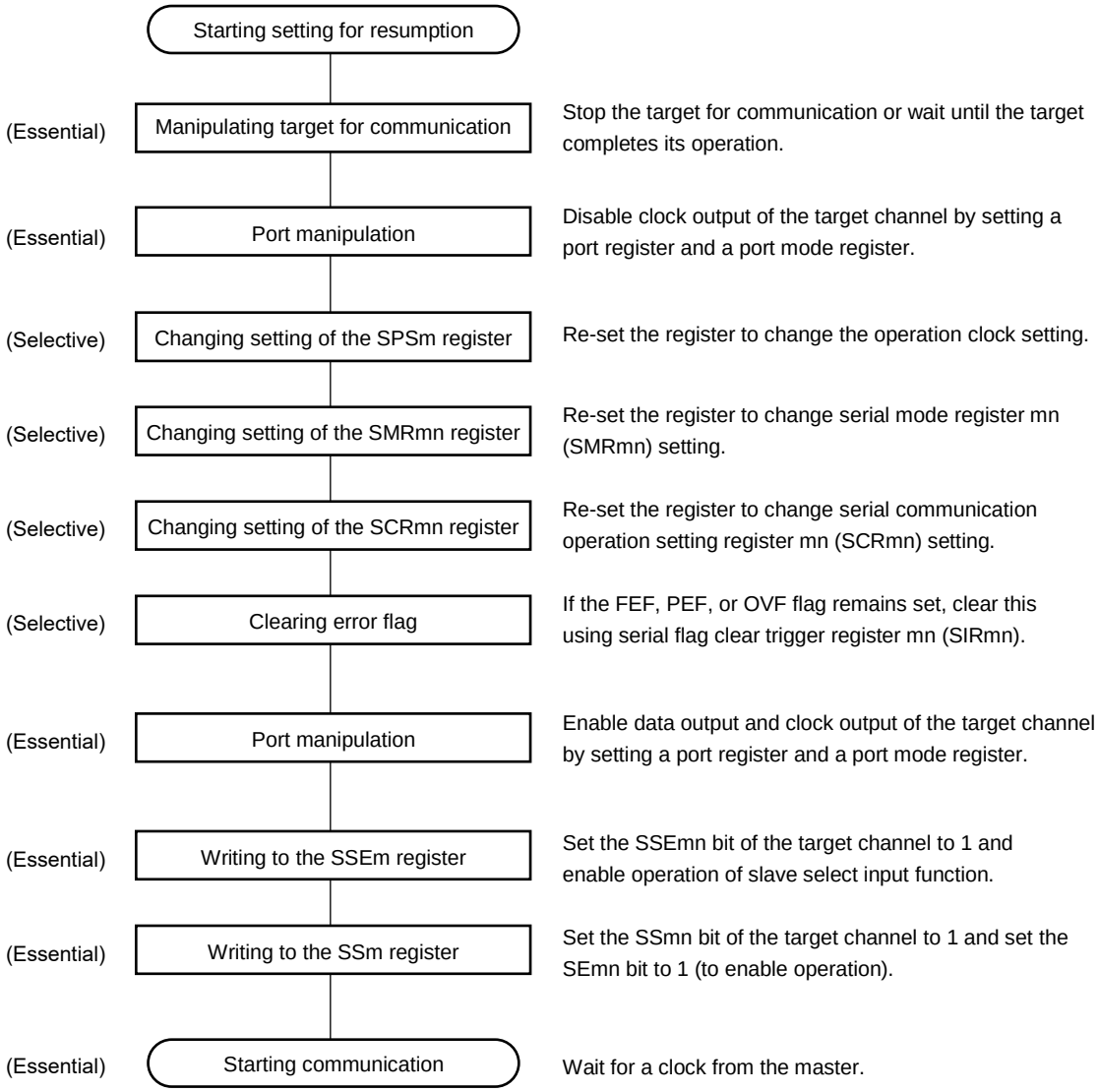


Figure 16-99. Procedure for Stopping Slave Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

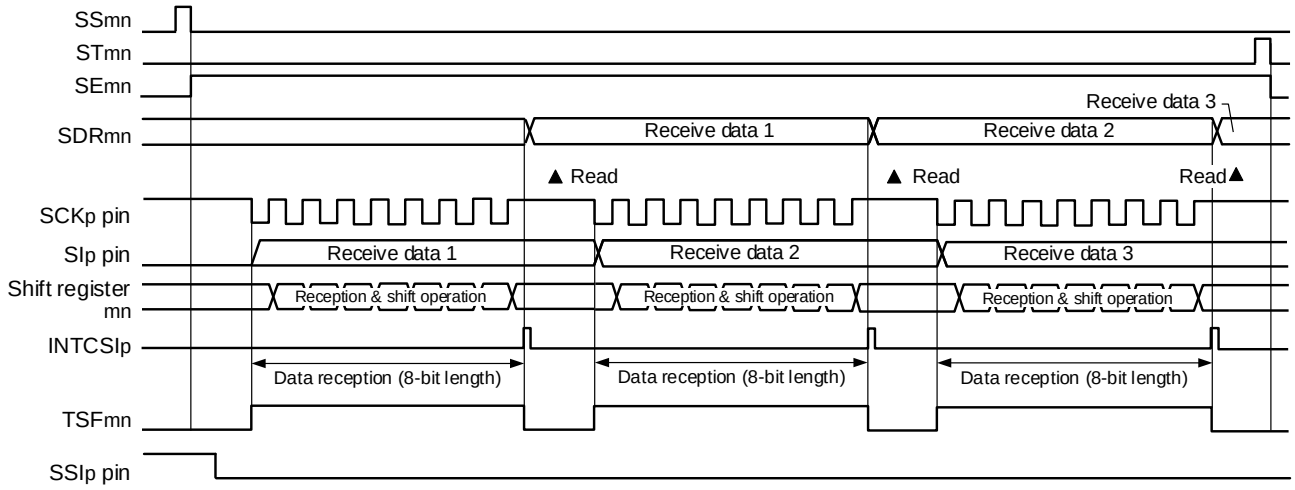
Figure 16-100. Procedure for Resuming Slave Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

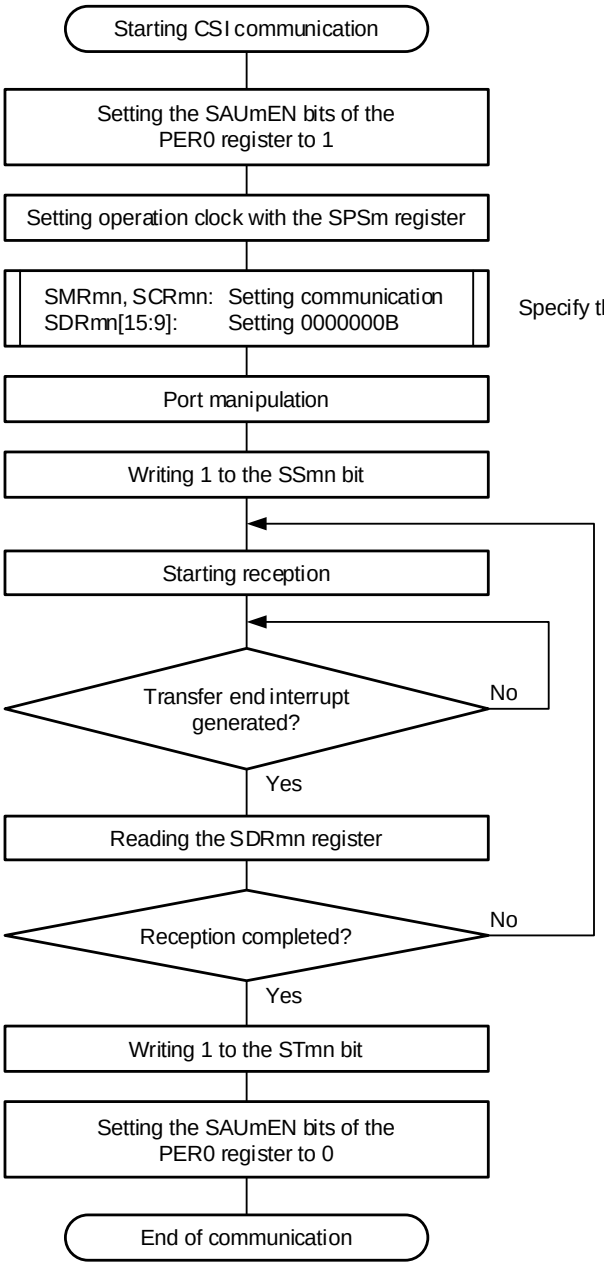
(3) Processing Flow (in Single-Reception Mode)

Figure 16-101. 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)

Figure 16-102. Flowchart of Slave Reception (in Single-Reception Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.6.6 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	SCK00, SI00, SO00, SSI00	SCK01, SI01, SO01, SSI01	SCK10, SI10, SO10, SSI10	SCK11, SI11, SO11, 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 • DAPmn = 0: Data I/O starts from the start of the serial clock operation. • 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 • CKPmn = 0: Forward • 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 SCK00, SCK01, SCK10, and 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)

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 1 (clock input from the SCKp pin).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 00B (CSI mode).
The MDmn0 bit selects interrupt source of channel n (0: transfer end, 1: buffer empty).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 11B (transmission/reception).
DAPmn and CKPmn bits select data and clock phase.
PTCmn1 and PTCmn0 bits set to 00B.
The DIRmn bit selects data transfer sequence (0: MSB first, 1: LSB first).
SLCmn1 and SLCmn0 bits are set to 00B.
DLSmn3 to DLSmn0 bits select data length (6H (7 bits) to FH (16 bits)).

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set bits 15 to 9 of the SDRmn register to 0000000B.
When operation is in progress (SEmn = 1): Transmit data setting/receive data in the SDRmn register.

Note Lower 8 bit in the SDRmn register: SDRpL register

(d) Serial output register m (SOm)

Set the SOmn bit of the target channel.

(e) Serial output enable register m (SOEm)

Set the SOEmn bit of the target channel to 1 (serial output enable).

(f) Serial slave select enable register m (SSEm)

Set the SSEmn bit of the target channel to 1 (enables SSIp pin input).

(g) Serial channel start register m (SSm)

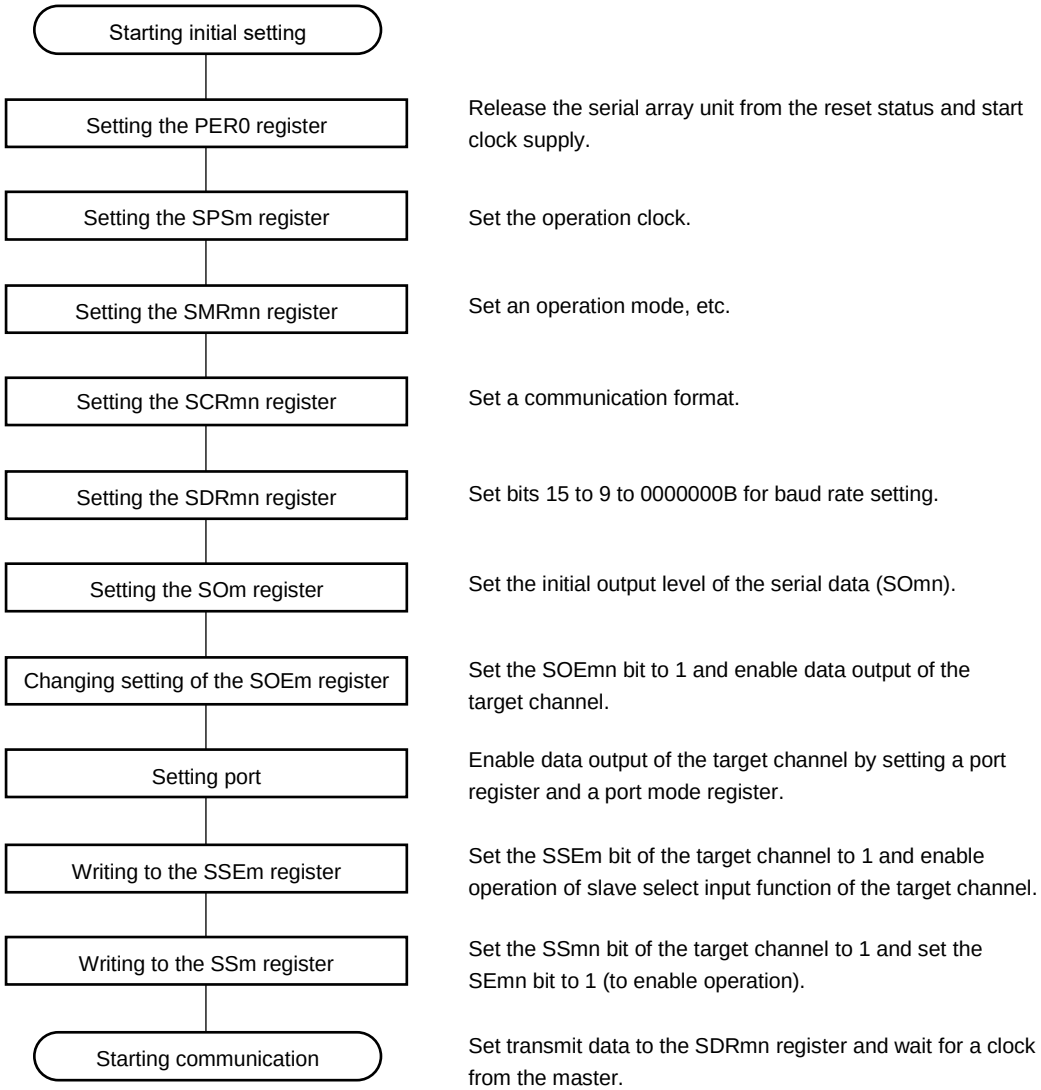
Set the SSmn bit of the target channel to 1 (operation start).

Caution Be sure to set the transmit data to SDRmn register before the serial clock is input from the master.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

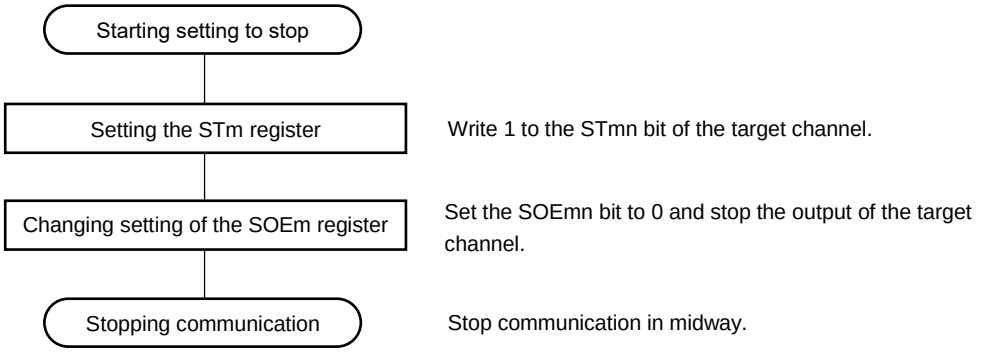
(2) Operation Procdure

Figure 16-103. Initial Setting Procedure for Slave Transmission/Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

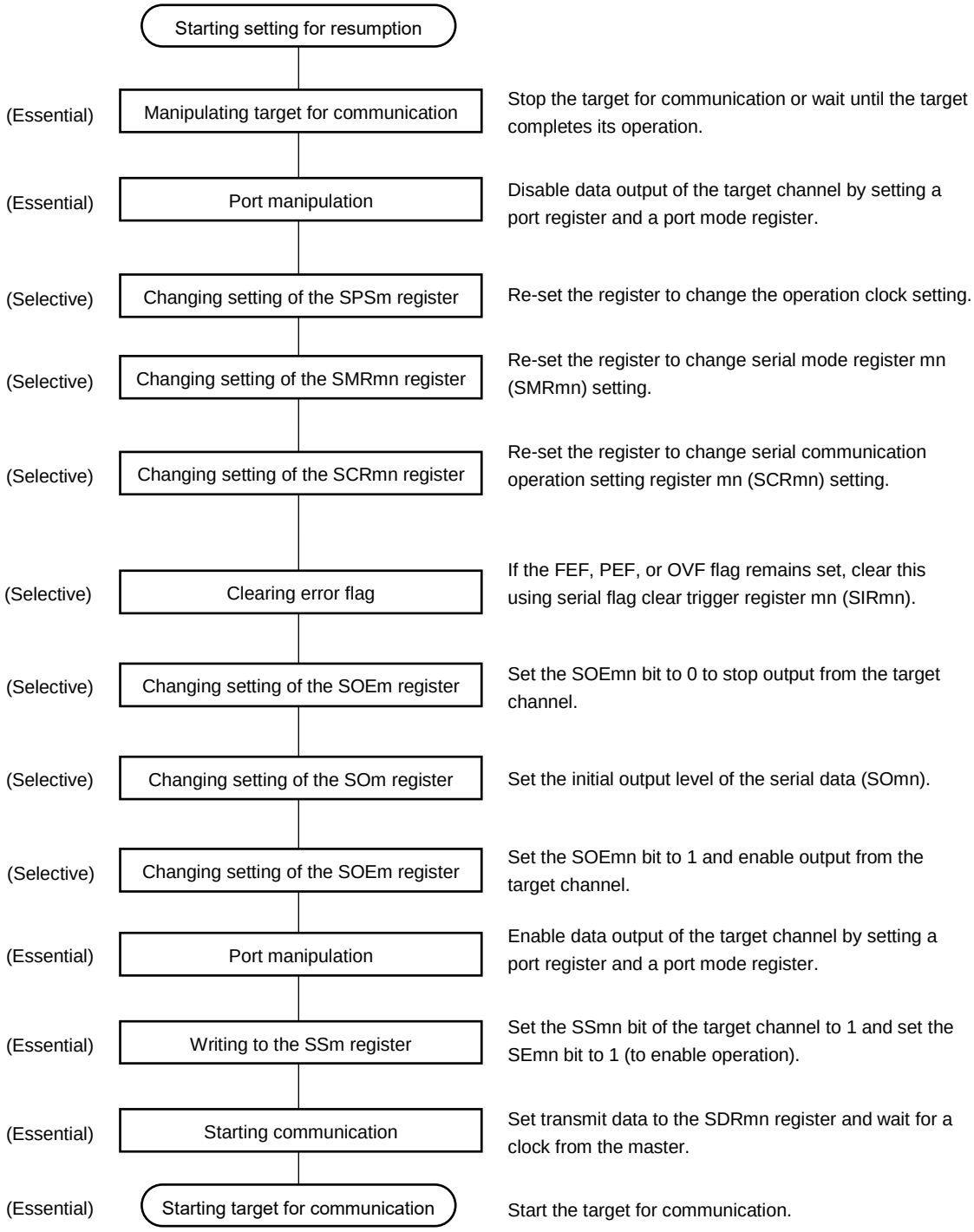
Figure 16-104. Procedure for Stopping Slave Transmission/Reception



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set SOm register (see **Figure 16-105. Procedure for Resuming Slave Transmission/Reception**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

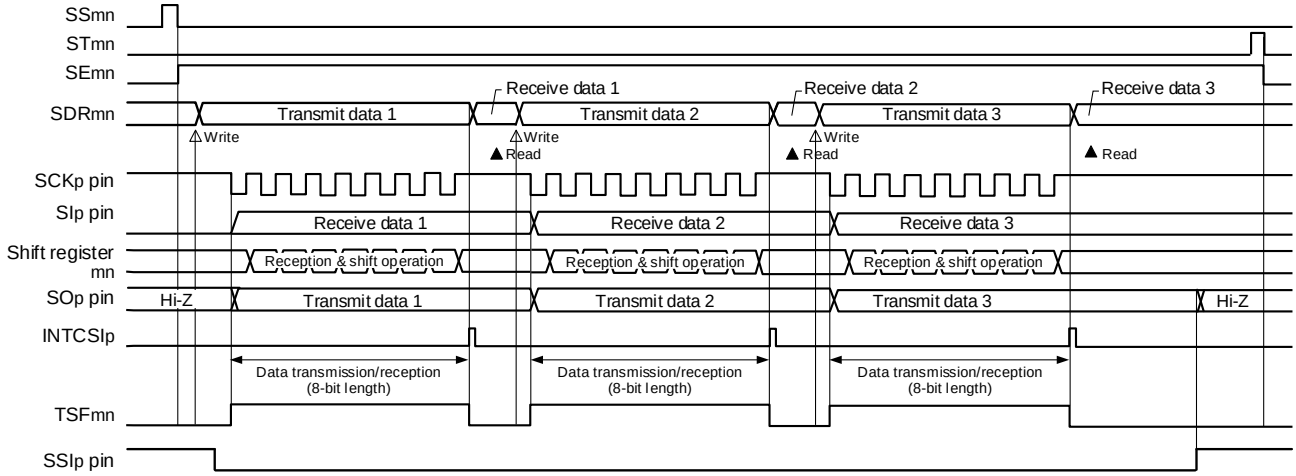
Figure 16-105. Procedure for Resuming Slave Transmission/Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

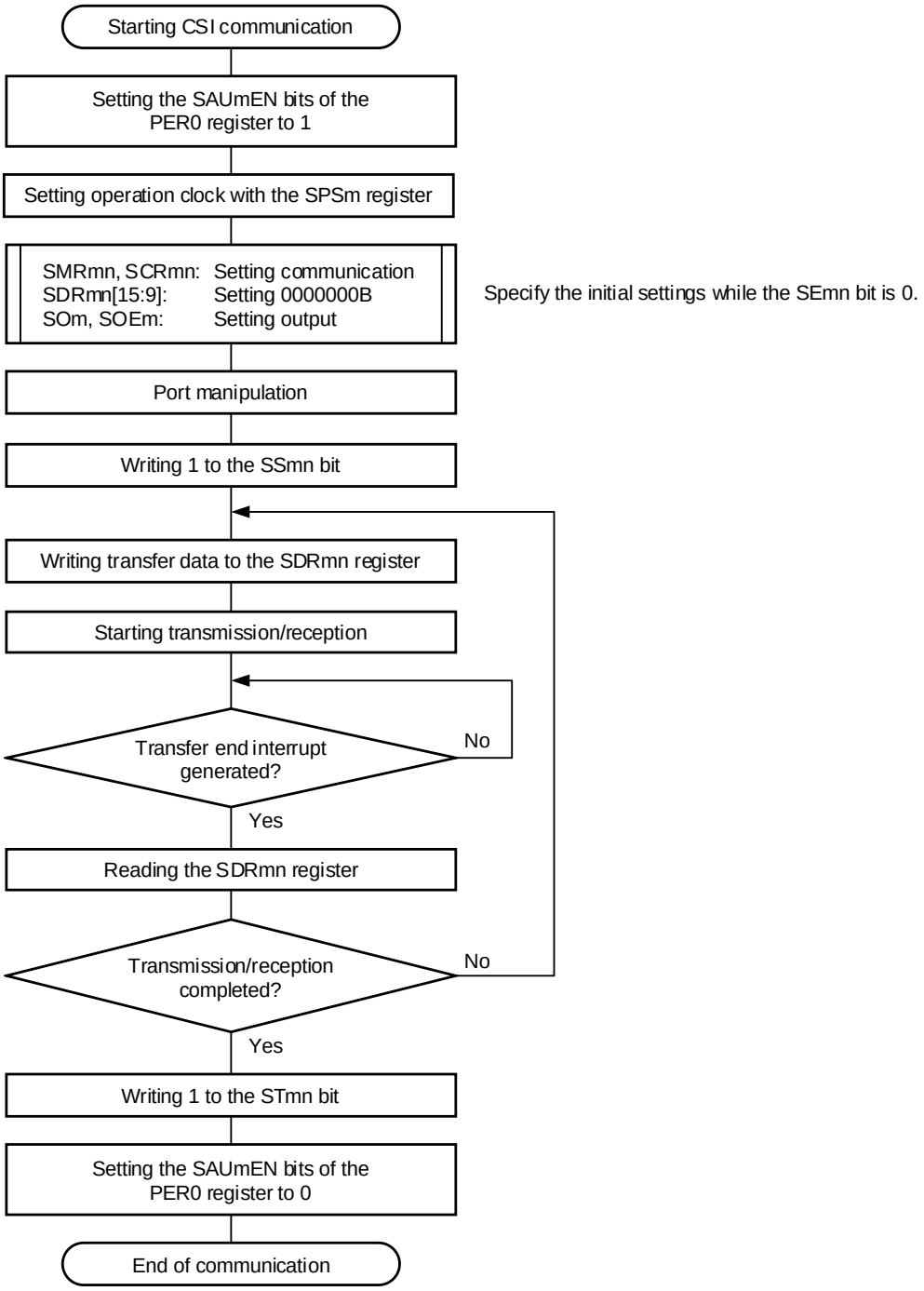
(3) Processing Flow (in Single-Transmission/Reception Mode)

Figure 16-106. 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)

Figure 16-107. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)

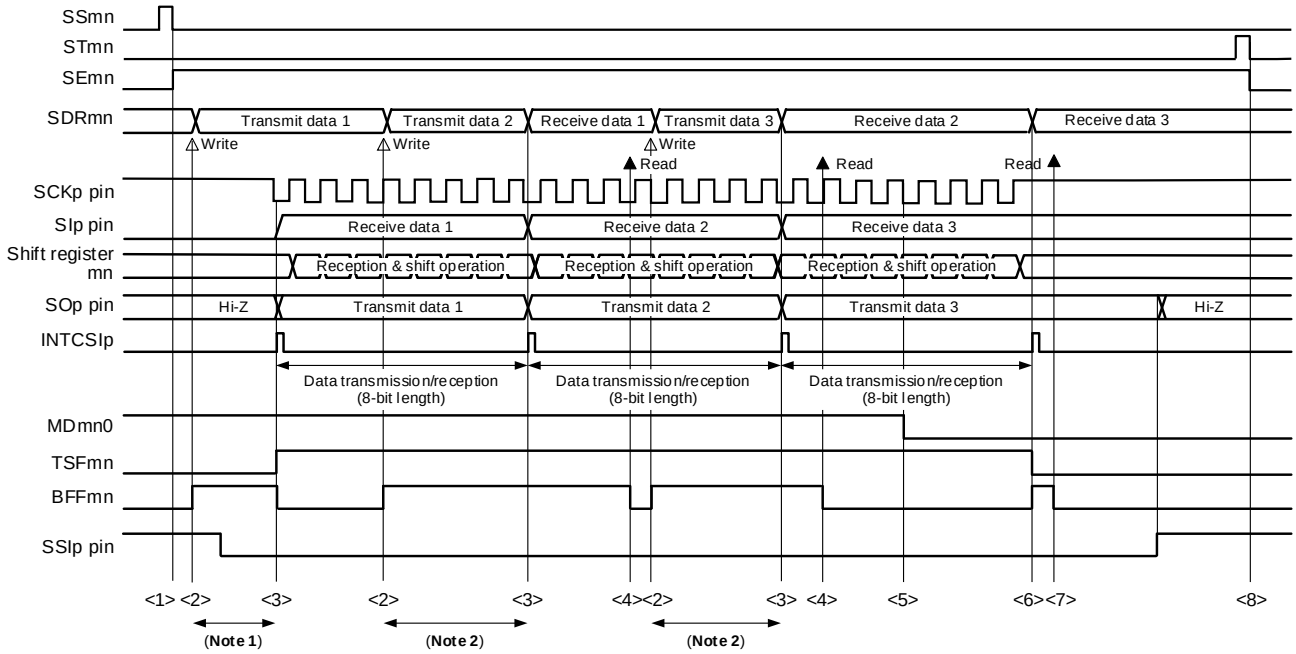


Caution Be sure to set the transmit data to SDRmn register before the serial clock is input from the master.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

(4) Processing Flow (in Continuous Transmission/Reception Mode)

Figure 16-108. 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.

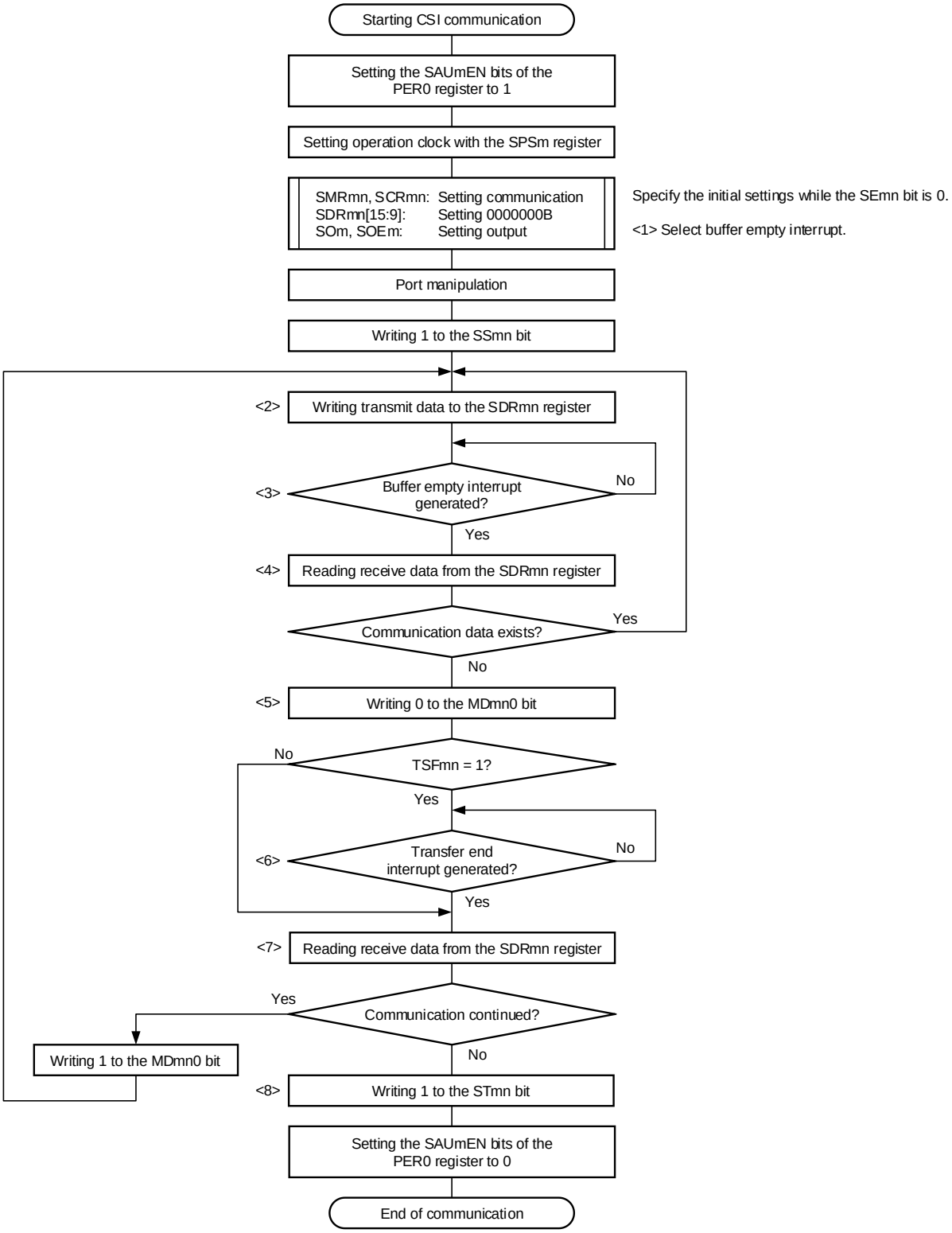
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 above figure correspond to <1> to <8> in **Figure 16-109**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11)

Figure 16-109. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution Be sure to set the transmit data to SDRmn register before the serial clock is input from the master.

Remarks 1. <1> to <8> in the above figure correspond to <1> to <8> in **Figure 16-108**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.6.7 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.

Remarks 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)

Table 16-4. 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	f_{CLK} = 40 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	32 MHz	40 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16 MHz	20 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8 MHz	10 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4 MHz	5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2 MHz	2.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1 MHz	1.25 MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500 kHz	625 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250 kHz	312.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125 kHz	156.25 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz	78.125 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz	39.0625 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz	19.53125 kHz
1	0	0	0	0	X	X	X	X	f_{CLK}	32 MHz	40 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16 MHz	20 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8 MHz	10 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4 MHz	5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2 MHz	2.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1 MHz	1.25 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500 kHz	625 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250 kHz	312.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125 kHz	156.25 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5 kHz	78.125 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25 kHz	39.0625 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63 kHz	19.53125 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)

16.6.8 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 16-110**.

Figure 16-110. Processing Procedure in Case of Overrun Error

Software Manipulation		Hardware Status	Remarks
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)

16.7 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]

- Transmission end interrupt/buffer empty interrupt
- Reception end 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.

- RL78/F25 Products and RL78/F22 48-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C	Simplified I^2S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function)		IIC01	–
1	0	CSI10 (supporting SPI function) Note 1	UART1	IIC10	Supported
	1	CSI11 (supporting SPI function)		IIC11	–

- RL78/F22 32-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C	Simplified I^2S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function)		IIC01	–
1	0	CSI10 (supporting SPI function) Note 1	UART1	IIC10	Supported
	1	–		–	–

- RL78/F22 24-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C	Simplified I^2S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function) Note 2		IIC01	–
1	0	–	UART1	–	Supported
	1	–		–	–

Notes 1. 48-pin and 32-pin products do not have SSI10 pin.
2. 24-pin products do not have SSI01 pin.

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.

.

UART performs the following four types of communication operations.

- UART transmission (See **16.7.1 UART Transmission**.)
- UART reception (See **16.7.2 UART Reception**.)
- LIN transmission (UART0 only) (See **16.8.1 LIN Transmission**.)
- LIN reception (UART0 only) (See **16.8.2 LIN Reception**.)

16.7.1 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 0 of SAU1
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) Reverse output (default: low level)	
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity	
Stop bit	The following selectable • Appending 1 bit • 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 = 0), mn = 00, 10

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 0 (operation clock selected by CKSmn bit).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 01B (UART mode).
The MDmn0 bit selects interrupt source of channel n (0: transfer end, 1: buffer empty).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 10B (transmission only).
DAPmn and CKPmn bits are set to 00B.
PTCmn1 and PTCmn0 bits select parity bit (00B: no parity, 01B: 0 parity, 10B: even parity, 11B: odd parity).
The DIRmn bit selects data transfer sequence (0: MSB first, 1: LSB first).
SLCmn1 and SLCmn0 bits select stop bit length (01B: 1 bit, 10B: 2 bits).
DLSmn3 to DLSmn0 bits select data length (6H (7 bits) to 8H (9 bits), and FH (16 bits)).

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set baud rate to bits 15 to 9 of the SDRmn register.
When operation is in progress (SEmn = 1): Set the transmit data in the SDRmn register.

Note Lower 8 bit in the SDRmn register: SDRmnL register.

(d) Serial output level register m (SOLm)

The SOLm0 bit selects inversion of the level of the transmit (0: normal (not inverted), 1: inverted).

(e) Serial output register m (SOm)

Set the SOm0 bit of the target channel.
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.

(f) Serial output enable register m (SOEm)

Set the SOEm0 bit of the target channel to 1 (serial output enable).

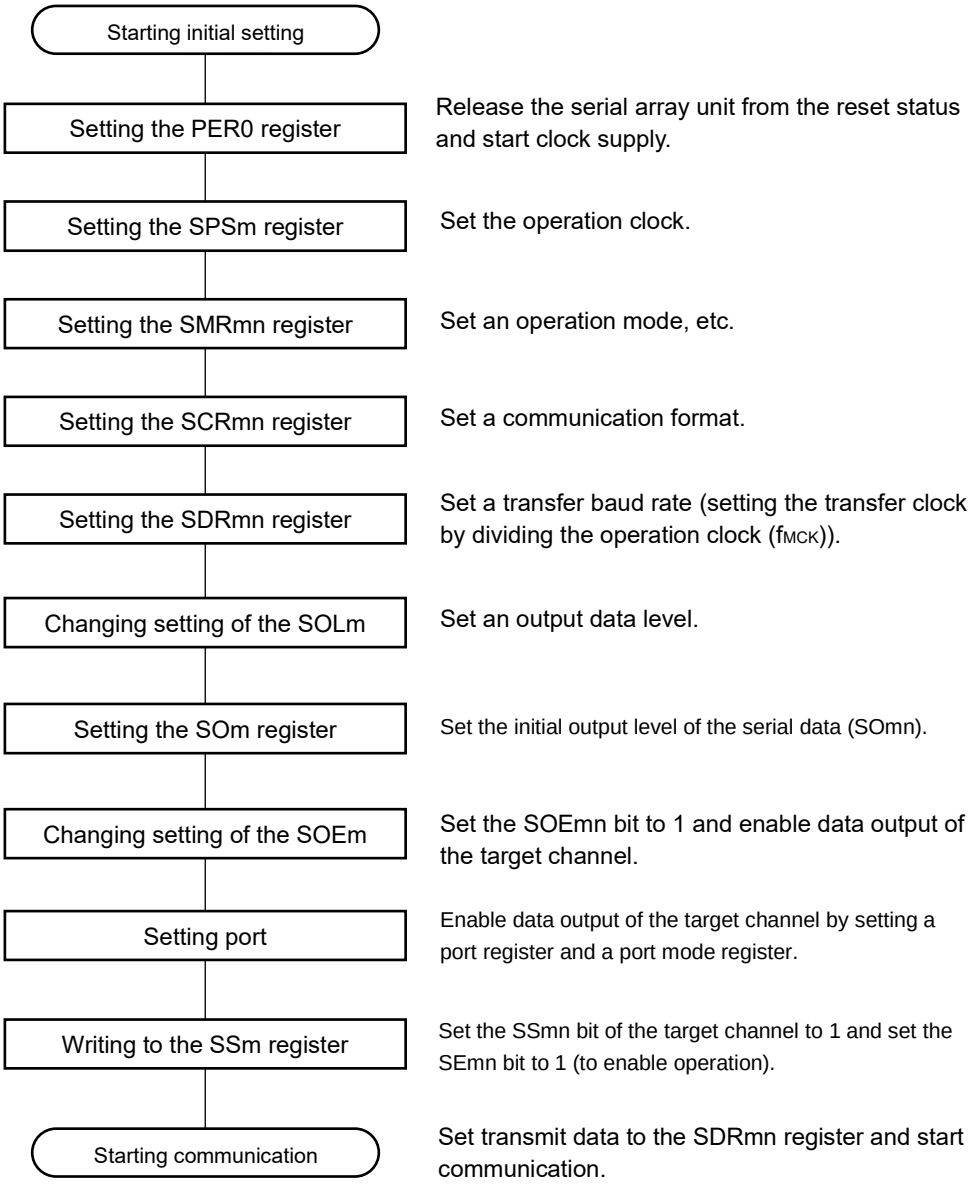
(g) Serial channel start register m (SSm)

Set the SSm0 bit of the target channel to 1 (operation start).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

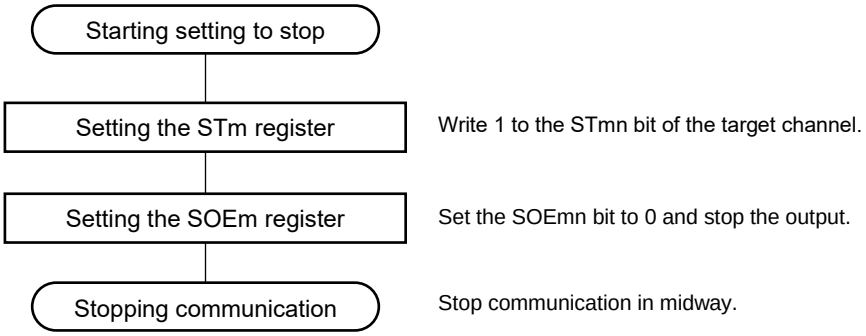
(2) Operation Procedure

Figure 16-111. Initial Setting Procedure for UART Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

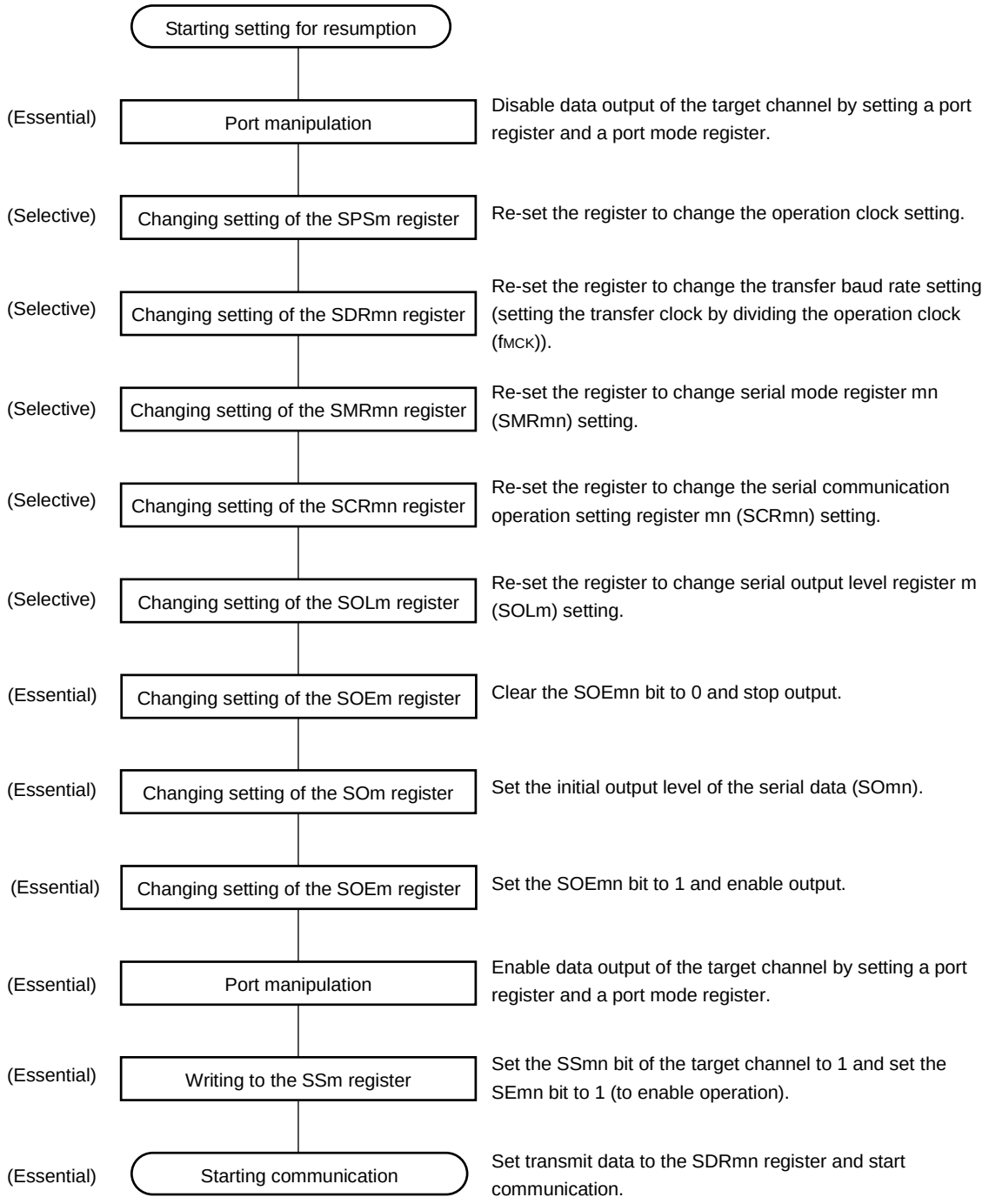
Figure 16-112. Procedure for Stopping UART Transmission



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 16-113. Procedure for Resuming UART Transmission**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

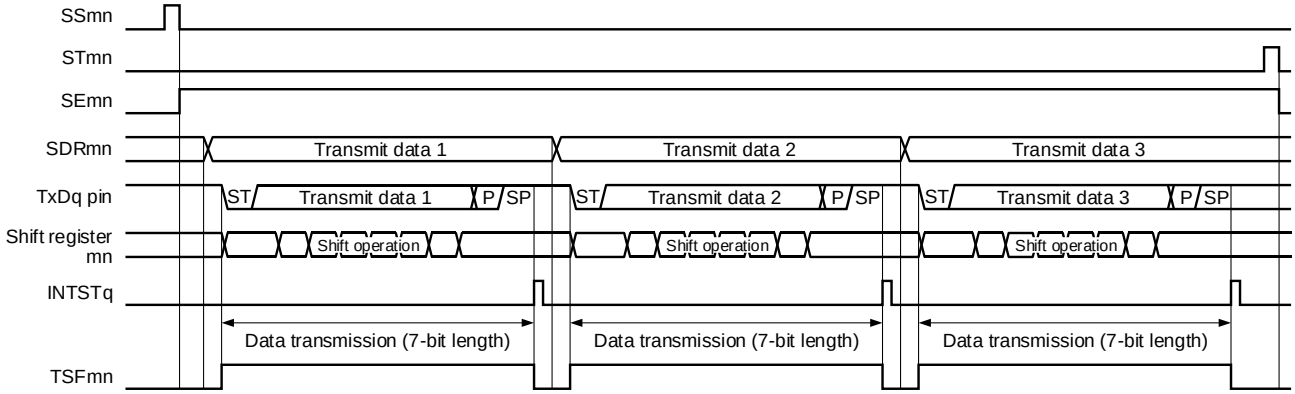
Figure 16-113. Procedure for Resuming UART Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

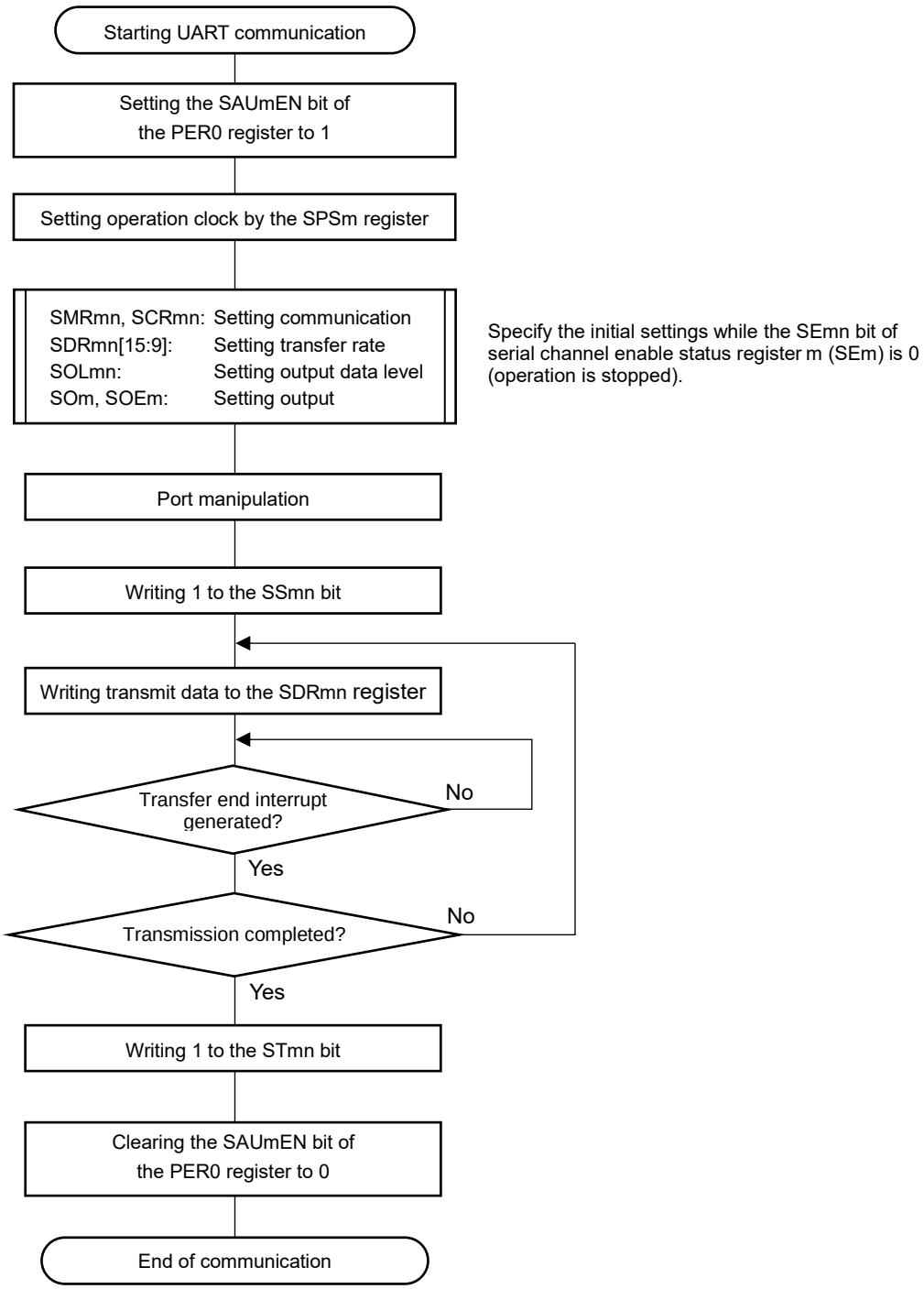
(3) Processing Flow (in Single-Transmission Mode)

Figure 16-114. Timing Chart of UART Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10, q: UART number (q = 0, 1)

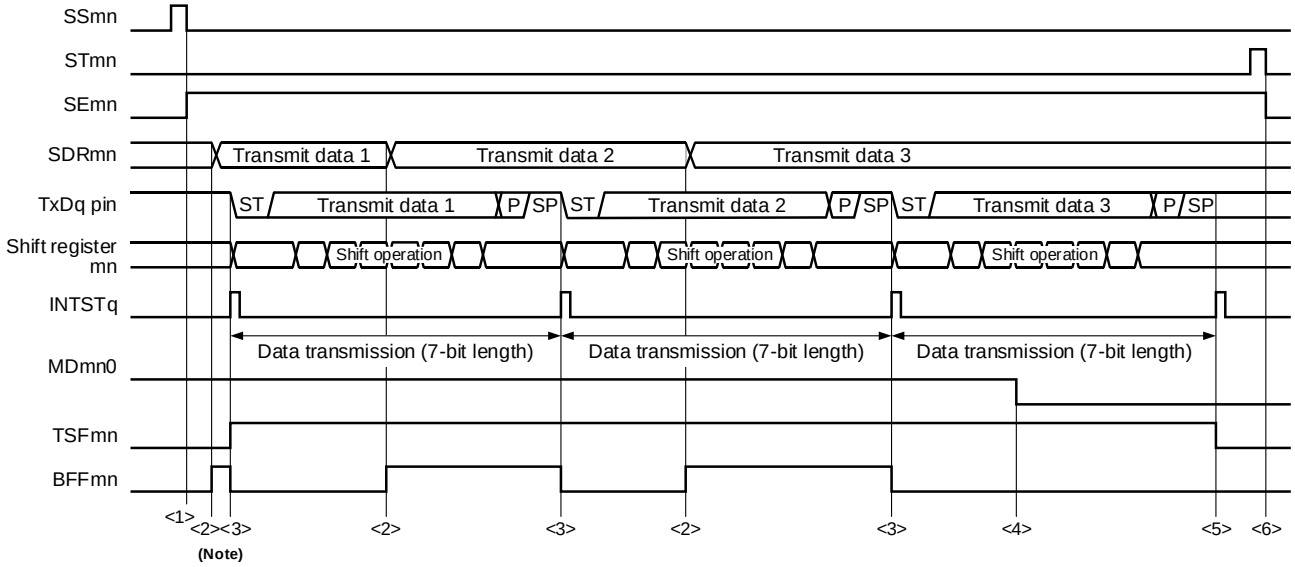
Figure 16-115. Flowchart of UART Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

(4) Processing Flow (in Continuous Transmission Mode)

Figure 16-116. Timing Chart of UART Transmission (in Continuous Transmission Mode)



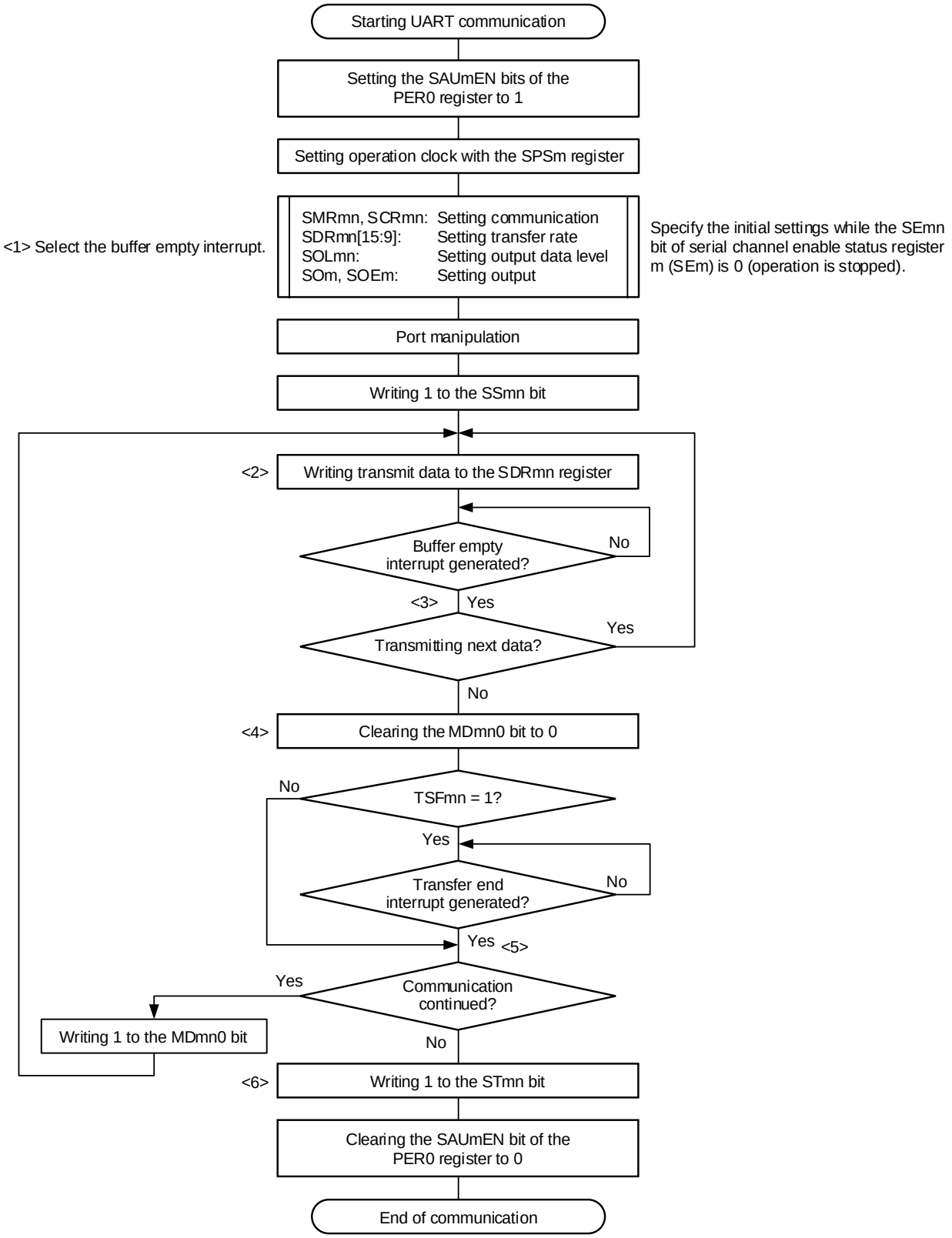
Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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 1. <1> to <6> in the figure correspond to <1> to <6> in **Figure 16-117**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10, q: UART number (q = 0, 1)

Figure 16-117. Flowchart of UART Transmission (in Continuous Transmission Mode)



Remarks 1. <1> to <6> in the figure correspond to <1> to <6> in **Figure 16-116**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0), mn = 00, 10

16.7.2 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) • Parity error detection flag (PEFmn) • 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) Reverse output (default: low level)	
Parity bit	The following selectable • No parity bit (no parity check) • Appending 0 parity (no parity check) • Appending even parity • 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

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 0 (operation clock selected by CKSmn bit).
The STSmn bit is set to 1 (valid edge of the RXDq pin).
The SISmn0 bit selects inversion of level of receive data (0: normal (not inverted), 1: inverted).
MDmn2 and MDmn1 bits set a value of 01B (UART mode).
The MDmn0 bit is set to 0 (interrupt source: transfer end).

(b) Serial mode register mn (SMRm0)

The CKSm0 bit sets the same value as the CKSmn bit.
The CCSm0 bit is set to 0.
The STSm0 bit is set to 0.
The SISm00 bit is set to 0.
MDm02 and MDm01 bits set a value of 01B (UART mode).
The MDm00 bit selects interrupt source of channel 0 (0: transfer end, 1: buffer empty).

(c) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 01B (reception only).
DAPmn and CKPmn bits are set to 00B.
PTCmn1 and PTCmn0 bits select parity bit (00B: no parity, 01B: no parity judgements, 10B: even parity, 11B: odd parity).
The DIRmn bit selects data transfer sequence (0: MSB first, 1: LSB first).
SLCmn1 and SLCmn0 bits are set to 01B (stop bit length: 1 bit).
DLSmn3 to DLSmn0 bits select data length (6H (7 bits) to 8H (9 bits), and FH (16 bits)).

(d) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set baud rate to bits 15 to 9 of the SDRmn register.
When operation is in progress (SEmn = 1): Receive data

Note Lower 8 bit in the SDRmn register: SDRmnL register.

(e) Serial output level register m (SOLm)

The register that not used in this mode.

(f) Serial output register m (SOm)

The register that not used in this mode.

(g) Serial output enable register m (SOEm)

The register that not used in this mode.

(h) Serial channel start register m (SSm)

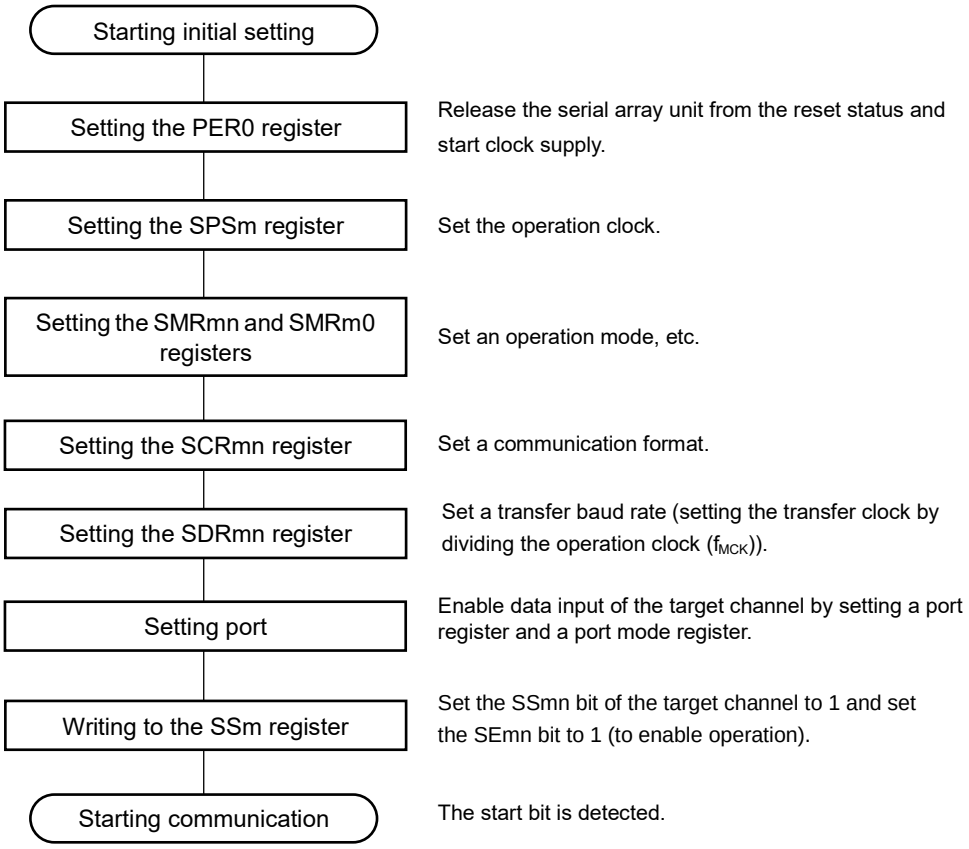
Set the SSmn bit of the target channel to 1 (operation start).

Caution For the UART reception, be sure to set the SMRm0 register paired with channel n (n = 1).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 01, 11, q: UART number (q = 0, 1)

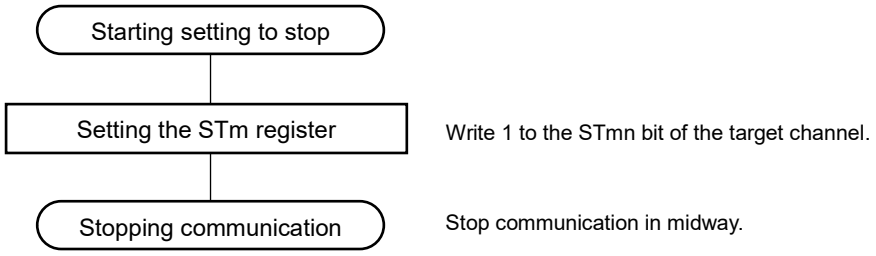
(2) Operation Procedure

Figure 16-118. Initial Setting Procedure for UART Reception



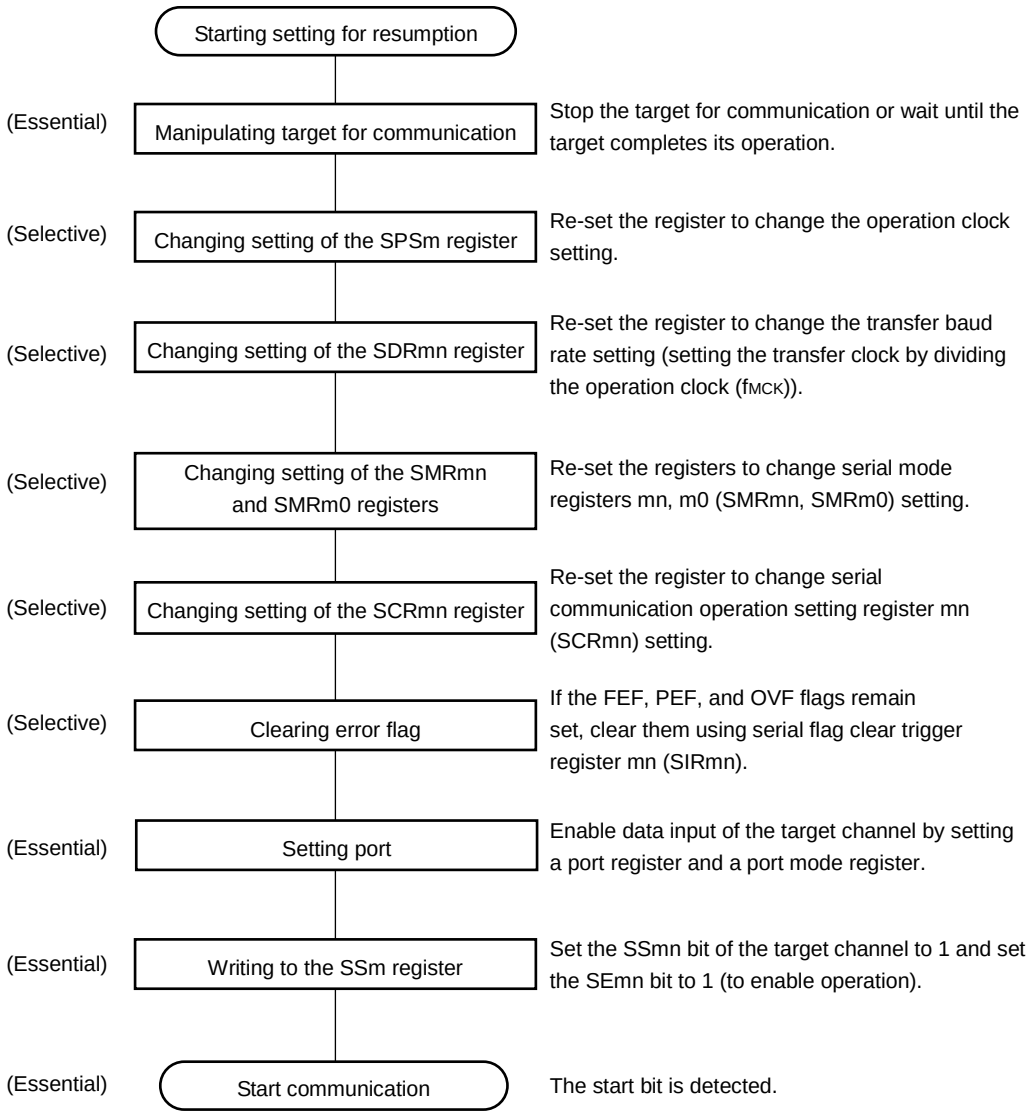
Caution For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{MCK} clocks have elapsed.

Figure 16-119. Procedure for Stopping UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 01, 11

Figure 16-120. Procedure for Resuming UART Reception

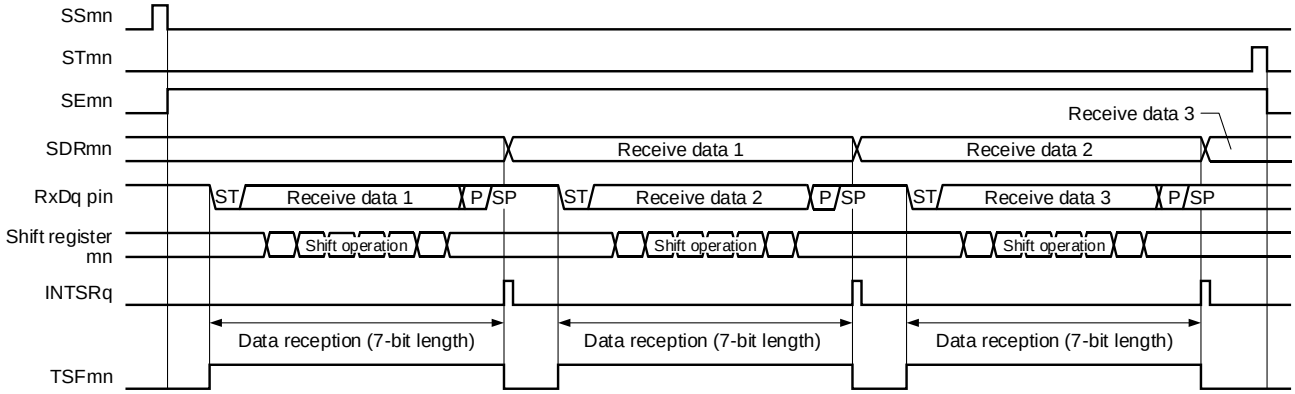


Caution For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{MCK} clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 01, 11

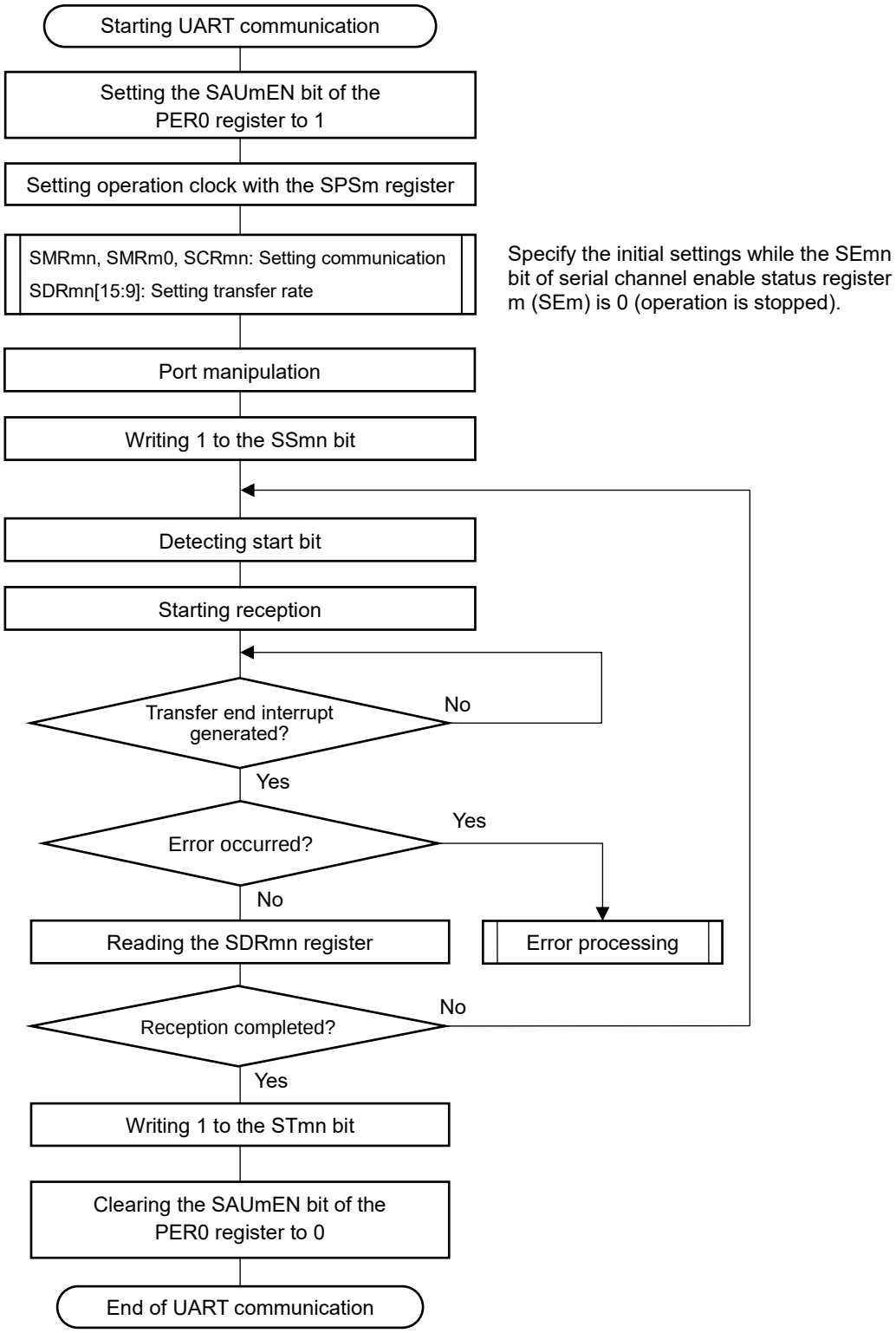
(3) Processing Flow

Figure 16-121. Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 01, 11, q: UART number (q = 0, 1)

Figure 16-122. 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

16.7.3 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)

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 16-5. 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	f_{CLK} = 40 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	32 MHz	40 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16 MHz	20 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8 MHz	10 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4 MHz	5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2 MHz	2.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1 MHz	1.25 MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500 kHz	625 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250 kHz	312.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125 kHz	156.25 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz	78.125 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz	39.0625 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz	19.53125 kHz
1	0	0	0	0	X	X	X	X	f_{CLK}	32 MHz	40 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16 MHz	20 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8 MHz	10 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4 MHz	5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2 MHz	2.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1 MHz	1.25 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500 kHz	625 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250 kHz	312.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125 kHz	156.25 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5 kHz	78.125 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25 kHz	39.0625 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63 kHz	19.53125 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)

(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 or 40 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	$f_{CLK}/2^{10}$	51	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^{9}$	51	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^{8}$	51	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^{7}$	51	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^{6}$	51	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^{5}$	51	9615.38 bps	+0.16 %
19200 bps	$f_{CLK}/2^{4}$	51	19230.77 bps	+0.16 %
31250 bps	$f_{CLK}/2^{3}$	63	31250.00 bps	±0.00 %
38400 bps	$f_{CLK}/2^{3}$	51	38461.54 bps	+0.16 %
76800 bps	$f_{CLK}/2^{2}$	51	76923.08 bps	+0.16 %
153600 bps	$f_{CLK}/2$	51	153846.15 bps	+0.16 %
312500 bps	f_{CLK}	50	313725.49 bps	+0.39 %

UART Baud Rate (Target Baud Rate)	f_{CLK} = 40 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^{10}$	64	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^{9}$	64	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^{8}$	64	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^{7}$	64	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^{6}$	64	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^{5}$	64	9615.38 bps	+0.16 %
19200 bps	$f_{CLK}/2^{4}$	64	19230.77 bps	+0.16 %
31250 bps	$f_{CLK}/2^{3}$	79	31250.00 bps	±0.00 %
38400 bps	$f_{CLK}/2^{3}$	64	38461.54 bps	+0.16 %
76800 bps	$f_{CLK}/2^{2}$	64	76923.08 bps	+0.16 %
153600 bps	$f_{CLK}/2$	64	153846.15 bps	+0.16 %
312500 bps	f_{CLK}	63	312500.00 bps	±0.00 %

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.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

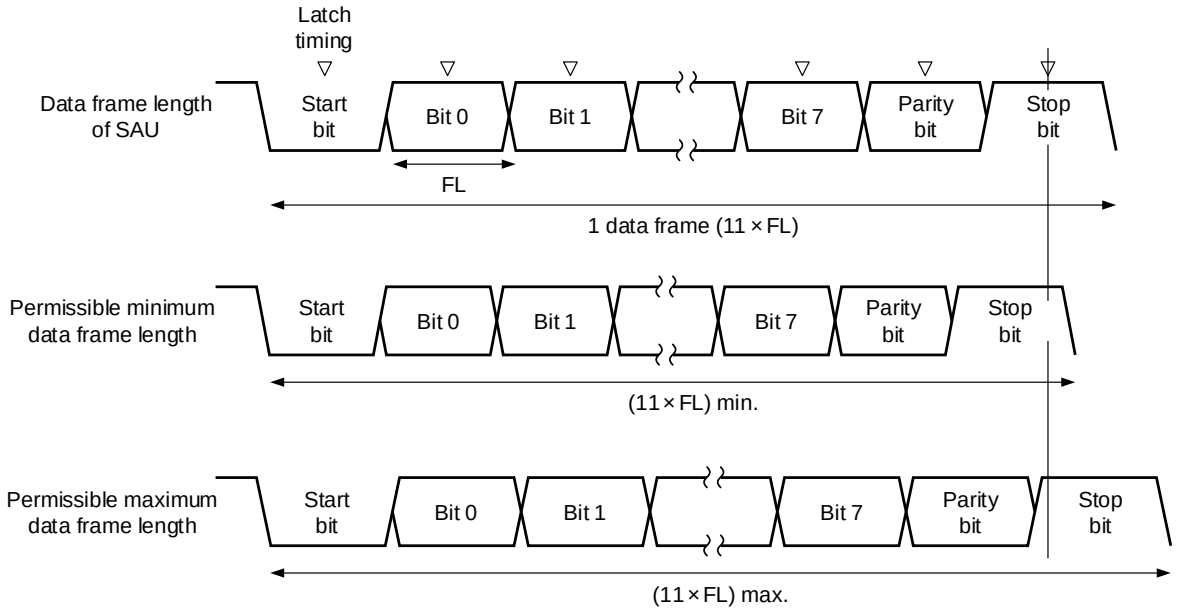
Brate: Calculated baud rate value at the reception side (See **16.7.3 (1) Baud Rate Calculation Expression**.)

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 16-123. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in **Figure 16-123**, 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.

16.7.4 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 **Figure 16-124** and **Figure 16-125**.

Figure 16-124. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation		Hardware Status	Remarks
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 16-125. Processing Procedure in Case of Framing Error

Software Manipulation		Hardware Status	Remarks
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 = 1), mn = 01, 11

16.8 LIN Communication

16.8.1 LIN Transmission

Of the UART transmissions, UART0 supports 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) 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 19.2 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.

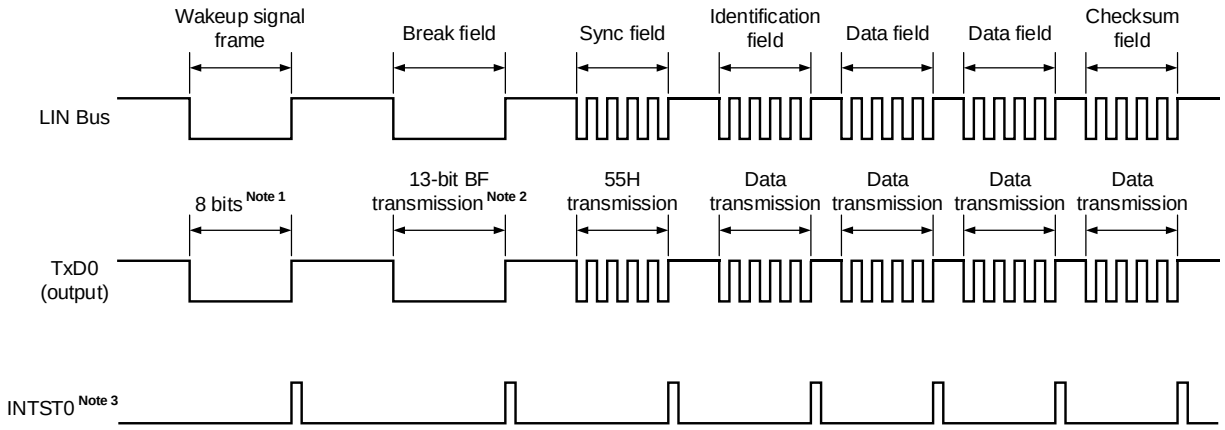
Usually, the master is connected to a network such as CAN (Controller Area Network).

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 16-126 outlines a transmission operation of LIN.

Figure 16-126. 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.

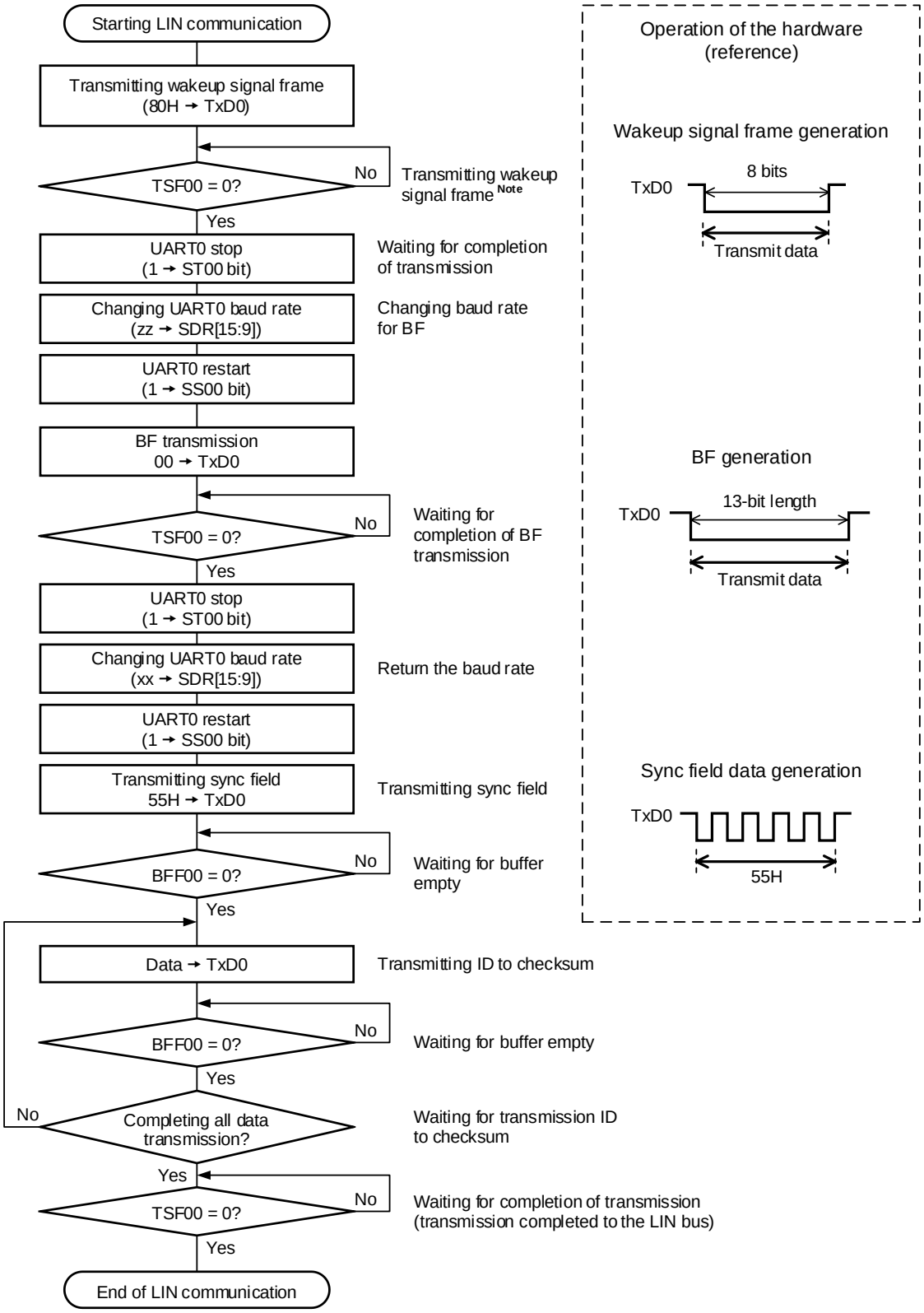
(Baud rate of break field) = 9/13 × N

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 16-127. 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.

16.8.2 LIN Reception

Of the UART reception, UART0 supports LIN communication.

For LIN reception, channel 1 of unit 0 is used.

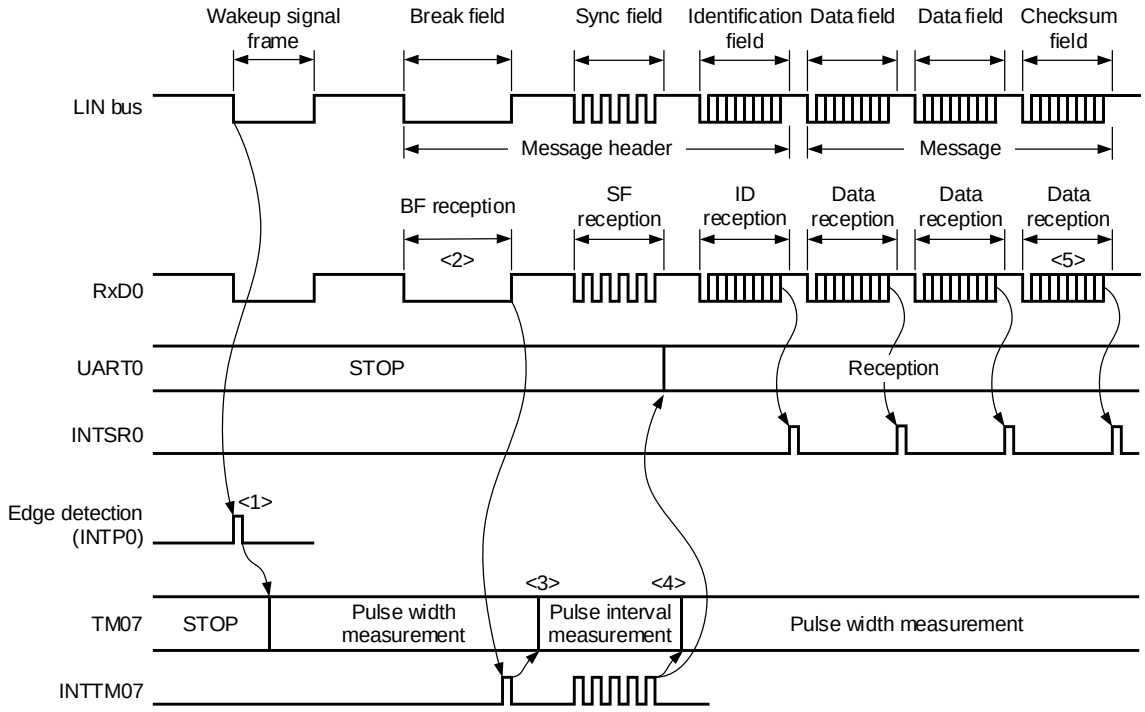
UART	UART0	UART1
Support of LIN communication	Supported	Not supported
Target channel	Channel 1 of SAU0	—
Pins used	RXD0	—
Interrupt	INTSR0	—
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	• Framing error detection flag (FEF01) • Overrun error detection 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) • 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 16-128 outlines a reception operation of LIN.

Figure 16-128. 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.

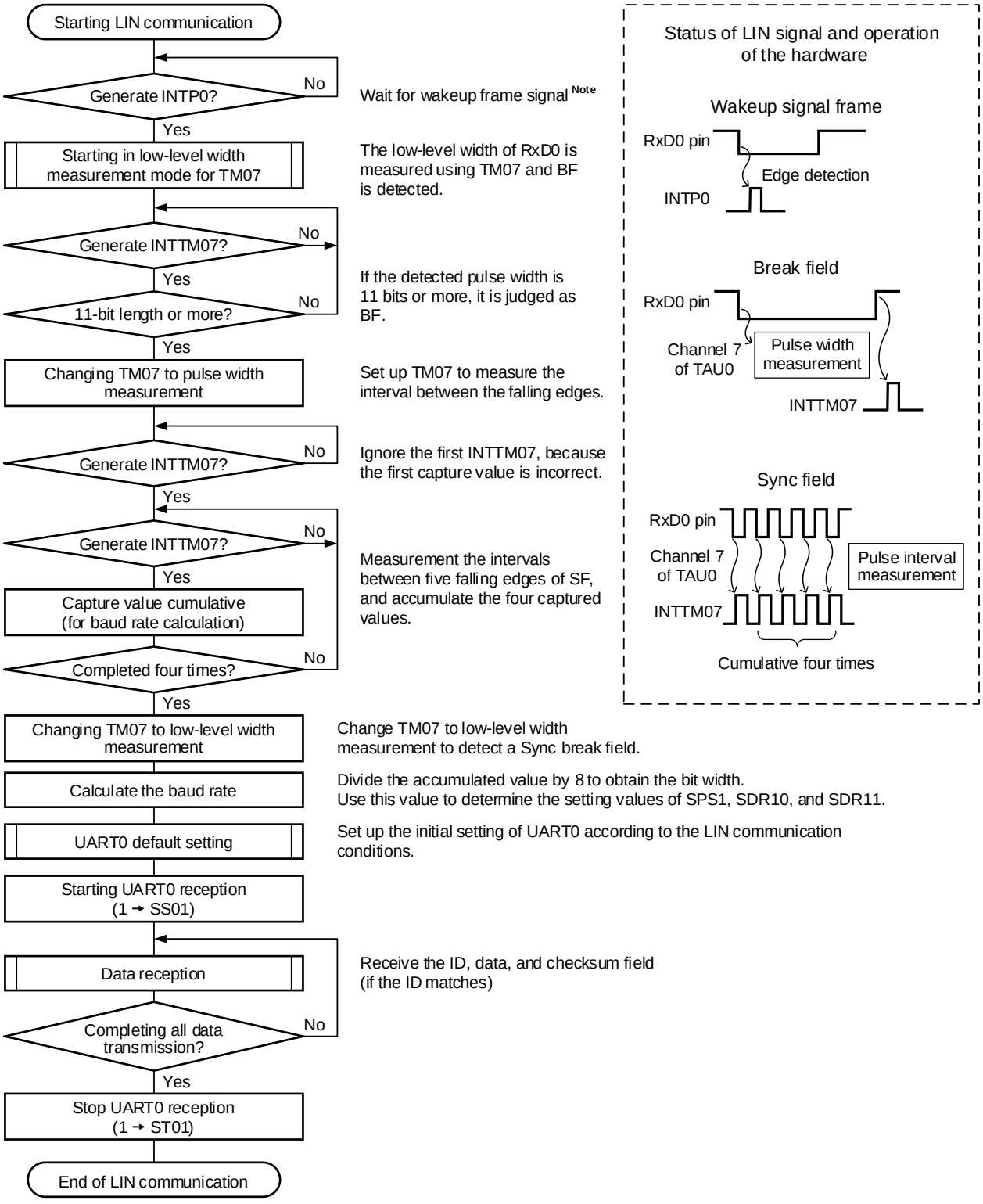
<2> TM07 starts measuring the low-level width upon detection of the falling edge of the BF signal, and then captures the data upon detection of the rising edge of the BF signal. The captured data is used to judge whether it is the BF signal.

<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. (see **6.7.4 Operation as Input Pulse Interval Measurement**).

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART0 once and adjust (re-set) the baud rate.

<5> 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 16-129. Flowchart for LIN Reception



Note Required in the sleep status only.

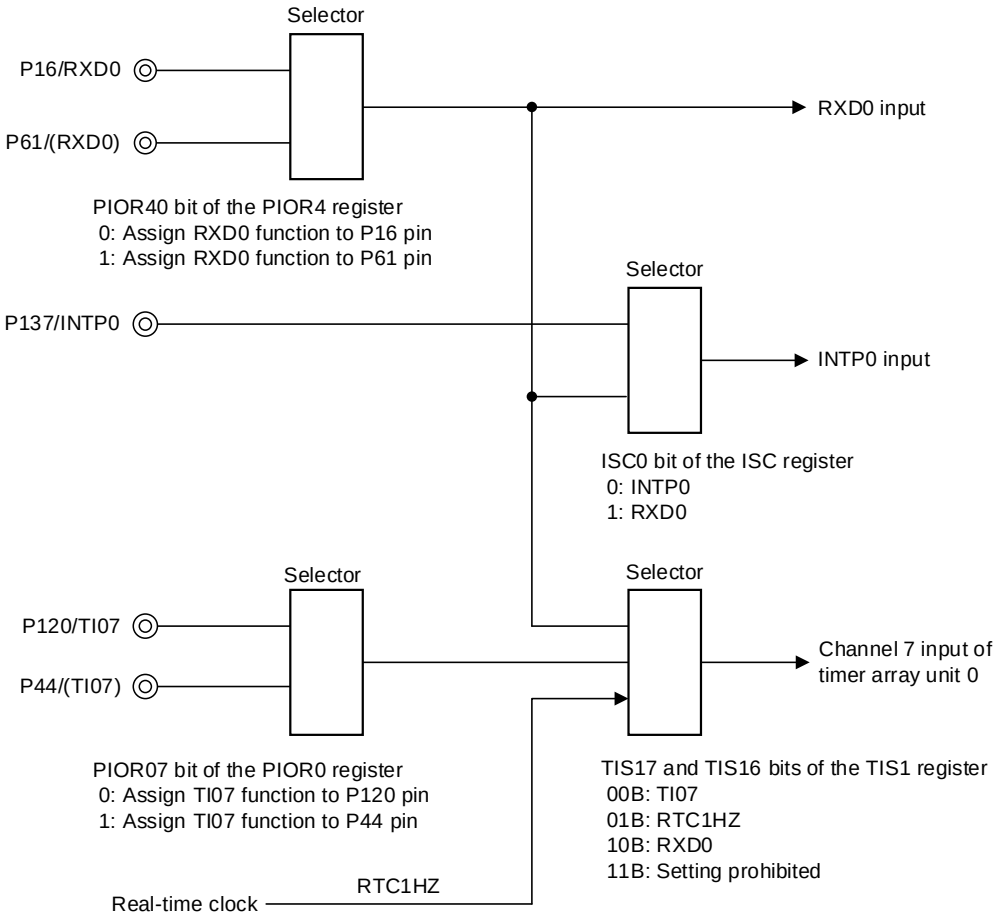
Caution For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{MCK} clocks have elapsed.

Figure 16-130 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 16-130. Port Configuration for Manipulating Reception of LIN



Remark ISC0: Bit 0 of the input switch control register (ISC) (See **Figure 16-19**.)

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)

16.9 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 **16.9.3 (2) Processing Flow** for details.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

The channel supporting simplified I²C (IIC00, IIC01, IIC10, IIC11) is channels 0 and 1 of SAU0 and channels 0 and 1 of SAU1.

- RL78/F25 Products and RL78/F22 48-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C	Simplified I²S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function)		IIC01	–
1	0	CSI10 (supporting SPI function) Note 2	UART1	IIC10	Supported
	1	CSI11 (supporting SPI function)		IIC11	–

- RL78/F22 32-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C	Simplified I²S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function)		IIC01	–
1	0	CSI10 (supporting SPI function) Note 2	UART1	IIC10	Supported
	1	–		–	–

- RL78/F22 24-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C	Simplified I²S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function) Note 1		IIC01	–
1	0	–	UART1	–	Supported
	1	–		–	–

Notes 1. 24-pin products do not have SSI01 pin.

2. 48-pin and 32-pin products do not have SSI10 pin.

Simplified I²C (IIC00, IIC01, IIC10, IIC11) performs the following four types of communication operations.

- Address field transmission (See **16.9.1 Address Field Transmission**.)
- Data transmission (See **16.9.2 Data Transmission**.)
- Data reception (See **16.9.3 Data Reception**.)
- Stop condition generation (See **16.9.4 Stop Condition Generation**.)

16.9.1 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 However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • 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_{DD} tolerance) mode (POMxx = 1) for the port output mode registers (POMx) (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_{DD} tolerance) mode (POMxx = 1) also for the clock 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)

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 0 (operation clock selected by CKSmn bit).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 10B (simplified I2C mode).
The MDmn0 bit is set to 0 (interrupt source: transfer end).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 10B (transmission only).
DAPmn and CKPmn bits are set to 00B.
PTCmn1 and PTCmn0 bits are set to 00B.
The DIRmn bit is set to 0.
SLCmn1 and SLCmn0 bits are set to 01B (stop bit length: 1 bit).
DLSmn3 to DLSmn0 bits are set to 0111B (data length: 8 bits).

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set baud rate to bits 15 to 9 of the SDRmn register.
When operation is in progress (SEmn = 1): Transmit data setting (address + R/W)

Note Lower 8 bit in the SDRmn register: SDRrL register.

(d) Serial output register m (SOm)

Set values in the CKOmn and SOmn bits.
Start condition is generated by manipulating the SOmn and CKOmn bits.

(e) Serial output enable register m (SOEm)

Sets a value in the SOEmn bit.
SOEmn = 0 until the start condition is generated, and SOEmn = 1 after generation.

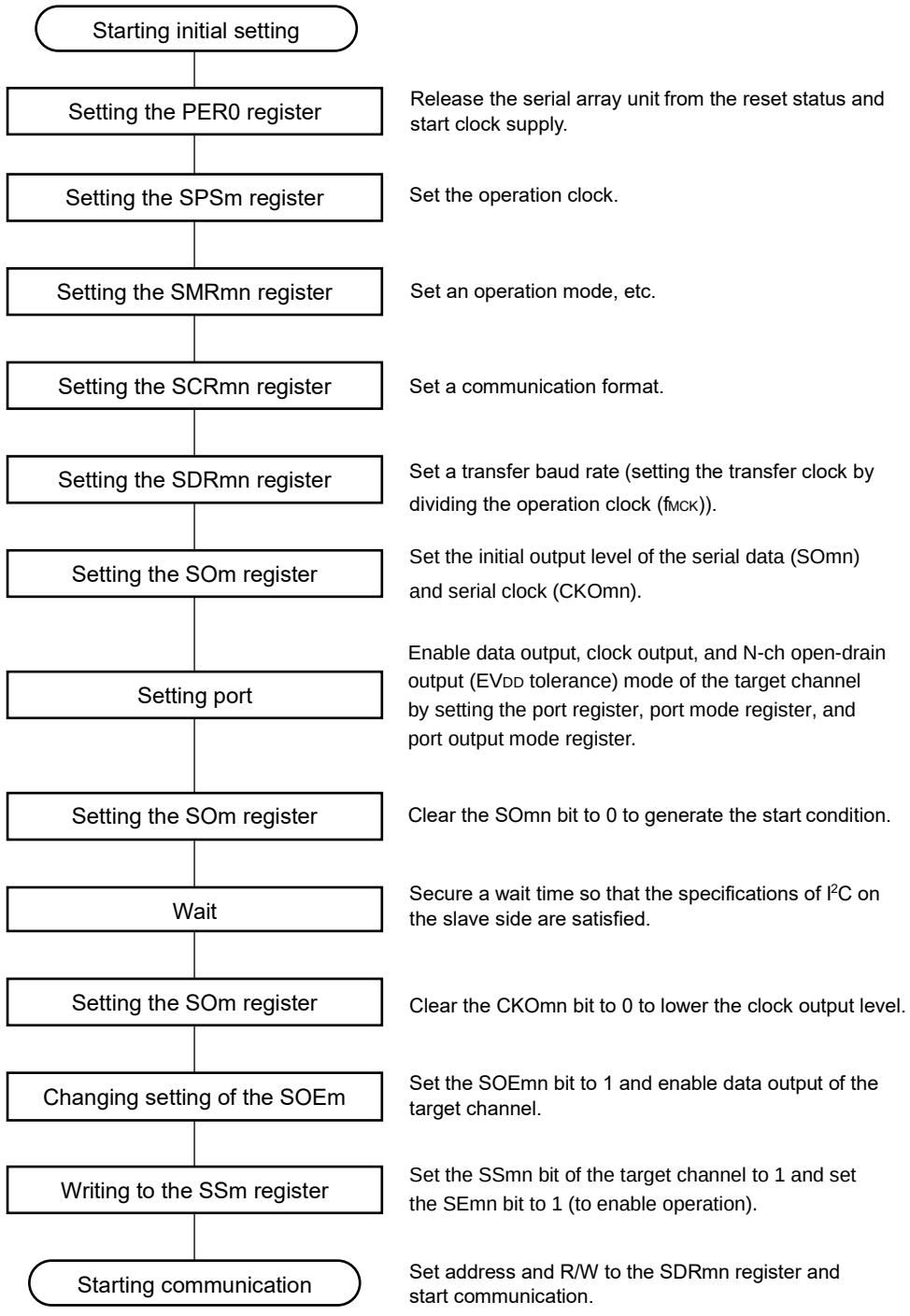
(f) Serial channel start register m (SSm)

Set the SSmn bit of the target channel to 1 (operation start).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11)

(2) Operation Procedure

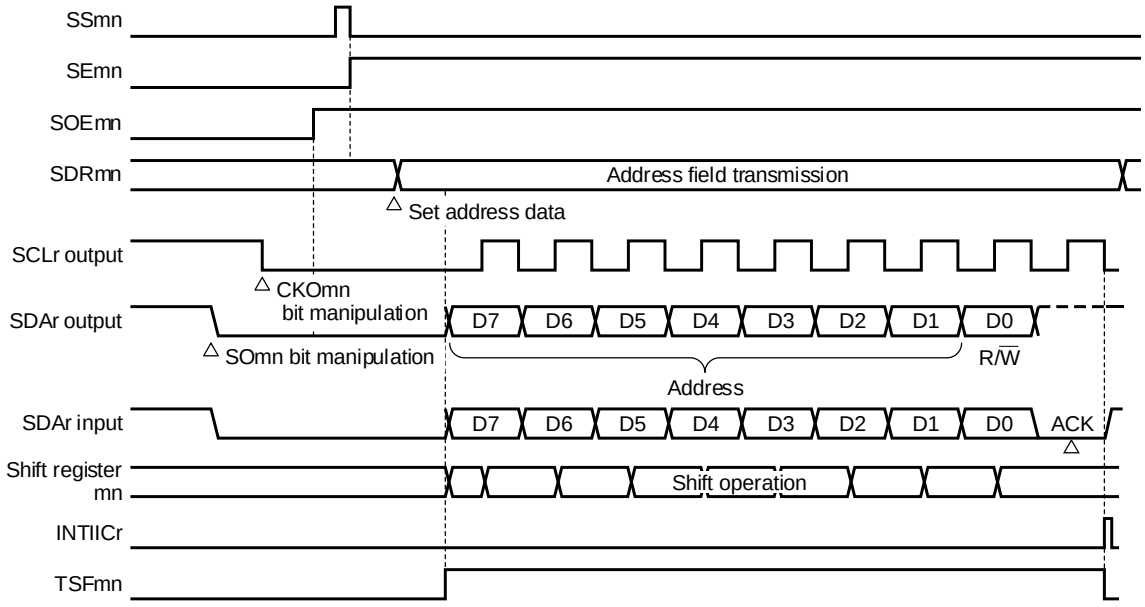
Figure 16-131. Initial Setting Procedure for Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

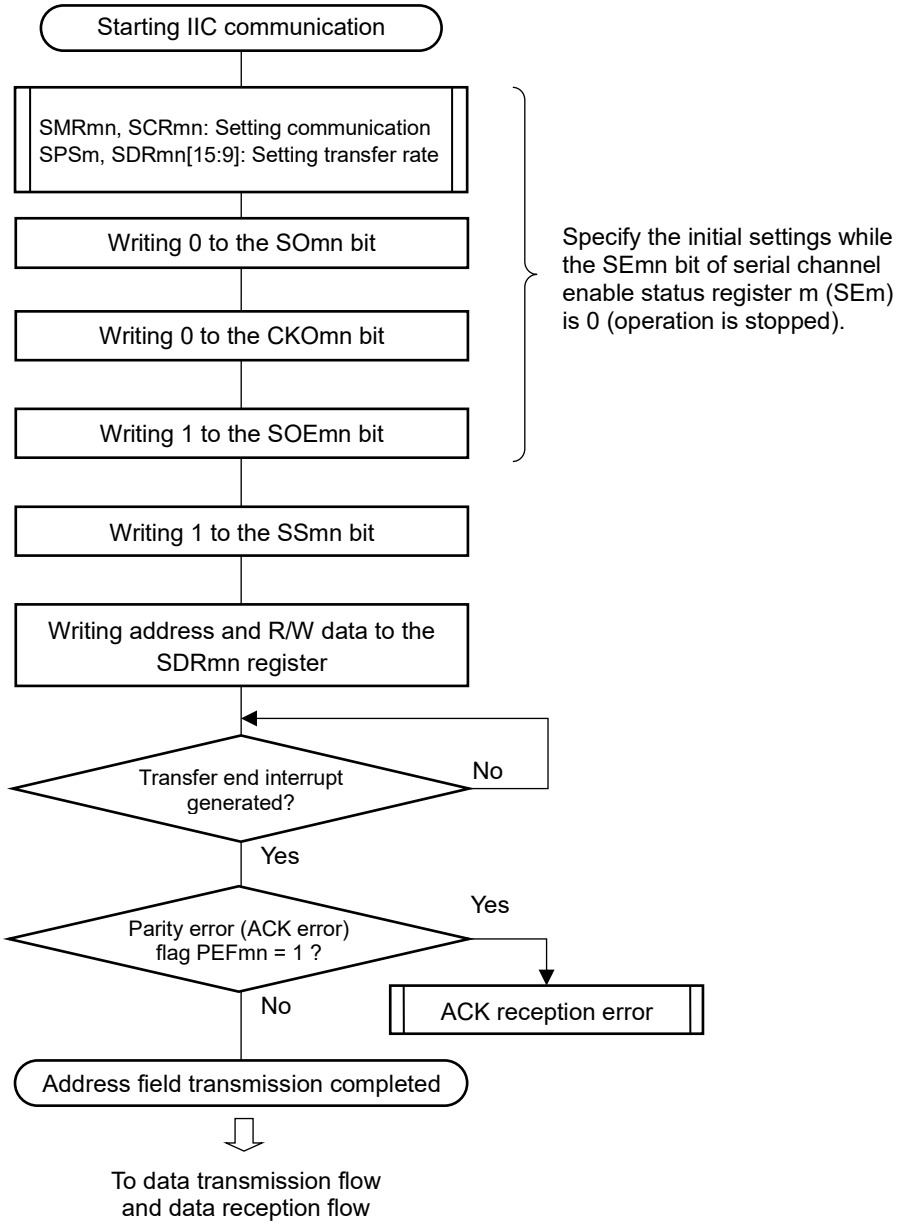
(3) Processing Flow

Figure 16-132. 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)

Figure 16-133. Flowchart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.9.2 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 However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • 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_{DD} tolerance) mode (POMxx = 1) for the port output mode registers (POMx) (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_{DD} tolerance) mode (POMxx = 1) also for the clock 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)

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 0 (operation clock selected by CKSmn bit).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 10B (simplified I^2C mode).
The MDmn0 bit is set to 0 (interrupt source: transfer end).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 10B (transmission only).
DAPmn and CKPmn bits are set to 00B.
PTCmn1 and PTCmn0 bits are set to 00B.
The DIRmn bit is set to 0.
SLCmn1 and SLCmn0 bits are set to 01B (stop bit length: 1 bit).
DLSmn3 to DLSmn0 bits are set to 0111B (data length: 8 bits).

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set baud rate to bits 15 to 9 of the SDRmn register.
Setting these bits are unnecessary because they are set for transmission of an address field.
When operation is in progress (SEmn = 1): Transmit data setting

Note Lower 8 bit in the SDRmn register: SDRrL register.

(d) Serial output register m (SOm)

(e) Serial output enable register m (SOEm)

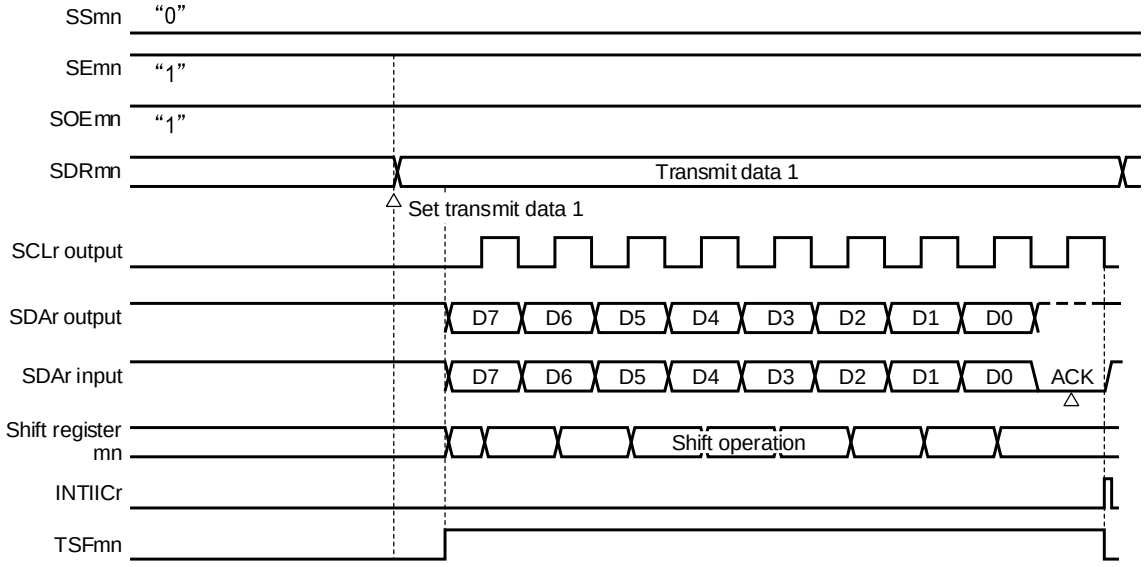
(f) Serial channel start register m (SSm)

Do not manipulate these registers during data transmission/reception.
Because these registers are set when the address field is transmitted, it does not need to be set for this process.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11)

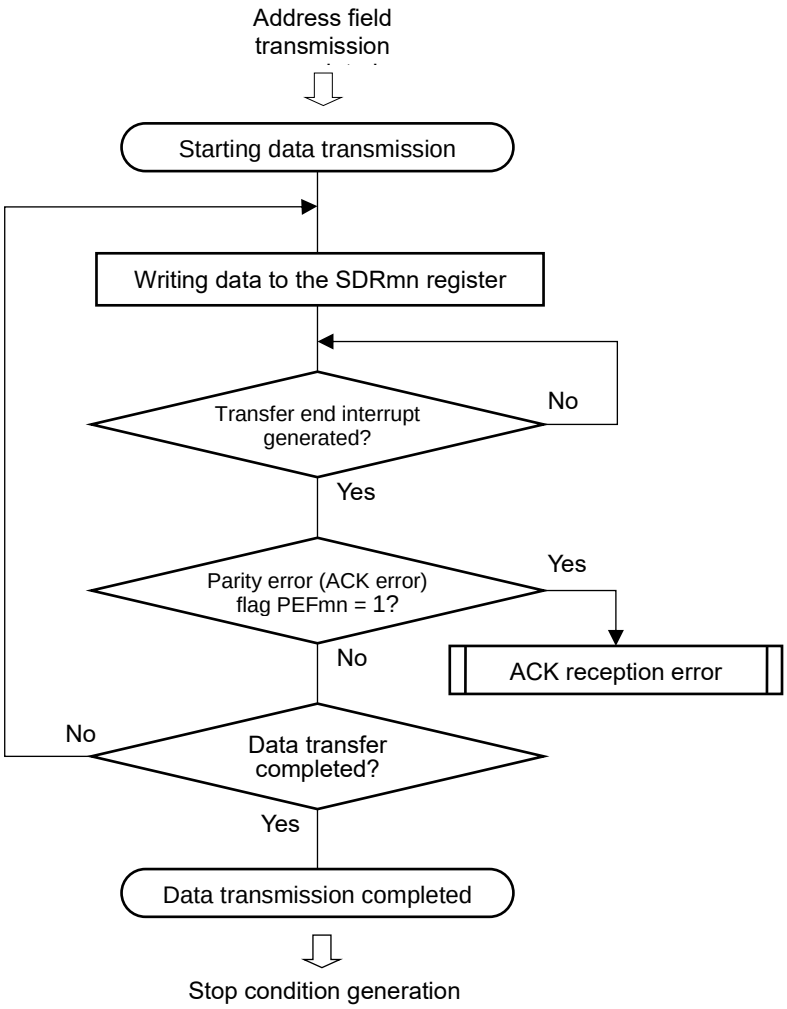
(2) Processing Flow

Figure 16-134. 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)

Figure 16-135. Flowchart of Data Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.9.3 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 However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • 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_{DD} tolerance) mode (POMxx = 1) for the port output mode registers (POMx) (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_{DD} tolerance) mode (POMxx = 1) also for the clock 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)

(1) Register Setting

(a) Serial mode register mn (SMRmn)

The CKSmn bit selects the operation clock (f_{MCK}) of channel n (0: CKm0, 1: CKm1).
The CCSmn bit is set to 0 (operation clock selected by CKSmn bit).
The STSmn bit is set to 0 (software trigger is valid).
The SISmn0 bit is set to 0.
MDmn2 and MDmn1 bits set a value of 10B (simplified I^2C mode).
The MDmn0 bit is set to 0 (interrupt source: transfer end).

(b) Serial communication operation setting register mn (SCRmn)

TXEmn and RXEmn bits are set to 01B (reception only).
Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.
Because this register sets when the address field is transmitted, it does not need to be set for this process.

(c) Serial data register mn (SDRmn) **Note**

When operation is stopped (SEmn bit of the SEm register = 0): Set baud rate to bits 15 to 9 of the SDRmn register.
Setting these bits are unnecessary because they are set for transmission of an address field.
When operation is in progress (SEmn = 1): Dummy transmit data setting (FFH)

Note Lower 8 bit in the SDRmn register: SDRrL register.

(d) Serial output register m (SOm)

(e) Serial output enable register m (SOEm)

(f) Serial channel start register m (SSm)

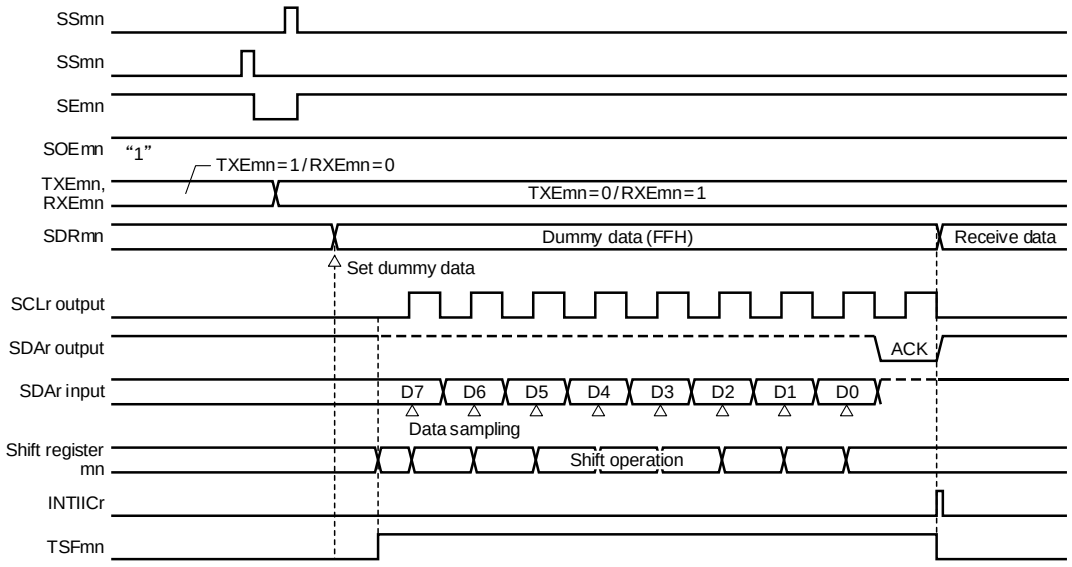
Do not manipulate these registers during data transmission/reception.
Because these registers are set when the address field is transmitted, it does not need to be set for this process.

Remarks m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11)

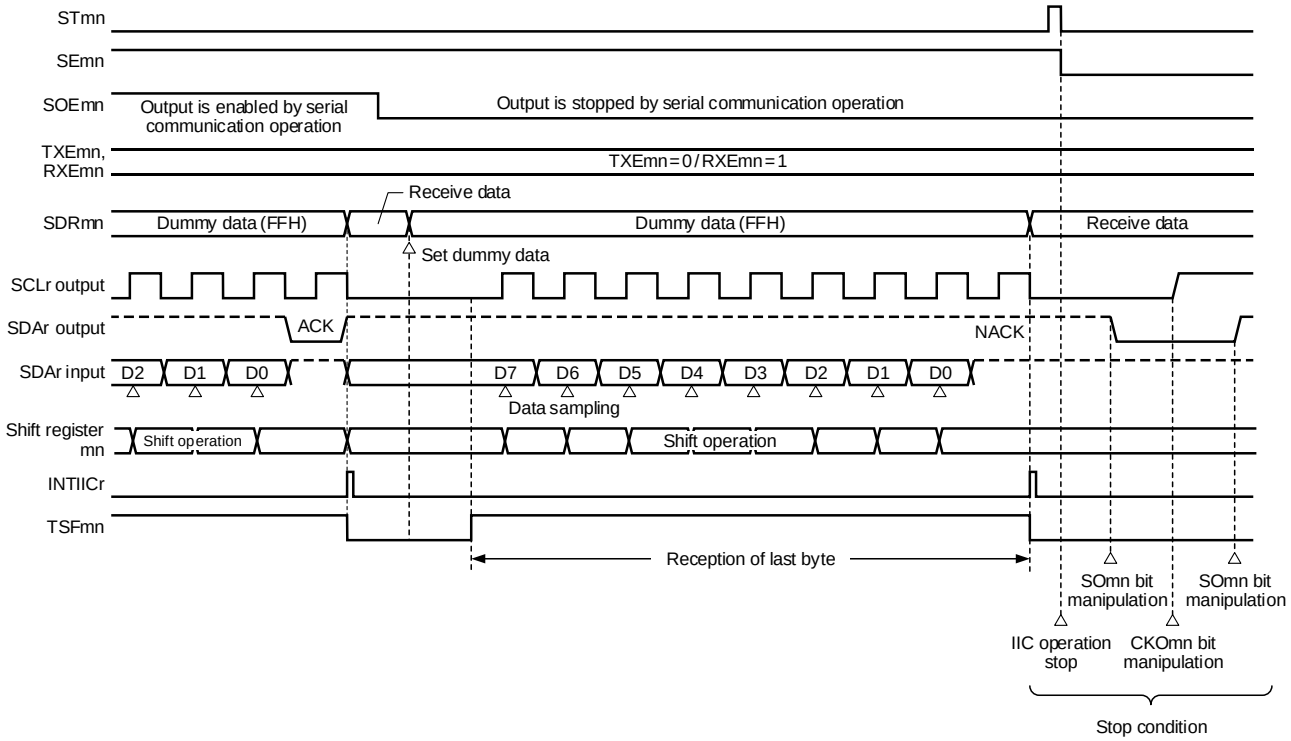
(2) Processing Flow

Figure 16-136. Timing Chart of Data Reception

(a) When starting data reception

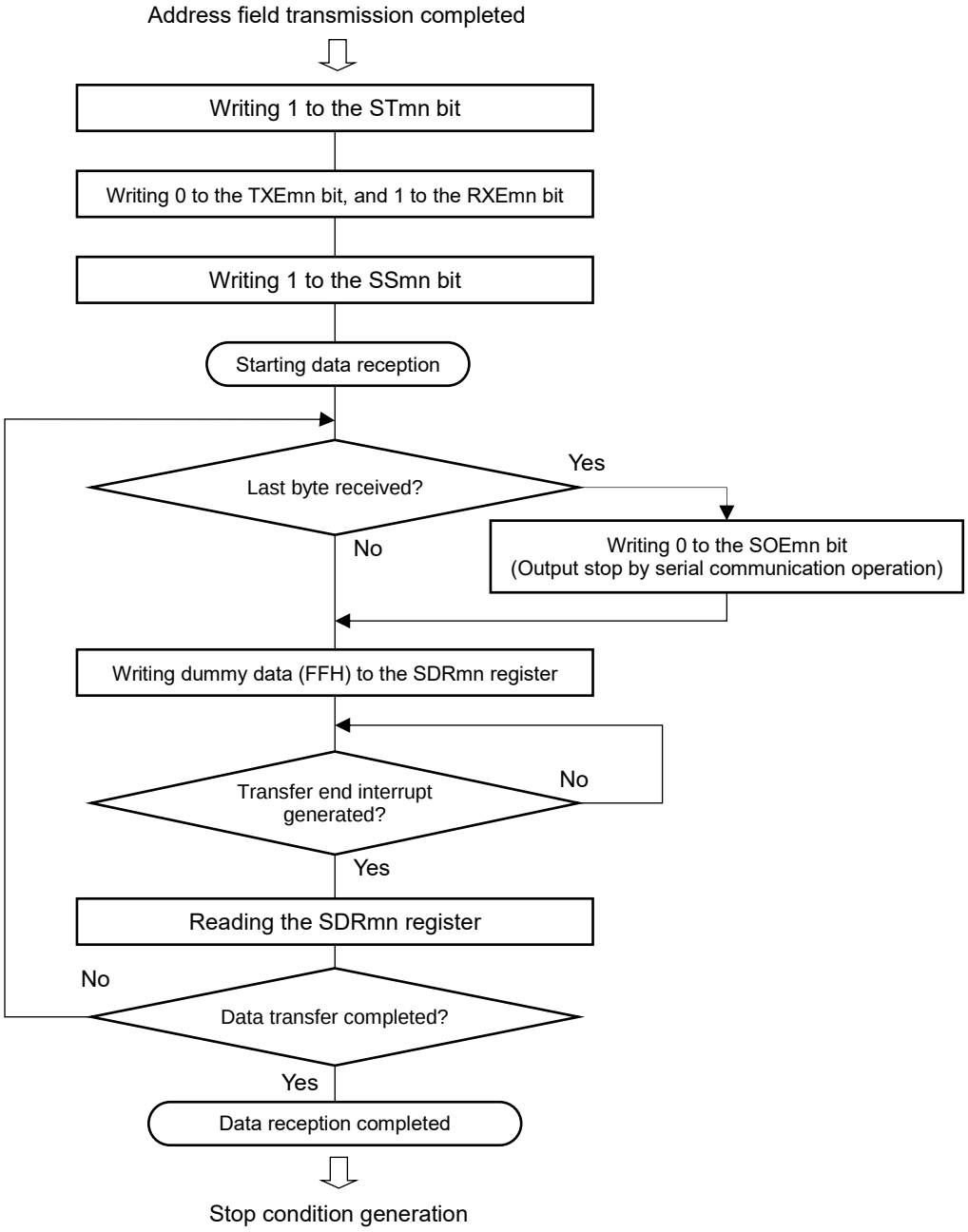


(b) When receiving last data



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01, 10, 11)

Figure 16-137. 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.

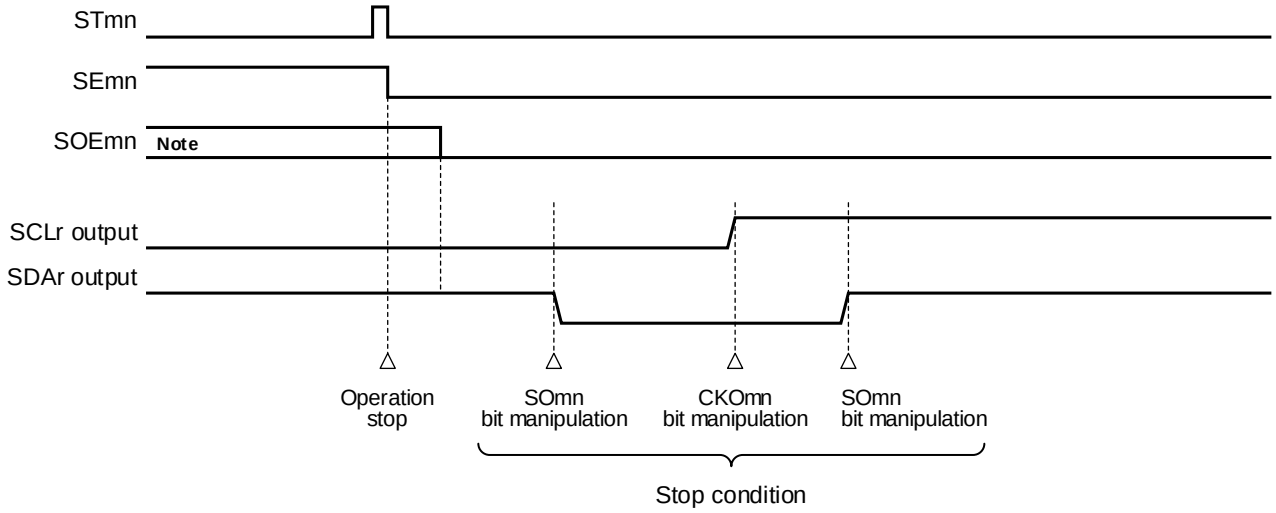
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.9.4 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

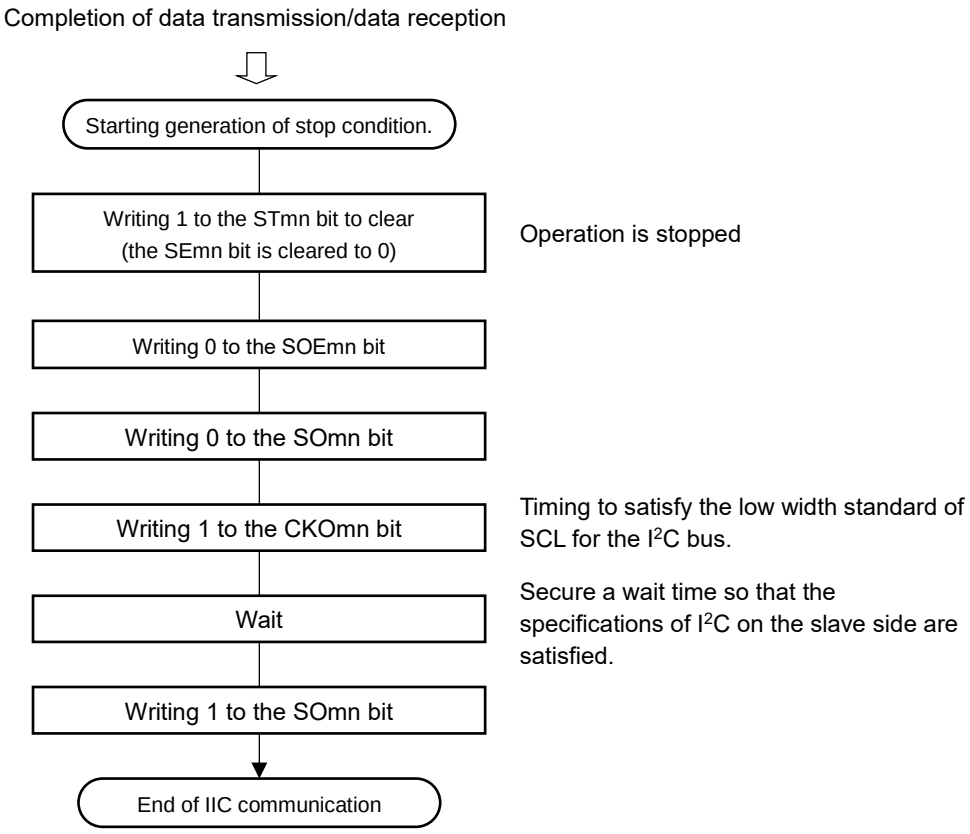
Figure 16-138. Timing Chart of Stop Condition Generation



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)

Figure 16-139. Flowchart of Stop Condition Generation



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.9.5 Calculating Transfer Rate

The transfer rate for simplified I^2C (IIC00, IIC01, IIC10, IIC11) 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)

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 16-6. 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	f_{CLK} = 40 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	32 MHz	40 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	16 MHz	20 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	8 MHz	10 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	4 MHz	5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	2 MHz	2.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	1 MHz	1.25 MHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	500 kHz	625 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	250 kHz	312.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	125 kHz	156.25 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz	78.125 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz	39.0625 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz	19.53125 kHz
1	0	0	0	0	X	X	X	X	f_{CLK}	32 MHz	40 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	16 MHz	20 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	8 MHz	10 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	4 MHz	5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	2 MHz	2.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	1 MHz	1.25 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	500 kHz	625 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	250 kHz	312.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	125 kHz	156.25 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	62.5 kHz	78.125 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	31.25 kHz	39.0625 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	15.63 kHz	19.53125 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)

Here is an example of setting an I^2C transfer rate where $f_{MCK} = f_{CLK}$ = 32 MHz or 40 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%

I^2C Transfer Mode (Desired Transfer Rate)	f_{CLK} = 40 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}/2$	99	100 kHz	0.0%
400 kHz	f_{CLK}	49	400 kHz	0.0%

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 1)

16.9.6 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 16-140** and **Figure 16-141**.

Figure 16-140. Processing Procedure in Case of Overrun Error

Software Manipulation		Hardware Status	Remarks
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 16-141. Processing Procedure in Case of Parity Error (ACK error) in Simplified I²C Mode

Software Manipulation		Hardware Status	Remarks
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)

16.10 Simplified I²S Communication

Simplified I²S communication operation is realized by using serial array unit (CSI10), channel 2 and 3 of timer array unit 1 (TAU12, TAU13), and a circuit dedicated to this communication.

This product supports only master communication.

- RL78/F25 Products and RL78/F22 48-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C	Simplified I²S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function)		IIC01	–
1	0	CSI10 (supporting SPI function) Note 1	UART1	IIC10	Supported
	1	CSI11 (supporting SPI function)		IIC11	–

- RL78/F22 32-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C	Simplified I²S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function)		IIC01	–
1	0	CSI10 (supporting SPI function) Note 1	UART1	IIC10	Supported
	1	–		–	–

- RL78/F22 24-pin Products

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C	Simplified I²S
0	0	CSI00 (supporting SPI function)	UART0 (supporting LIN-bus)	IIC00	–
	1	CSI01 (supporting SPI function) Note 2		IIC01	–
1	0	–	UART1	–	Supported
	1	–		–	–

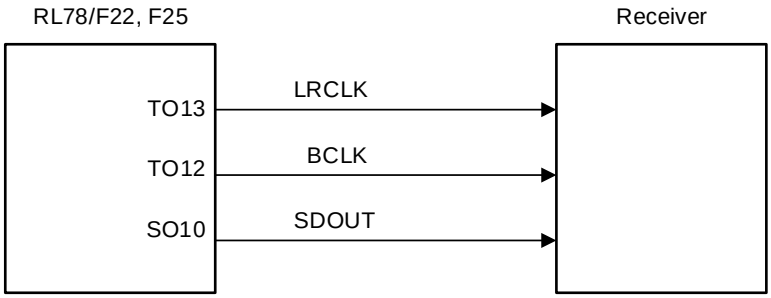
Notes 1. 48-pin and 32-pin products do not have SSI10 pin.

2. 24-pin products do not have SSI01 pin.

16.10.1 Overview

Configuration for the simplified I²S communication is shown in **Figure 16-142**.

Figure 16-142. Configuration for Simplified I²S Communication



Signal list for the simplified I²S is shown below.

Table 16-7. Signal List for Simplified I²S Communication

Pin Name	Function
LRCLK	• LR clock (word select). • Signal to distinguish L-channel/R-channel, matches the sampling frequency.
BCLK	I²S bit clock
SDOUT	Audio data

Table 16-8 shows the specifications of simplified I²S communication.

Table 16-8. Communication Format of Simplified I²S Communication

Item	Description
Audio data format	PCM
I²S function	Master communication only (Slave communication is not supported)
BCLK frequency	32 fs, 48 fs, 64 fs (fs: Sampling frequency)
Sampling frequency	8 kHz to 96 kHz
Transfer data length	Up to 32 bits
Terms of use	2.7 V ≤ V_{DD} ≤ 5.5 V, 8 MHz ≤ f_{CLK} ≤ 40 MHz **Note**

Note The simplified I²S communication function uses a combination of timer array unit (TO12, TO13) and serial array unit (CSI10). Check the communication specifications of the connection destination and set the BCLK frequency for simplified I²S communication.

16.10.2 Master Transmission

Simplified I²S communication operation is realized by using serial array unit (CSI10), channel 2 and 3 of timer array unit 1 (TAU12, TAU13), and a circuit dedicated to this communication. See the corresponding chapters for the registers related to each function.

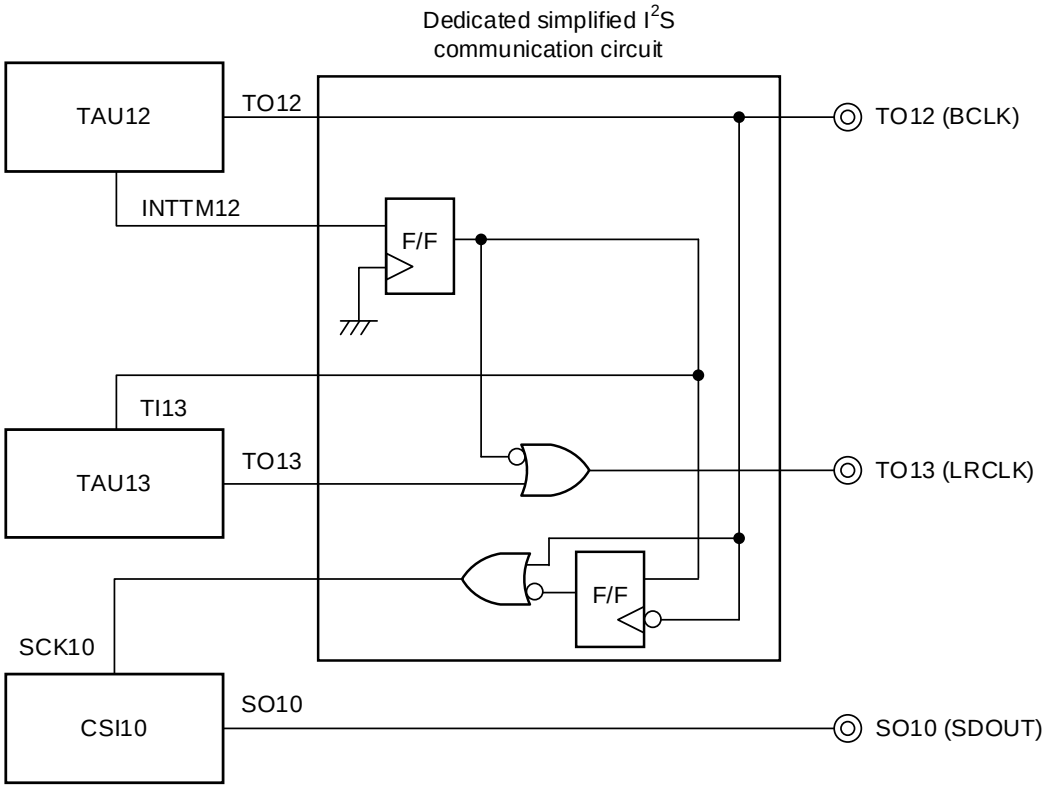
- Channel 2 of timer array unit 1 (TAU12): Interval timer mode.
- Channel 3 of timer array unit 1 (TAU13): External event counter mode.
- 3-wire serial I/O (CSI10): Slave transmission mode.

Signals LRCLK and BCLK outputs use timer array unit 1. BCLK signal is generated by TO12 signal of timer array unit 1. TO12 signal is input to SCK10 of the CSI10 function via a dedicated simplified I²S communication circuit.

Channel 3 of the timer array unit 1 (TAU13) counts the TO12 signal in event counter mode and generates the TO13 signal. TO13 signal is output as LRCLK signal.

Figure 16-143 shows the block diagram of simplified I²S communication function.

Figure 16-143. Block Diagram of Simplified I²S Communication Function



16.10.3 Operation of Simplified I²S Communication

(1) Register Setting

(a) Simplified I²S function control register (I2SER)
The I2SEN bit is set to 1 (enables simplified I²S communication function).

(b) Set the TO121, TO120 bits of PWMDLY2 register to 01B. (PWM output of TO12 is set to be delayed by 1 cycle of f_{CLK}.)

(c) Set channel 2 of timer array unit 1 (TAU12) to interval timer mode. (See **Chapter 6**.)

(d) Set channel 3 of timer array unit 1 (TAU13) to event counter mode. (See **Chapter 6**.)

(e) Set channel 0 of serial array unit 1 to 3-wire serial I/O communication (CSI10) slave transmission mode. (See **Section 16.5**)

(2) Communication Procedure

<1> Set channel 2 of timer array unit 1 (TAU12) to interval timer mode and channel 3 of timer array unit 1 (TAU13) to event counter mode. Also, set channel 0 of serial array unit 1 to CSI10 slave transmission mode.

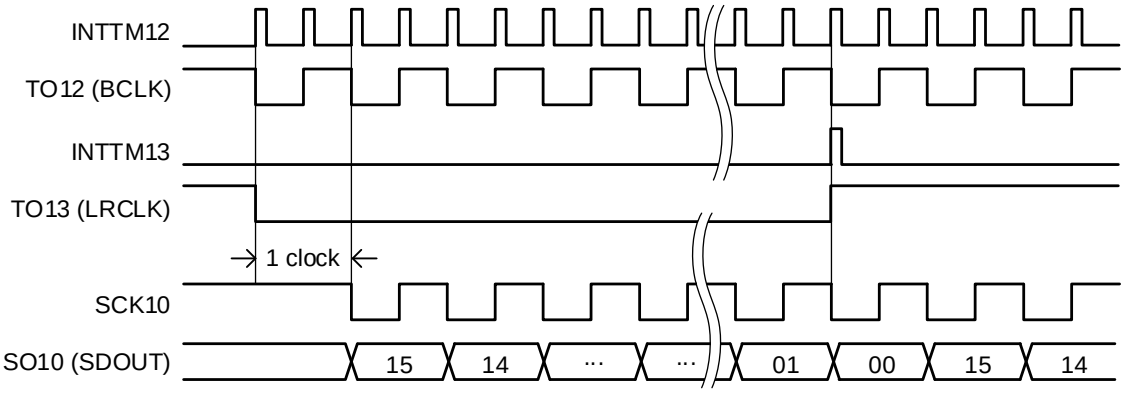
<2> Set the BCLK frequency to the TAU12 and start TAU12 and TAU13 counting.

<3> INTTM12 signal of TAU12 enters dedicated simplified I²S communication circuit, and TO13 (LRCLK) outputs low level.

<4> Internal signal (TO12) is delayed by 1 clock by dedicated simplified I²S communication circuit and input to SCK10 signal of CSI10 function.

<5> CSI10 function outputs transmission data from SO10 pin in synchronization with SCK10 signal.

Figure 16-144. Timing Chart of Simplified I²S Communication



Remarks 1. INTTM12: Interrupt request signal for channel 2 of timer array unit 1 (TAU12)

2. INTTM13: Interrupt request signal for channel 3 of timer array unit 1 (TAU13)

CHAPTER 17 SERIAL INTERFACE IICA

17.1 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 generate “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 17-1 shows a block diagram of serial interface IICA.

Figure 17-1. Block Diagram of Serial Interface IICA

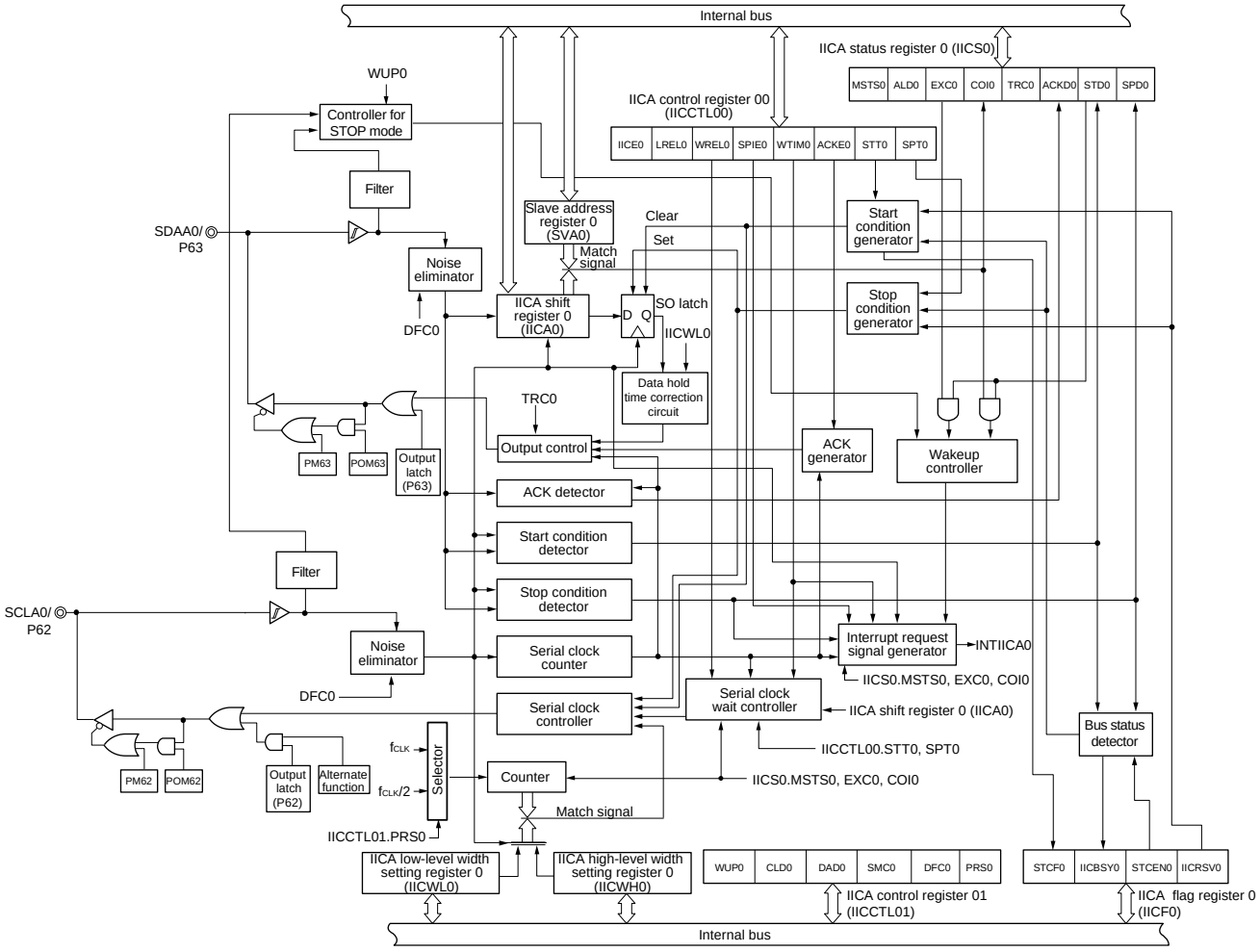
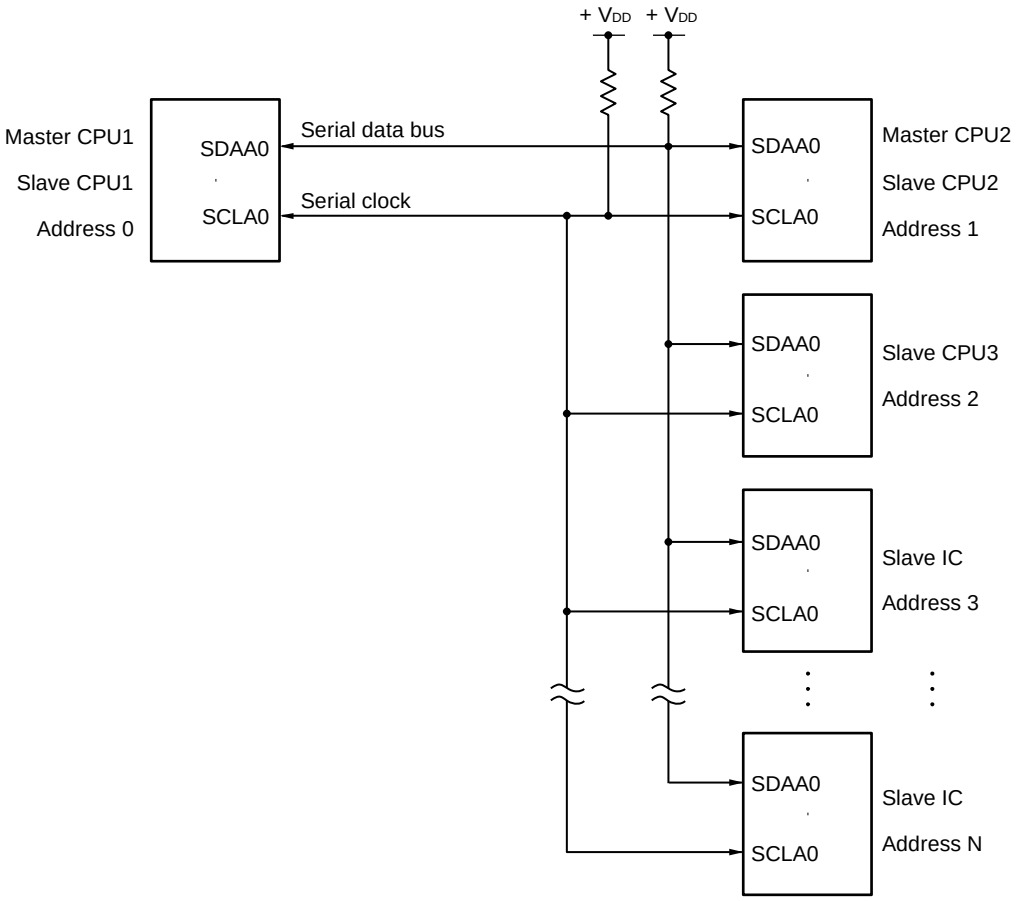


Figure 17-2 shows a serial bus configuration example.

Figure 17-2. Serial Bus Configuration Example Using I²C Bus



17.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 17-1 Configuration of Serial Interface IICA

Item	Configuration
Shift register	IICA shift register 0 (IICA0)
Slave address register	Slave address register 0 (SVA0) (For storing the local address when in slave mode.)
Controll register	• IICA control register 00 (IICCTL00) • IICA control register 01 (IICCTL01) • IICA low-level width setting register 0 (IICWL0) • IICA high-level width setting register 0 (IICWH0) • Slave address register 0 (SVA0) • IICA shift register 0 (IICA0) • IICA status register 0 (IICS0) • IICA flag register 0 (IICF0)
Serial clock input/output	SCLA0 pin (N-ch open-drain output / Schmitt input)
Serial data bus	SDAA0 pin (N-ch open-drain output / Schmitt input)
Interrupt request signal	INTIICA0

(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 17-3. 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-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 17-4. 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)

17.3 Registers Controlling Serial Interface IICA

The control registers of the serial interface IICA are shown below.

Table 17-1. Configuration of IICA Control Register

Address	Register Name	Symbol	After Reset	Access Size
F02C0H	Peripheral enable register 0	PER0	00H	1, 8
F0230H	IICA control register 00	IICCTL00	00H	1, 8
F0231H	IICA control register 01	IICCTL01	00H	1, 8
F0232H	IICA low-level width setting register 0	IICWL0	FFH	8
F0233H	IICA high-level width setting register 0	IICWH0	FFH	8
F0234H	Slave address register 0	SVA0	00H	8
FFF50H	IICA shift register 0	IICA0	00H	8
FFF51H	IICA status register 0	IICS0	00H	1, 8
FFF52H	IICA flag register 0	IICF0	00H	1, 8
FFF26H	Port mode register 6	PM6	FFH	1, 8
F0056H	Port output mode register	POM6	00H	1, 8

Remark For information about the IICA shift register 0 (IICA0) and slave address register 0 (SVA0), see the previous section, **17.2 Configuration of Serial Interface IICA**.

17.3.1 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 17-5. Format of Peripheral Enable Register 0 (PER0)

Address: F02C0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

IICA0EN	Control of serial interface IICA0 input clock supply
0	Stops input clock supply. • SFR used by serial interface IICA0 cannot be written. • Serial interface IICA0 is in the reset status.
1	Enables input clock supply. • SFR used by serial interface IICA0 can be read and written.

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 bit 6 to 0.

17.3.2 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 17-6. 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 • 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. Its uses include cases in which a locally irrelevant extension code has been received. The SCLA0 and SDAA0 lines are set to high impedance. The following flags of IICA control register 00 (IICCTL00) and the IICA status register 0 (IICS0) are cleared to 0. • 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. • After a stop condition is detected, restart is in master mode. • 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 • 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 • Reset	• Set by instruction

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

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 17-6. 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 • 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. Master mode: After output of eight clocks, clock output is set to low level and wait is set. 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. Master mode: After output of nine clocks, clock output is set to low level and wait is set. 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 (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 • 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 • 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 17-6. 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): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation function is enabled (IICRSV0 = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSV0 = 1) 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. In the wait state (when master device): 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. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • 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 17-6. 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 • Automatically cleared after stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • 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.

17.3.3 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 17-7. 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 • When ALD0 = 1 (arbitration loss) • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • 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 • When the IICE0 bit changes from 1 to 0 (operation stop) • 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 17-7. 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 • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • 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 • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • 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> • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Cleared by WREL0 = 1 Note (wait cancel) • When the ALD0 bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTS0, EXC0, COI0 = 0) <Master> • When 1 is output to the LSB (transfer direction specification bit) of the first byte <Slave> • When a start condition is detected • When 0 is input to the LSB (transfer direction specification bit) of the first byte.	<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer). <Slave> • 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 17-7. Format of IICA Status Register 0 (IICS0) (3/3)

ACKD0	Detection of acknowledge (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 • At the rising edge of the next byte's first clock • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • 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 • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • 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 • When the WUP0 bit changes from 1 to 0 • When the IICE0 bit changes from 1 to 0 (operation stop) • 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)

17.3.4 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 17-8. 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 • When IICE0 = 0 (operation stop) • 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 • When IICE0 = 0 (operation stop) • Reset	• Detection of start condition • 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 • Detection of start condition • 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 • 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)

17.3.5 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 17-9. 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 17-23 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.

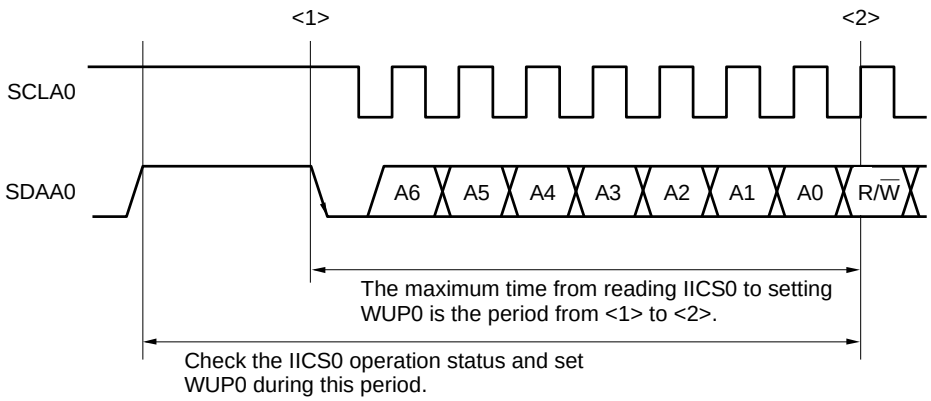


Figure 17-9. Format of IICA Control Register 01 (IICCTL01) (2/2)

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 • When IICE0 = 0 (operation stop) • 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 • When IICE0 = 0 (operation stop) • 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. In fast mode and fast mode plus, the transfer clock does not vary, regardless of the DFC0 bit being set (1) or cleared (0). 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)

17.3.6 IICA Low-level Width Setting Register 0 (IICWL0)

This register is used to set the low-level width (tLOW) 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 I2C 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 **17.4.2 Setting Transfer Clock by Using IICWL0 and IICWH0 registers**.

The data hold time is one-quarter of the time set by the IICWL0 register.

Figure 17-10. 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								

17.3.7 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).

For details about setting the IICWH0 register, see **17.4.2 Setting Transfer Clock by Using IICWL0 and IICWH0 registers**.

Reset signal generation sets this register to FFH.

Figure 17-11. 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 The minimum serial clock cycle is (IICWL0 + 1) + (IICWH0 + 1).

17.3.8 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 17-12. Format of Port Mode Register 6 (PM6) (100-pin products)

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

PM6n	P6n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution PM62 and PM63 are used for the IICA serial interface.

17.3.9 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 17-13. 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.

17.4 I²C Bus Mode Functions

17.4.1 Pin Configuration

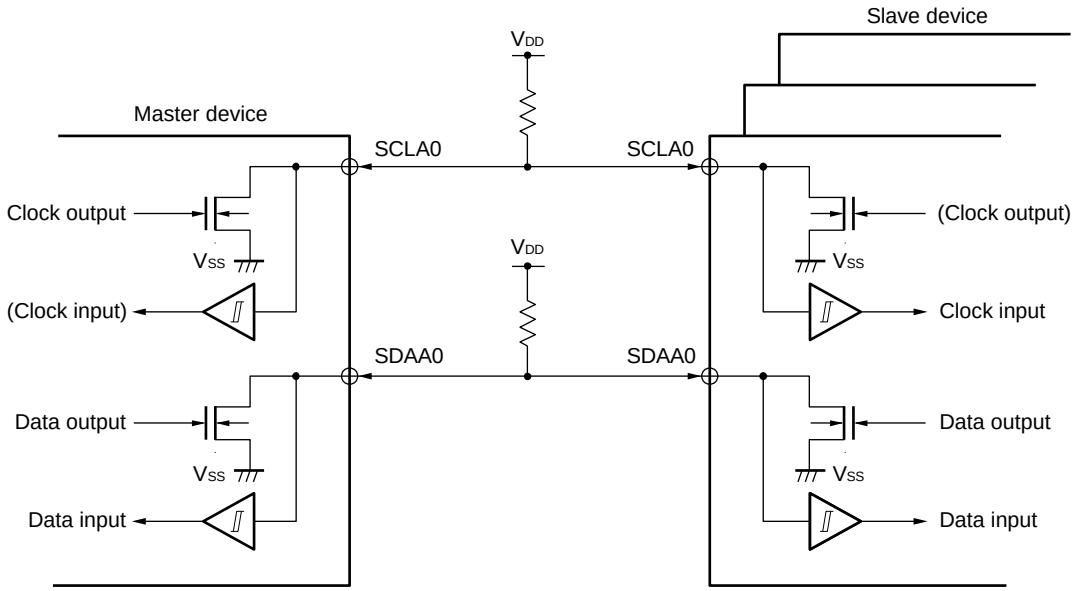
The serial clock pin (SCLA0) and the serial data bus pin (SDAA0) are configured as follows.

(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 17-14. Pin Configuration Diagram



17.4.2 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

$$IICWL0 = \frac{0.52}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWH0 = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the normal mode

$$IICWL0 = \frac{0.47}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWH0 = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the fast mode plus

$$IICWL0 = \frac{0.50}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWH0 = \left(\frac{0.50}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

(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.)

3. The minimum serial clock cycle is (IICWL0 + 1) + (IICWH0 + 1). Determine the values to be set in the IICWL0 and IICWH0 registers by considering the rise time (t_R) and fall time (t_F) of the SDAA0 and SCLA0 signals.

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

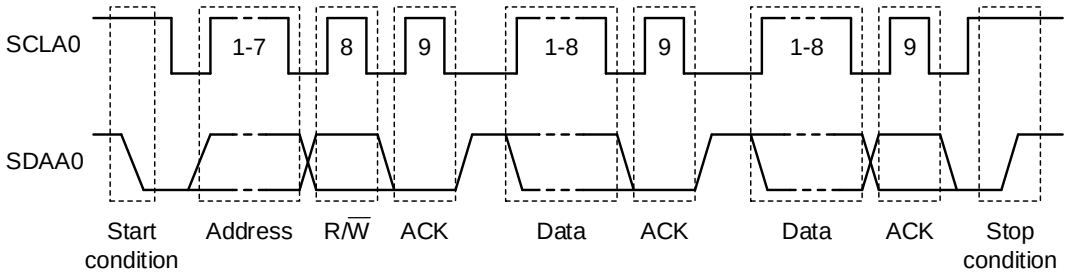
t_R: SDAA0 and SCLA0 signal rising times

f_{MCK}: Frequency of the IICA operation clock

17.5 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 17-15** shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 17-15. I²C Bus Serial Data Transfer Timing



The master device generates the start condition, slave address, and stop condition.

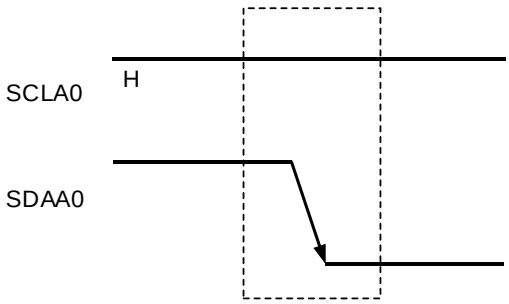
The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

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.

17.5.1 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 17-16. 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).

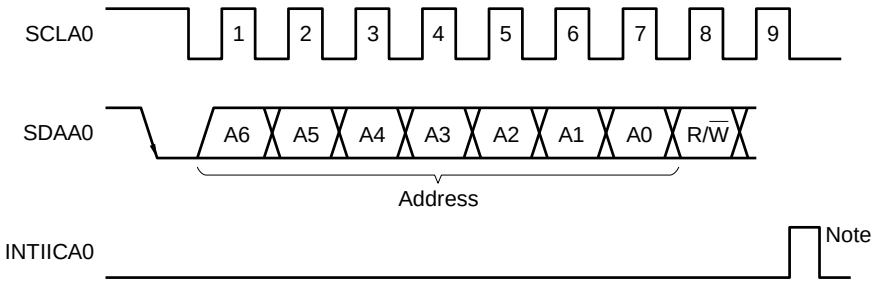
17.5.2 Addresses

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 17-17. 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 **17.5.3 Transfer Direction Specification** are written to the IICA shift register 0 (IICA0). The received addresses are written to the IICA0 register.

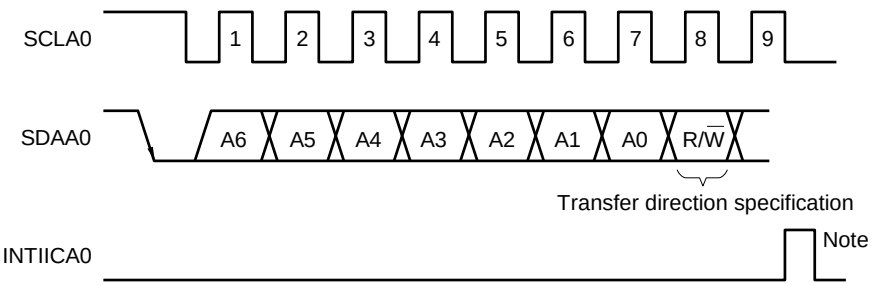
The slave address is assigned to the higher 7 bits of the IICA0 register.

17.5.3 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 17-18. Transfer Direction Specification



Note INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

17.5.4 Acknowledge (ACK)

ACK is used to check the status of serial data at the transmission and reception sides.

The reception side returns ACK each time it has received 8-bit data.

The transmission side usually receives ACK after transmitting 8-bit data. When ACK is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether 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 ACK and instead generates a stop condition. If a slave does not return ACK after receiving data, the master outputs a stop condition or restart condition and stops transmission. If 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.

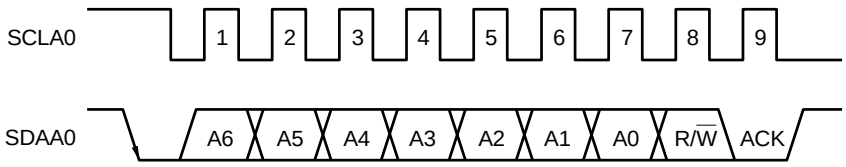
To generate ACK, the reception side makes the SDAA0 line low at the ninth clock (indicating normal reception).

Automatic generation of 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 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 17-19. ACK



When the local address is received, ACK is automatically generated, regardless of the value of the ACKE0 bit. When an address other than that of the local address is received, ACK is not generated (NACK).

When an extension code is received, ACK is generated if the ACKE0 bit is set to 1 in advance.

How ACK is generated when data is received differs as follows depending on the setting of the wait timing.

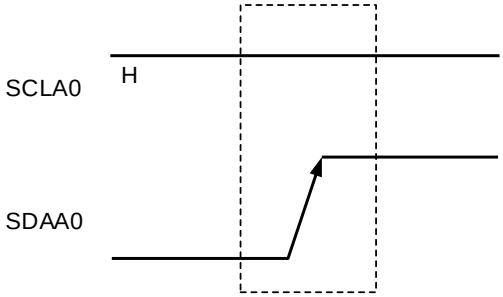
- When 8-clock wait state is selected (bit 3 (WTIM0) of IICCTL00 register = 0):
  By setting the ACKE0 bit to 1 before releasing the wait state, ACK is generated at the falling edge of the eighth clock of the SCLA0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM0) of IICCTL00 register = 1):
  ACK is generated by setting the ACKE0 bit to 1 in advance.

17.5.5 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 17-20. 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.

17.5.6 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 17-21. 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)

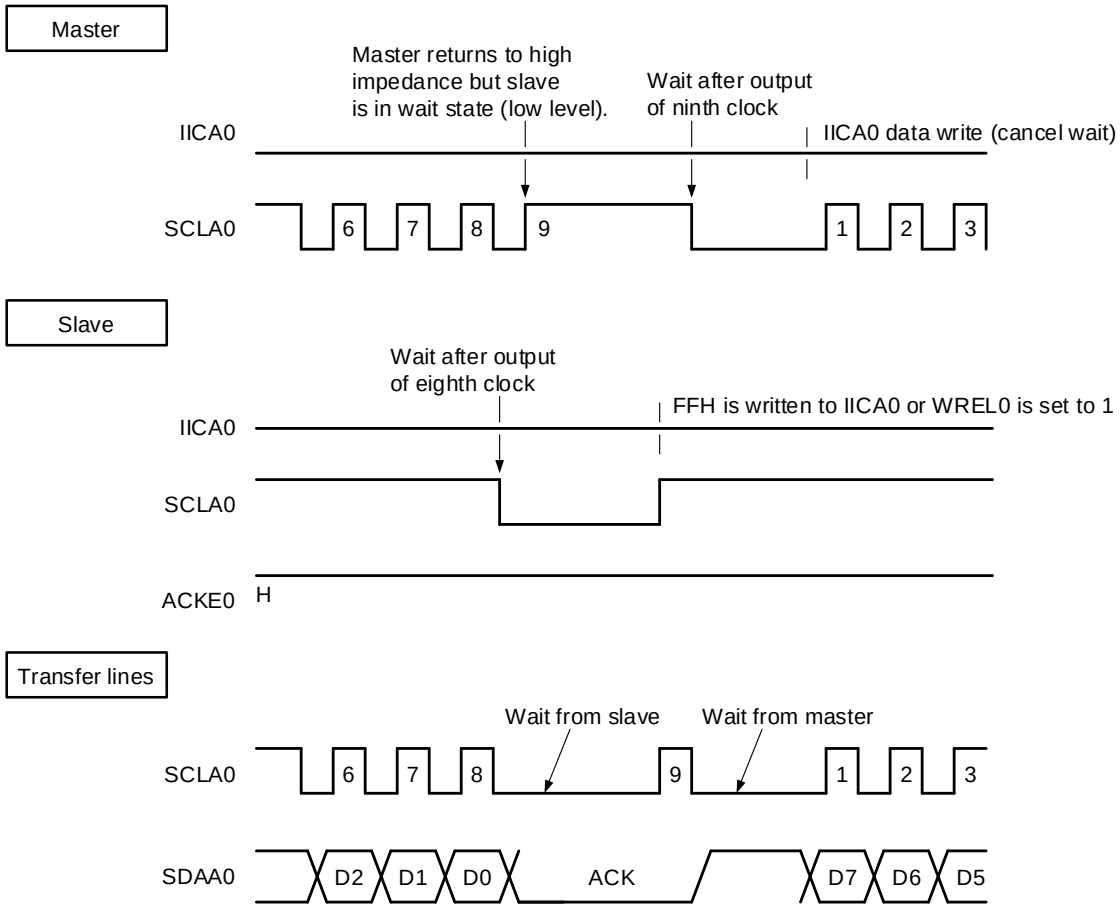
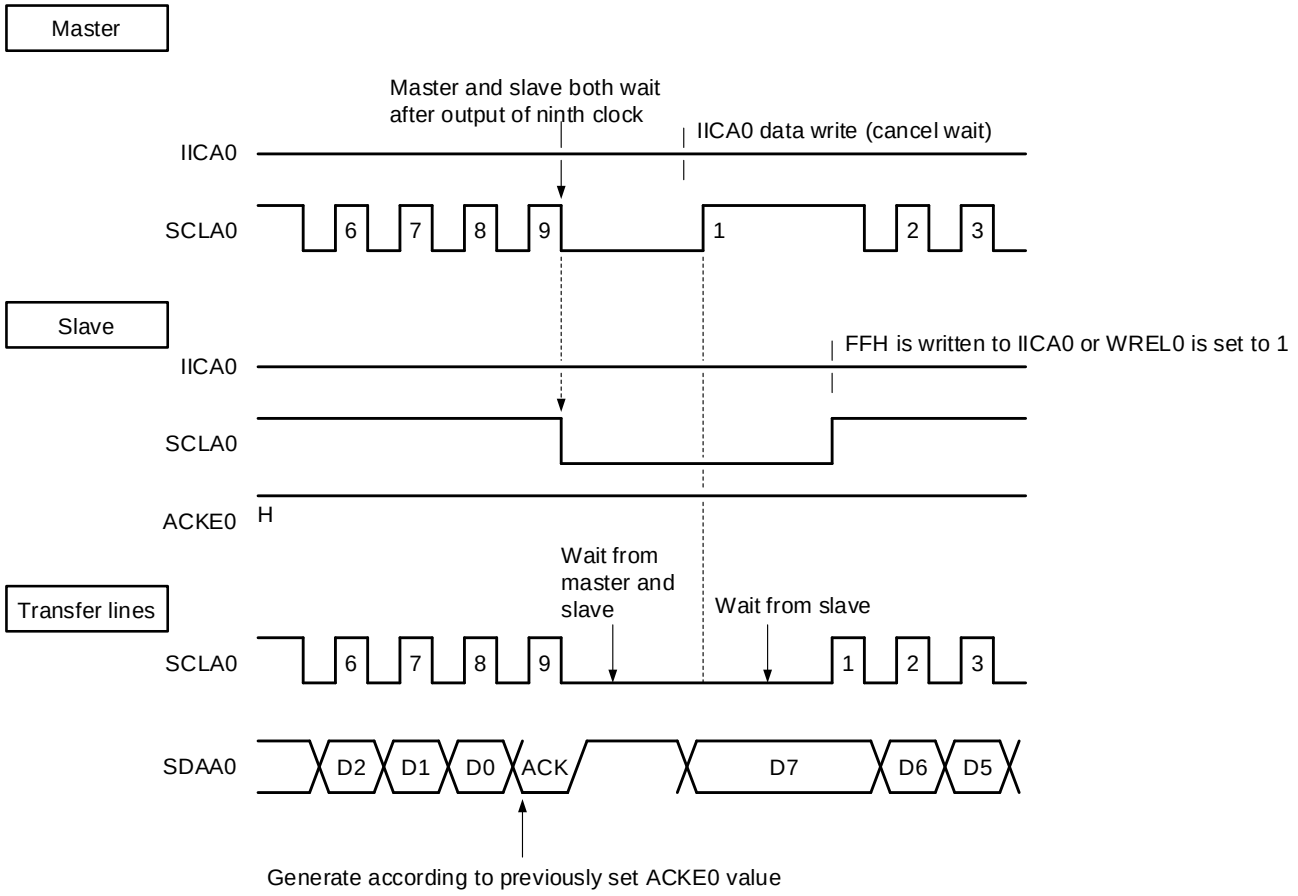


Figure 17-21. Wait (2/2)

(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

17.5.7 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.

17.5.8 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 17-2**.

Table 17-2. 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, 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 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).

17.5.9 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.

17.5.10 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.

17.5.11 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 17-3. 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.

17.5.12 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 **17.5.8 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 17-22. Arbitration Timing Example

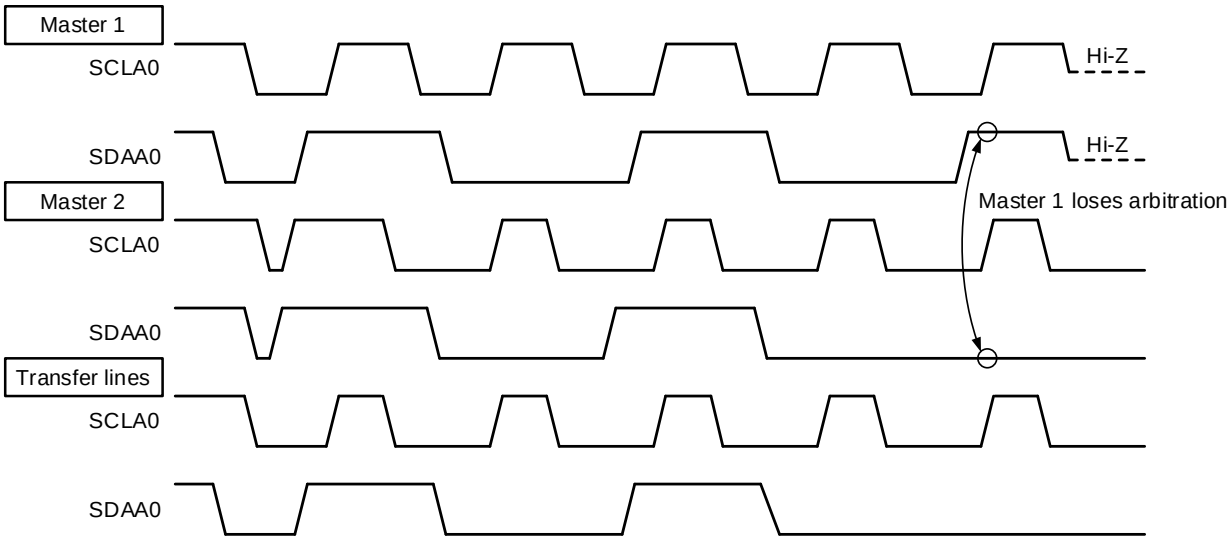


Table 17-4. Status During Arbitration and Interrupt Request Generation Timing

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 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)

17.5.13 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 17-23 shows the flow for setting WUP0 = 1 and **Figure 17-24** shows the flow for setting WUP0 = 0 upon an address match.

Figure 17-23. Flow When Setting WUP0 = 1

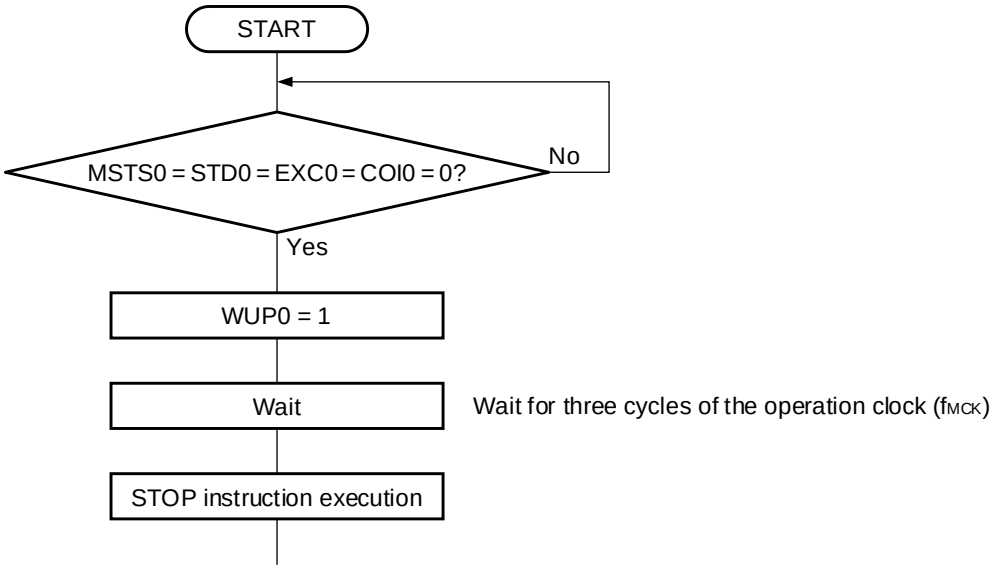
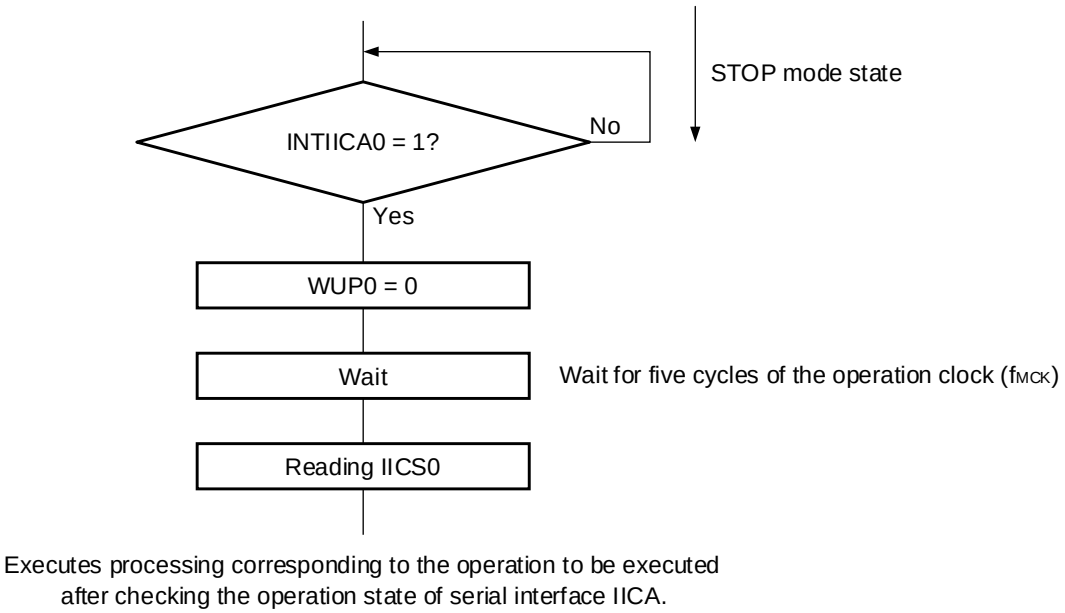


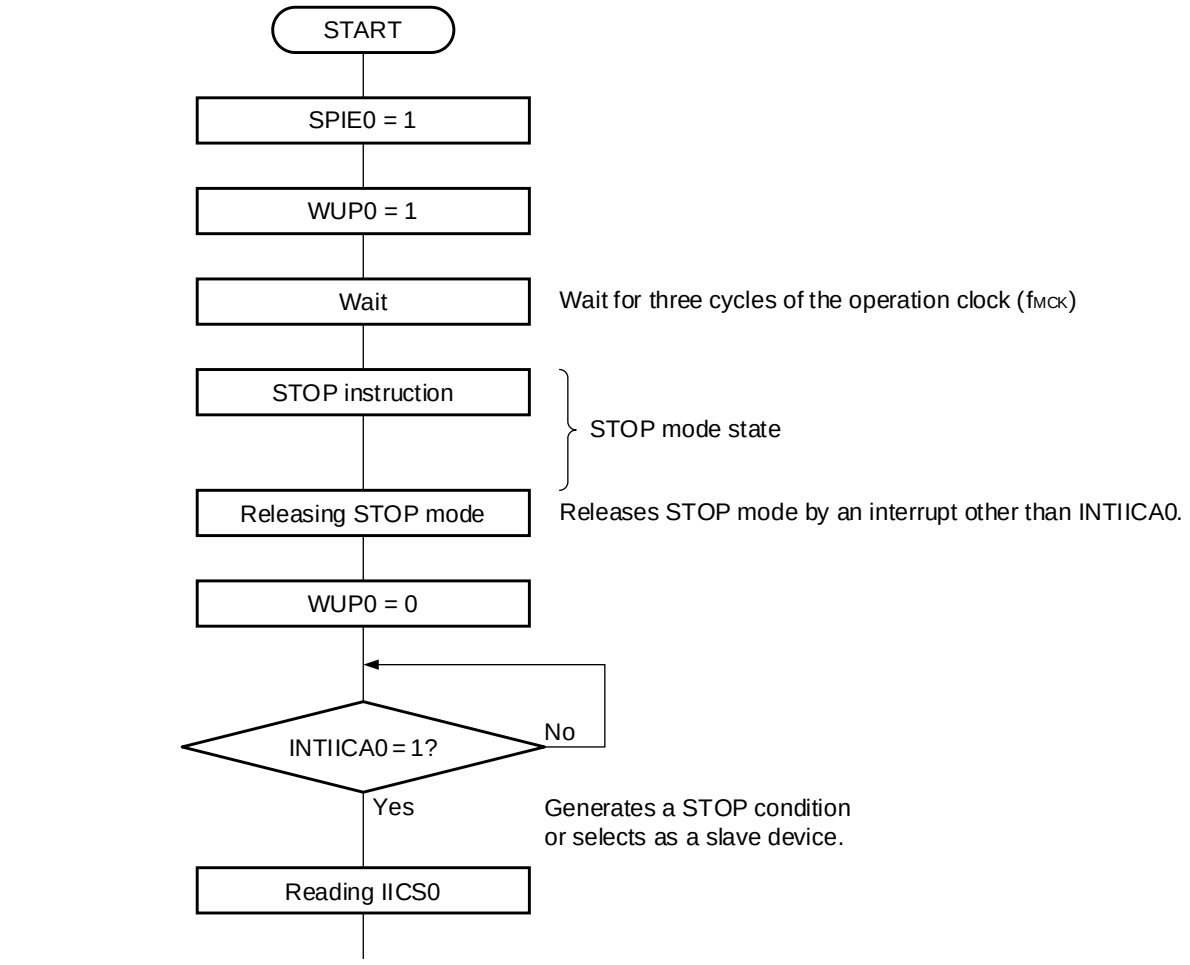
Figure 17-24. Flow When Setting WUP0 = 0 upon Address Match (Including Extension Code Reception)



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 17-25**
- Slave device operation: Same as the flow in **Figure 17-24**. The value of WUP0 must be kept 1 until the INTIICA0 is set to 1.

Figure 17-25. When Operating as Master Device after Releasing STOP Mode other than by INTIICA0



17.5.14 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 (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

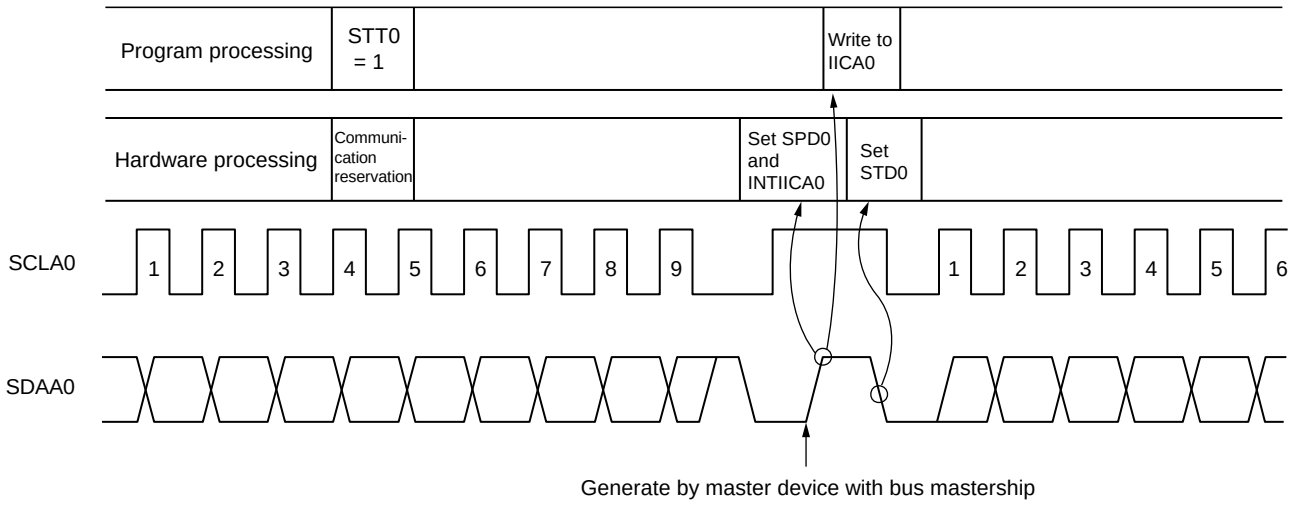
Check whether the communication reservation operates or not by using the MSTS0 bit (bit 7 of the IICA status register 0 (IICS0)) after the STT0 bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

Wait time from setting STT0 = 1 to checking the MSTS0 flag:
(IICWL0 setting value + IICWH0 setting value + 4) / f_{MCK} + t_F × 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 17-26 shows the communication reservation timing.

Figure 17-26. 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 17-27**. 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 17-27. Timing for Accepting Communication Reservations

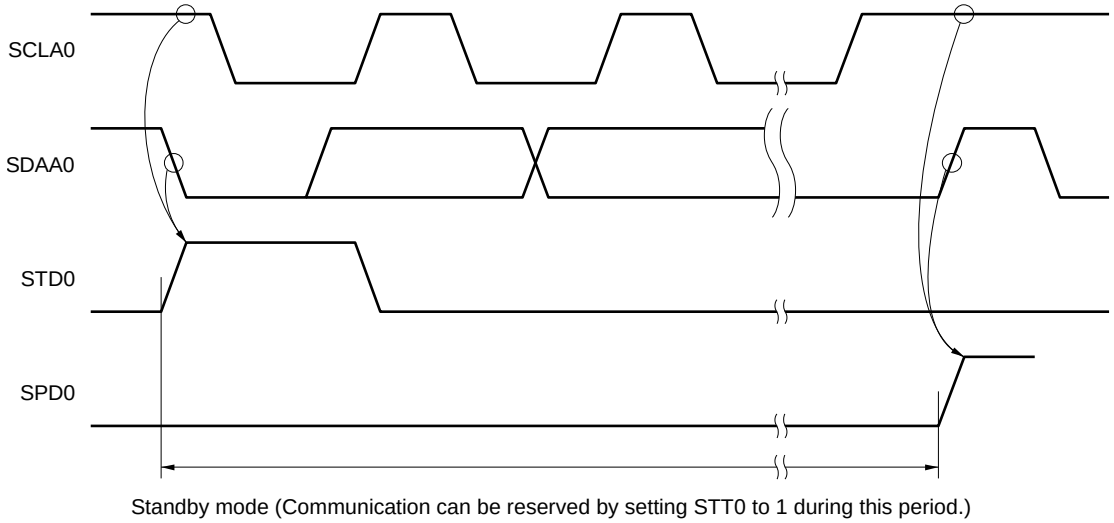
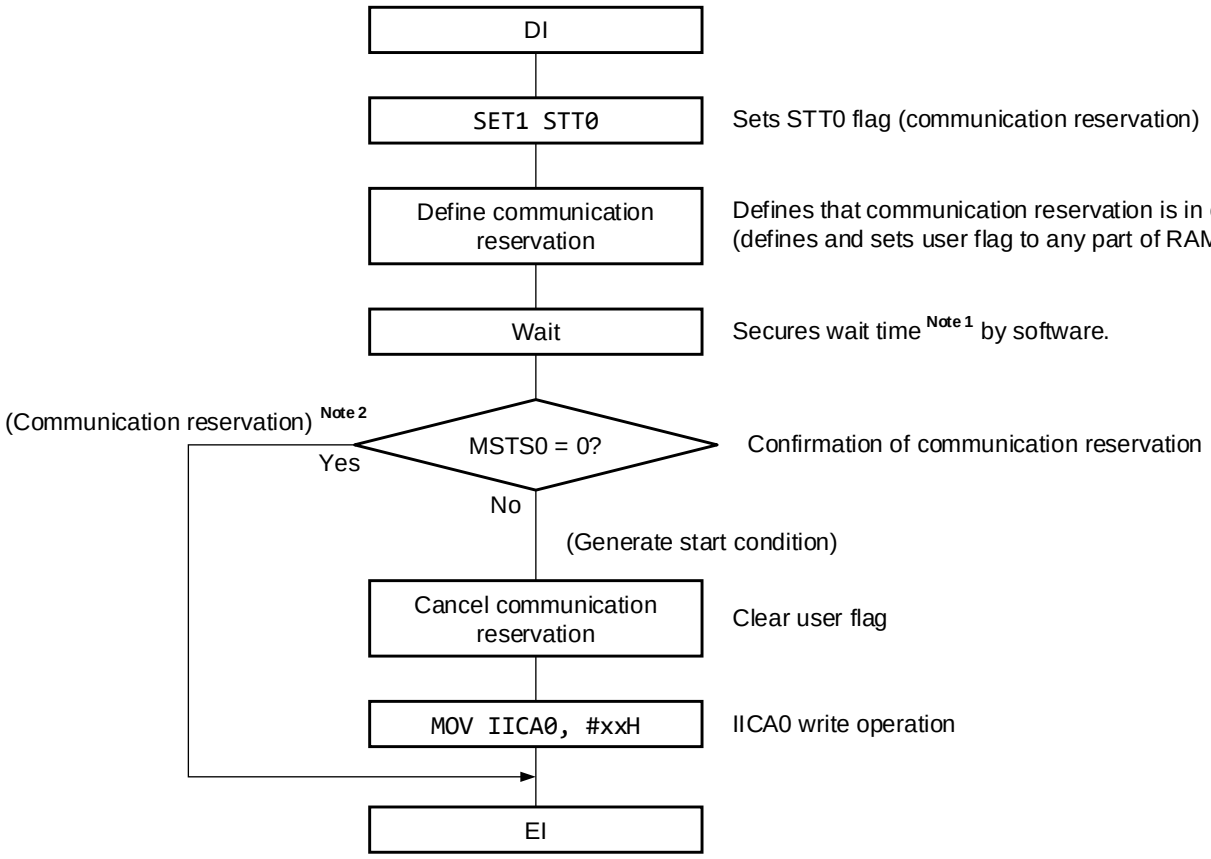


Figure 17-28 shows the communication reservation protocol.

Figure 17-28. Communication Reservation Protocol



Notes 1. The wait time is calculated as follows.
(IICWL0 setting value + IICWH0 setting value + 4) / f_{MCK} + t_F × 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 (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.

17.5.15 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, 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 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.

17.5.16 Communication Operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the RL78/F22 and RL78/F25 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/F22 and RL78/F25 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/F22 and RL78/F25 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

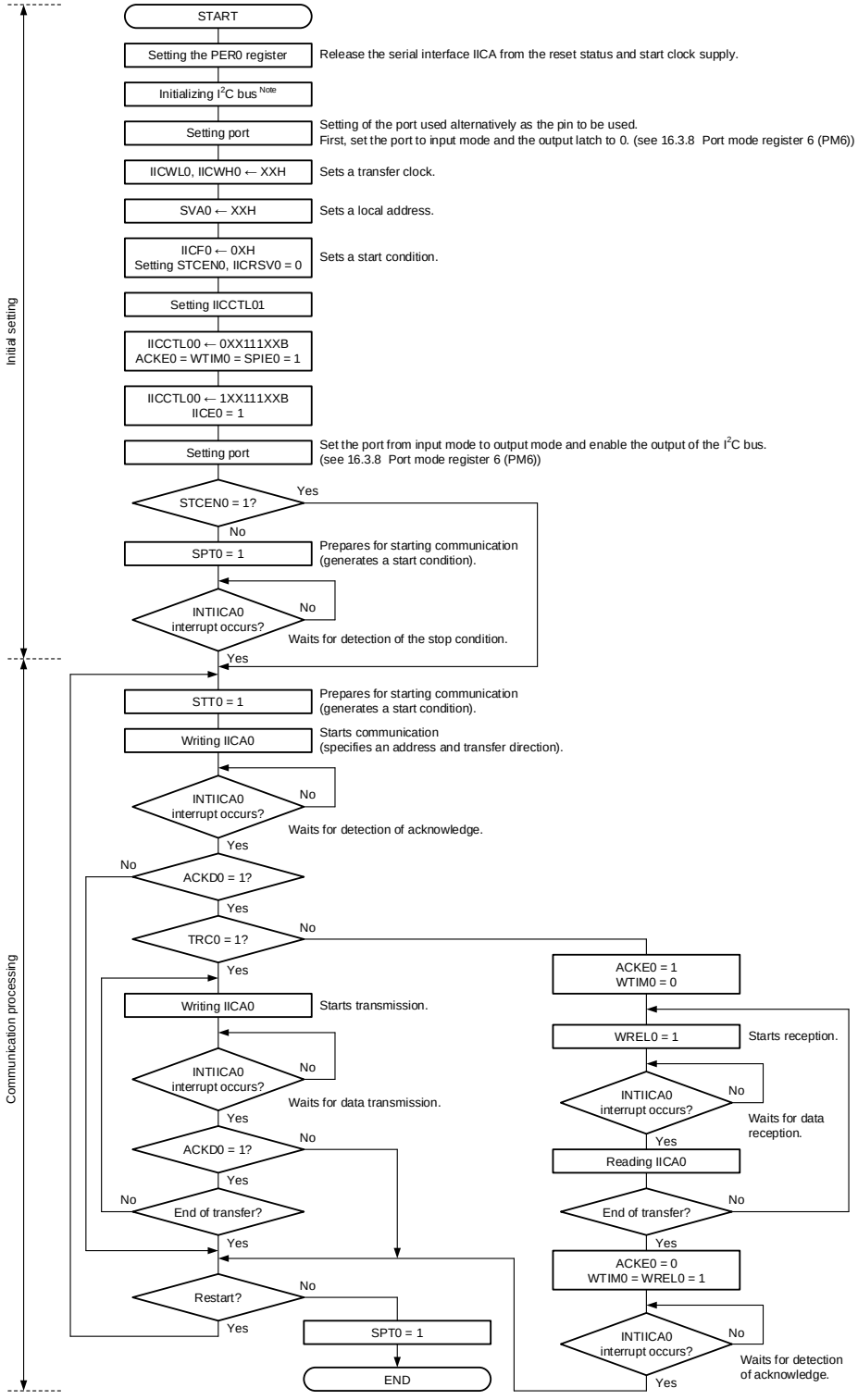
An example of when the RL78/F22 and RL78/F25 are used as the I^2C bus slave is shown below.

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 17-29. Master Operation in Single-Master System

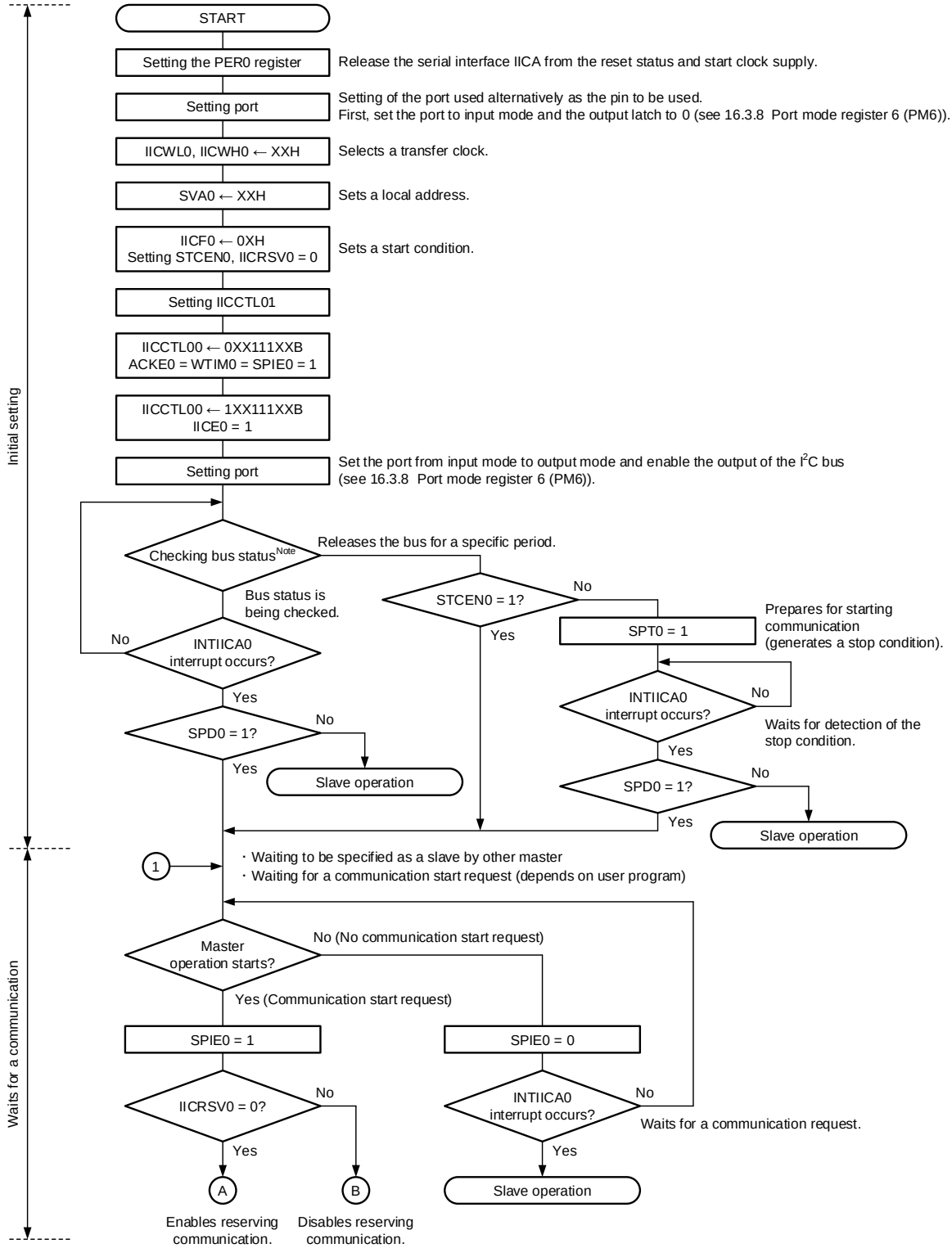


Note Release (SCLA0 and SDAA0 pins = high level) the I²C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAA0 pin, for example, set the SCLA0 pin in the output port mode, and output a clock pulse from the output port until the 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.

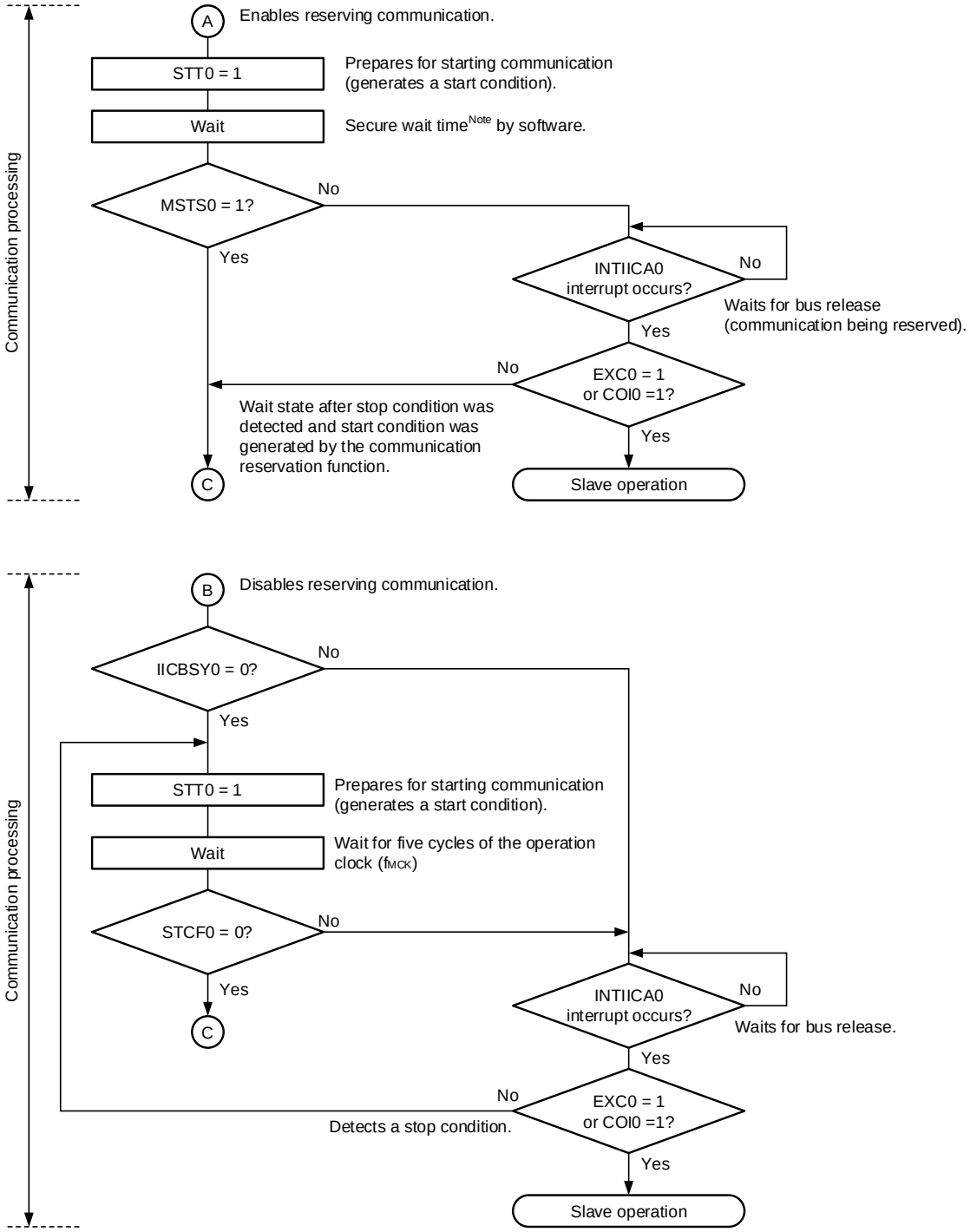
(2) Master operation in multi-master system

Figure 17-30. 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 I^2C bus (SCLA0 and SDAA0 pins = high level) in conformance with the specifications of the product that is communicating.

Figure 17-30. 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 × 2

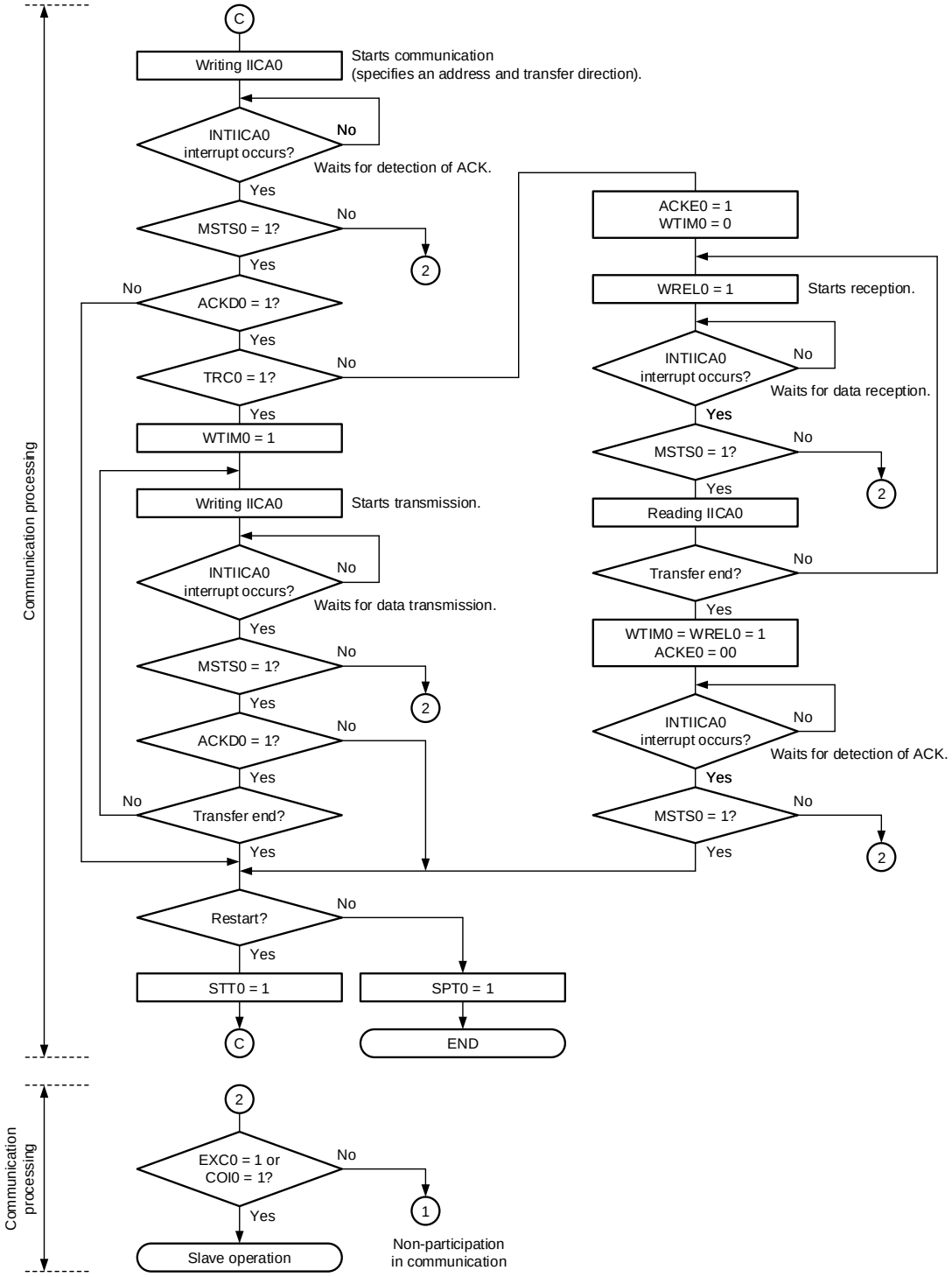
Remark IICWL0: IICA low-level width setting register 0

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 17-30. 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.

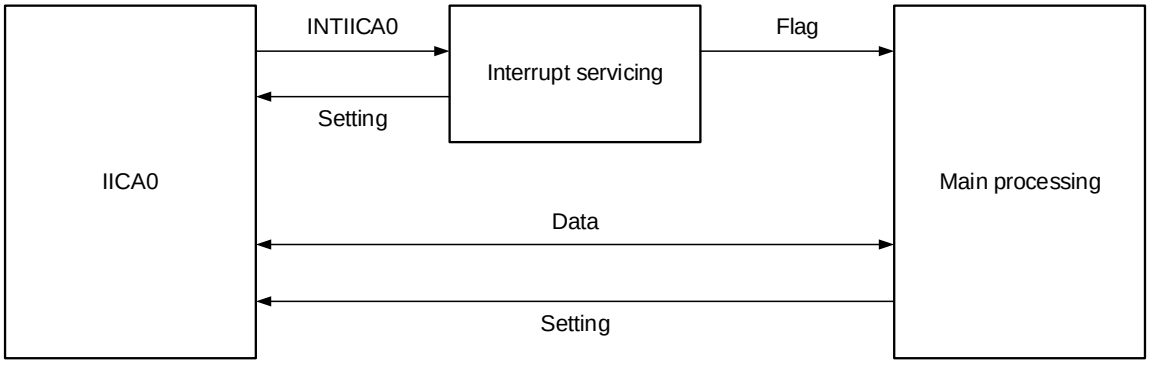
2. To use the device as a master in a multi-master system, read the MSTS0 bit each time interrupt INTIICA0 has occurred to check the arbitration result.

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 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.

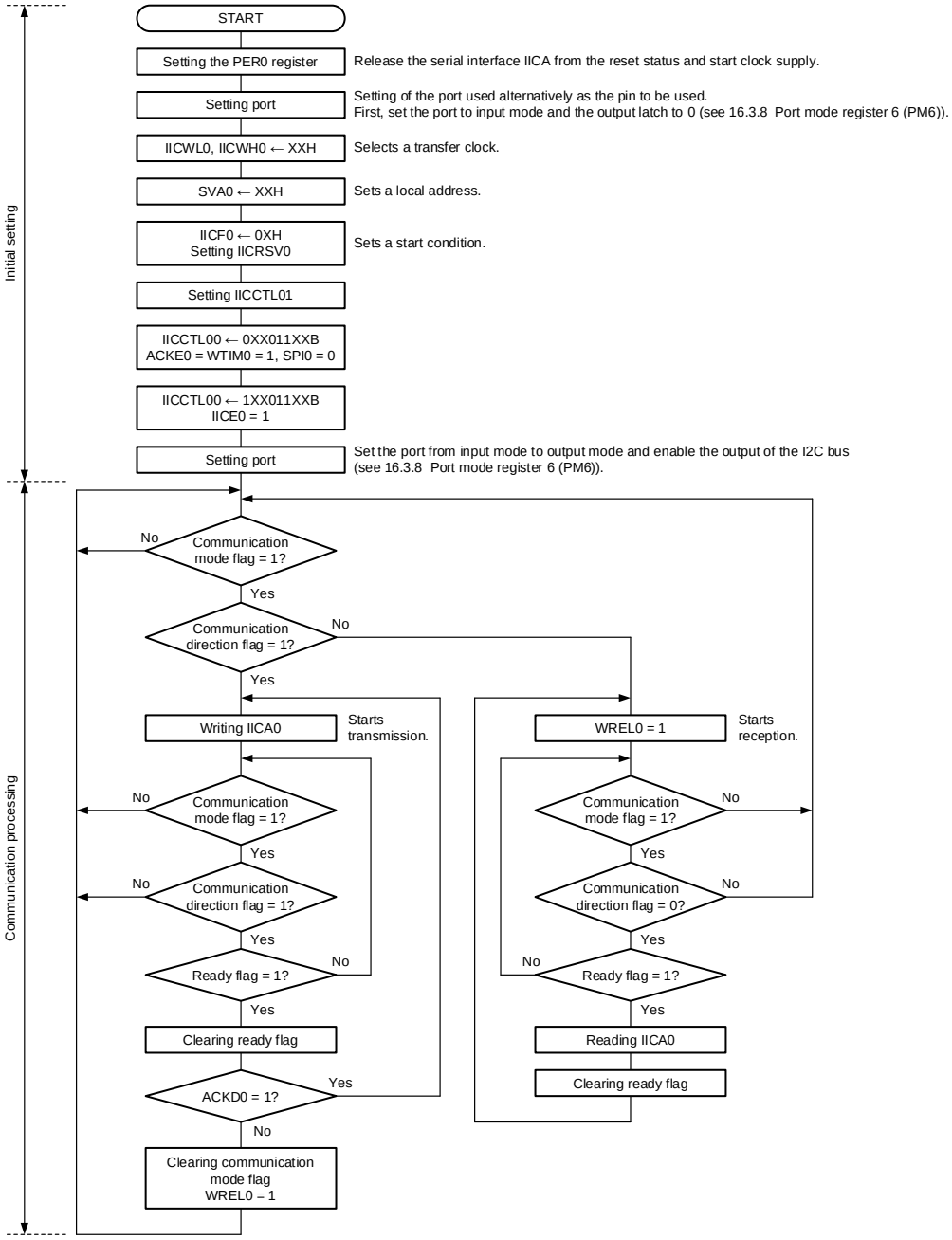
The main processing of the slave operation is explained next.

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 ACK. If ACK is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, 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 17-31. Slave Operation Flowchart (1)



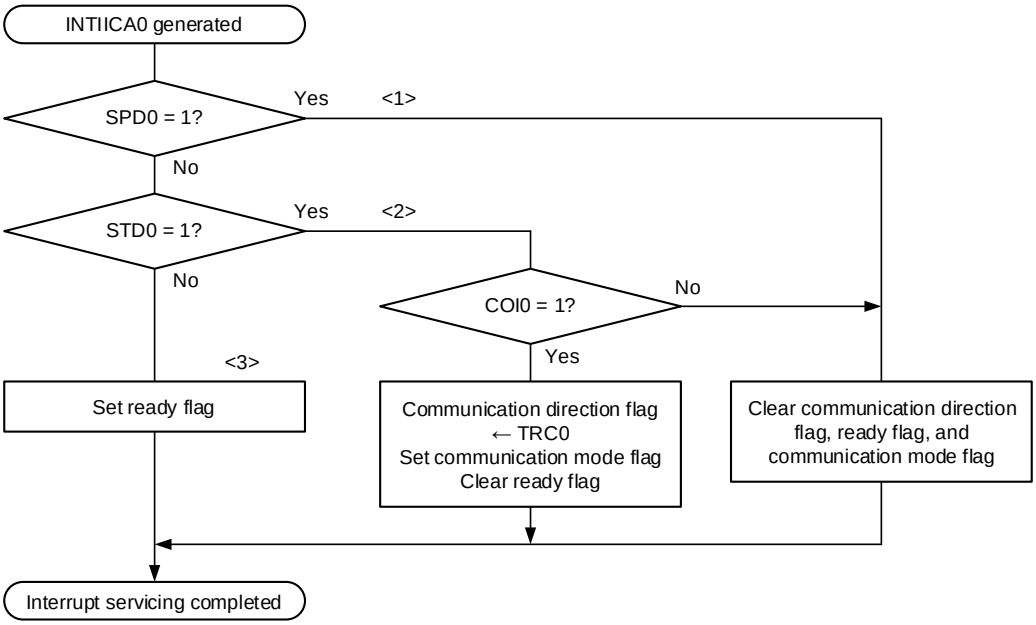
Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

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 17-31** Slave Operation Flowchart (2).

Figure 17-31. Slave Operation Flowchart (2)



17.5.17 Timing of I²C Interrupt Request (INTIICA0) Occurrence

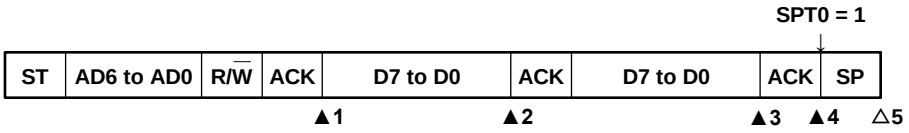
The timing of transmitting or receiving data and generation of interrupt request signal INTIICA0, and the value of the IICA status register 0 (IICS0) when the INTIICA0 signal is generated are shown below.

Remark ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
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

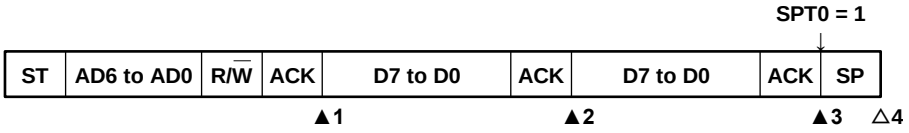


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B
▲3: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note]
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

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

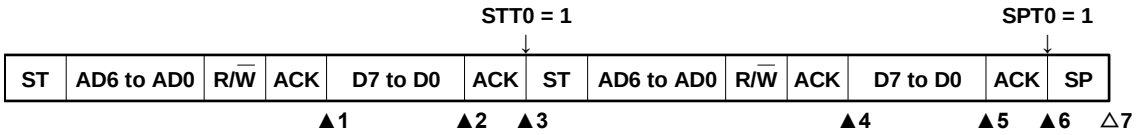


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B
▲3: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0 = 0

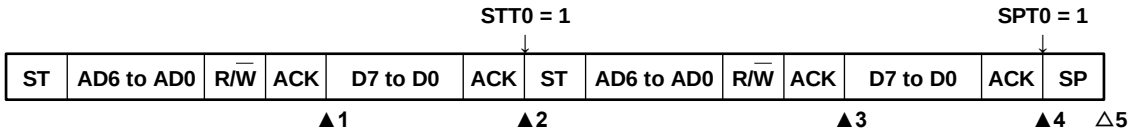


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)Note 1
▲3: IICS0 = 1000××00B (Clears the WTIM0 bit to 0Note 2, sets the STT0 bit to 1)
▲4: IICS0 = 1000×110B
▲5: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)Note 3
▲6: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△7: IICS0 = 00000001B

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

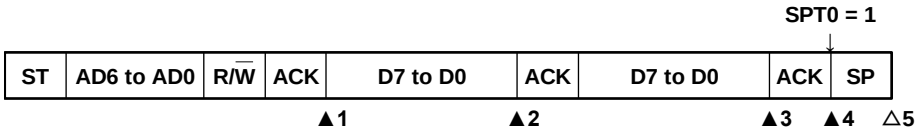


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
▲3: IICS0 = 1000×110B
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0

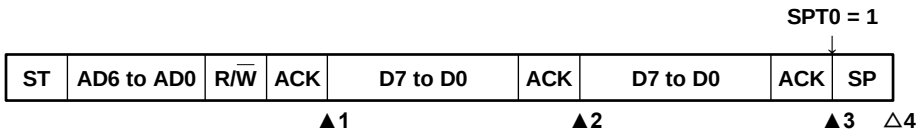


▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×000B
▲3: IICS0 = 1010×000B (Sets the WTIM0 bit to 1)[Note]
▲4: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

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



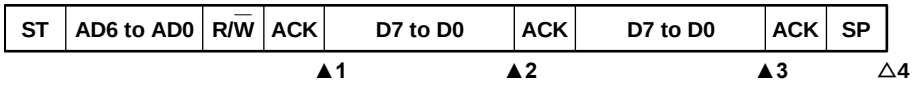
▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×100B
▲3: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00001001B

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

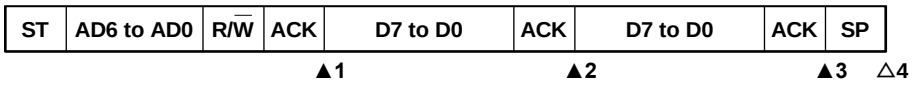
(i) When WTIM0 = 0



▲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

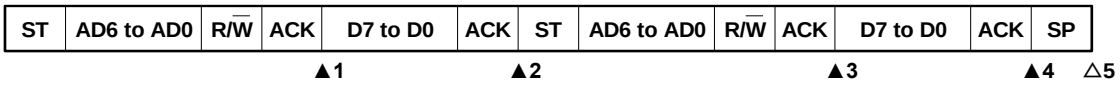
(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲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

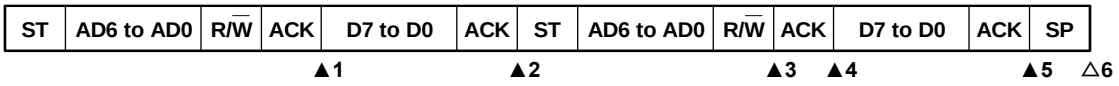
(i) When WTIM0 = 0 (after restart, does not match address (= extension code))



▲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))

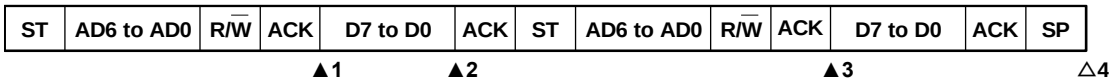


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×110B
▲5: IICS0 = 0010××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

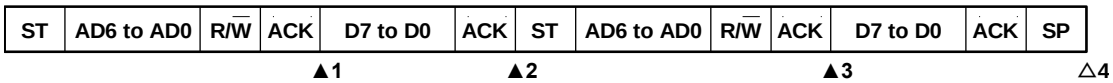
(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))



▲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

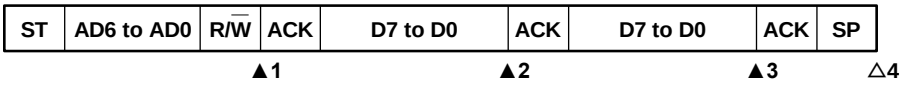
Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(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

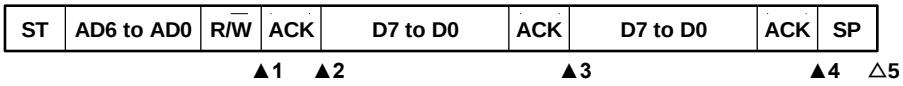
(i) When WTIM0 = 0



▲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

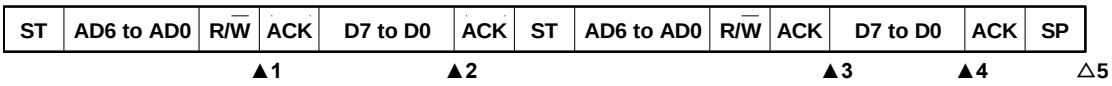


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

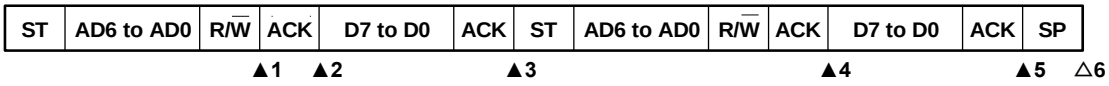
(i) When WTIM0 = 0 (after restart, matches SVA0)



▲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)

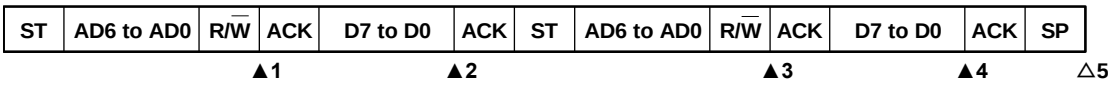


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0001×110B
▲5: IICS0 = 0001××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

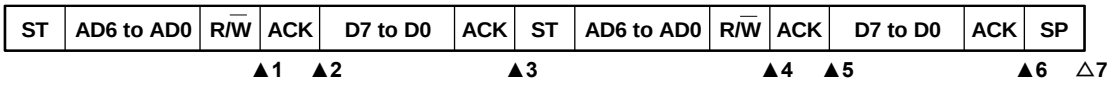
(i) When WTIM0 = 0 (after restart, extension code reception)



▲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)

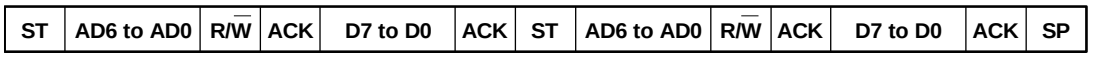


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0010×010B
▲5: IICS0 = 0010×110B
▲6: IICS0 = 0010××00B
△7: IICS0 = 00000001B

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))

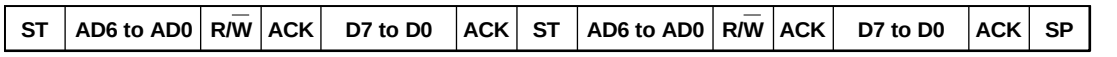


▲1 ▲2 ▲3 △4

▲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))



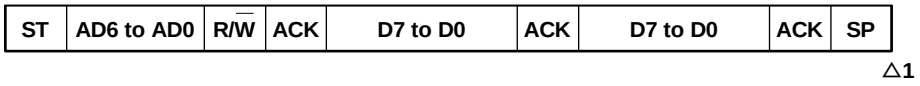
▲1 ▲2 ▲3 ▲4 △5

▲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

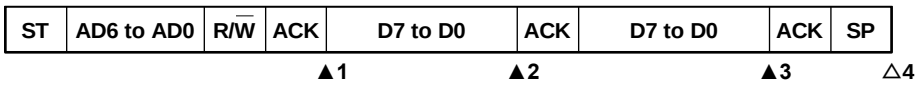
Remark △: Generated only when SPIE0 = 1

(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

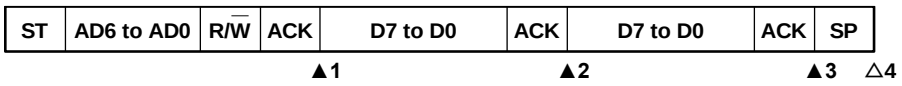
(i) When WTIM0 = 0



▲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

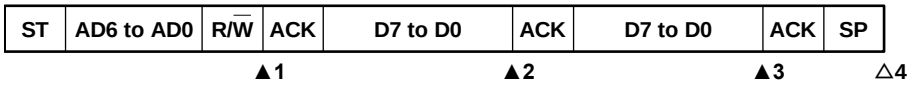


▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

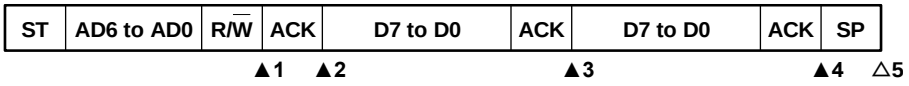
(i) When WTIM0 = 0



▲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

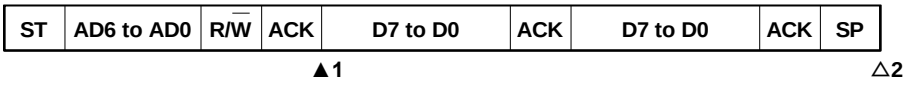
▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

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)

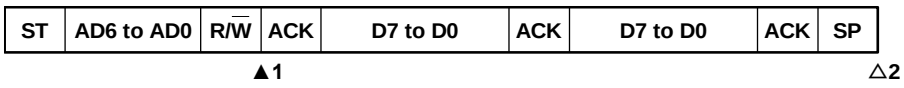


▲1 △2

▲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

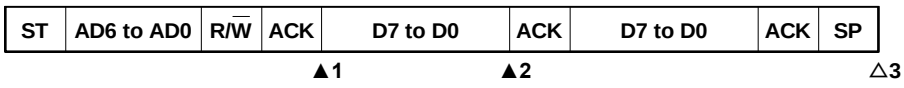
Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

(c) When arbitration loss occurs during transmission of data

(i) When WTIM0 = 0



▲1: IICS0 = 10001110B

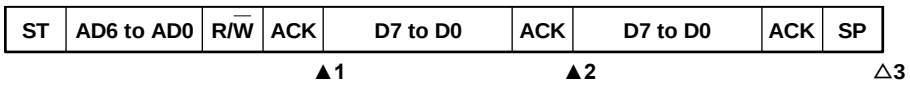
▲2: IICS0 = 01000000B

△3: IICS0 = 00000001B

Remark ▲: Always generated

△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1

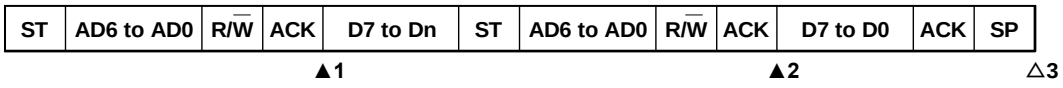


▲1: IICS0 = 10001110B
▲2: IICS0 = 01000100B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

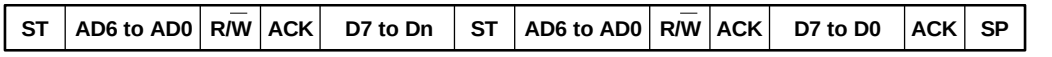
(i) Not extension code (Example: unmatches with SVA0)



▲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

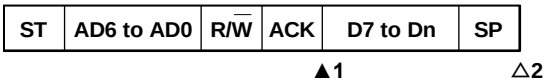
Remark ▲: Always generated

△: 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

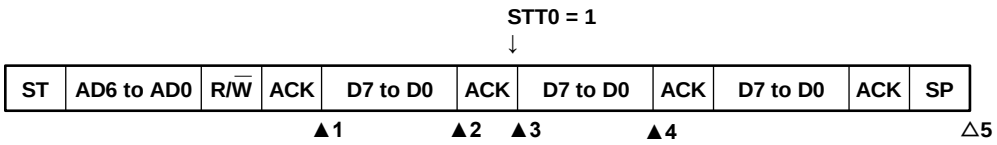
△: Generated only when SPIE0 = 1

×: Don't care

n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

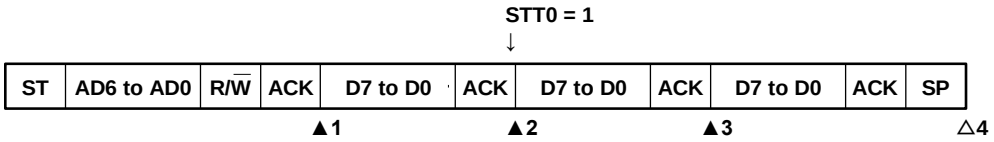
(i) When WTIM0 = 0



▲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

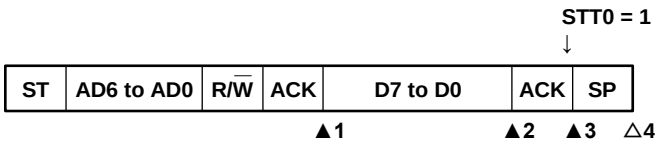


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets the STT0 bit to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

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

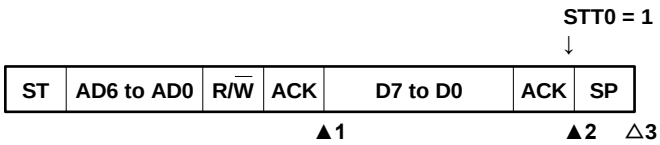
(i) When WTIM0 = 0



▲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

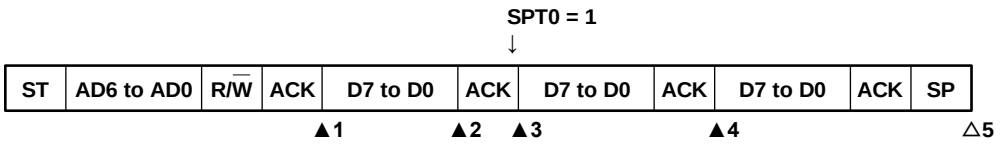


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
△3: IICS0 = 01000001B

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

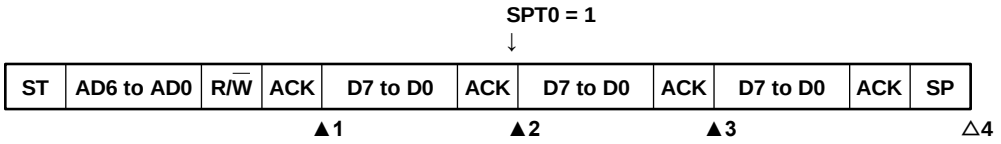
(i) When WTIM0 = 0



▲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

17.6 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.

Figure 17-32 and **Figure 17-33** show timing charts of the data communication.

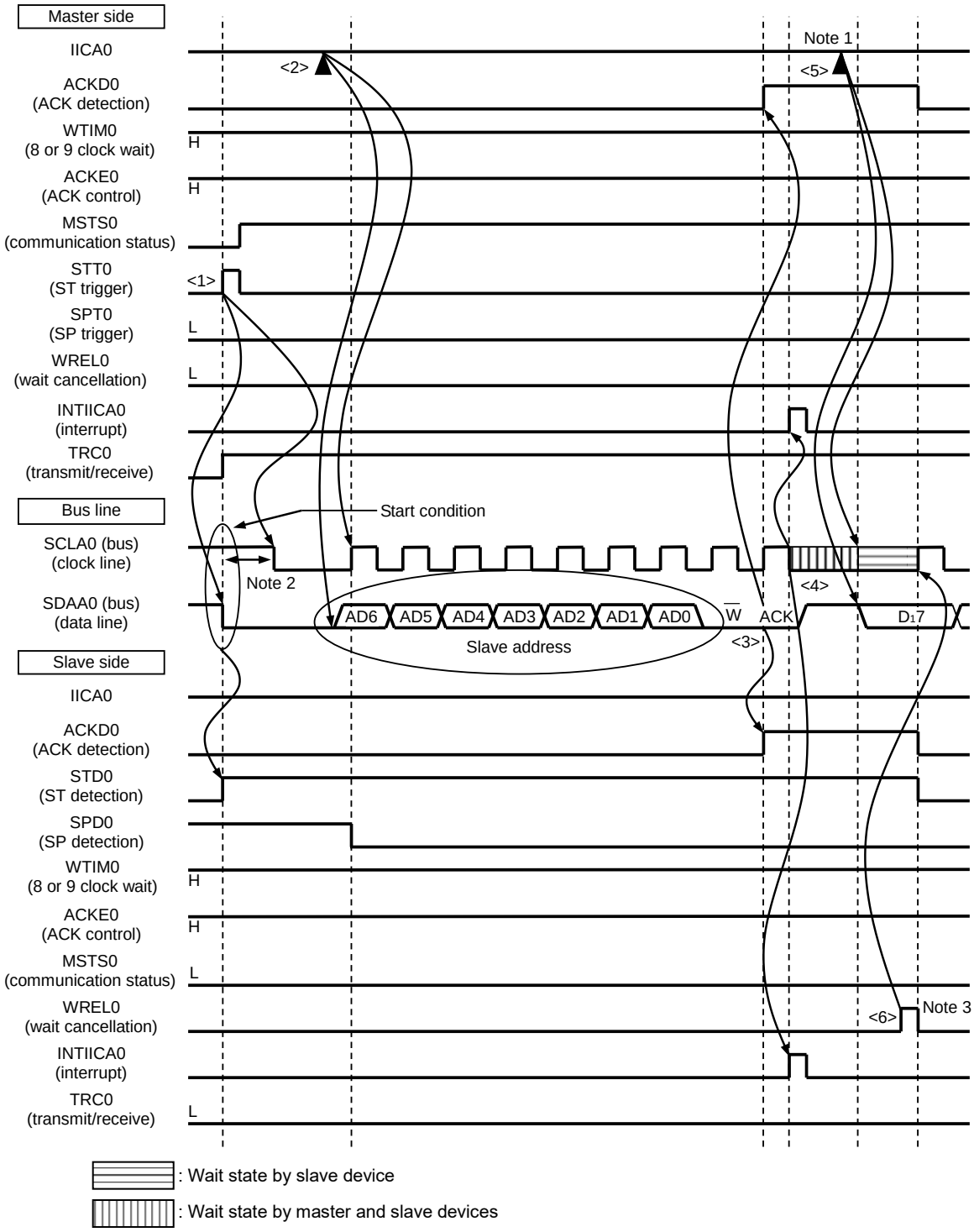
The IICA shift register 0 (IICA0)'s shift operation is synchronized with the falling edge of the serial clock (SCLA0). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAA0 pin.

Data input via the SDAA0 pin is captured into IICA0 at the rising edge of SCLA0.

17.6.1 Example Timing Charts of Master to Slave Communications

Figure 17-32. 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 17-32** 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.

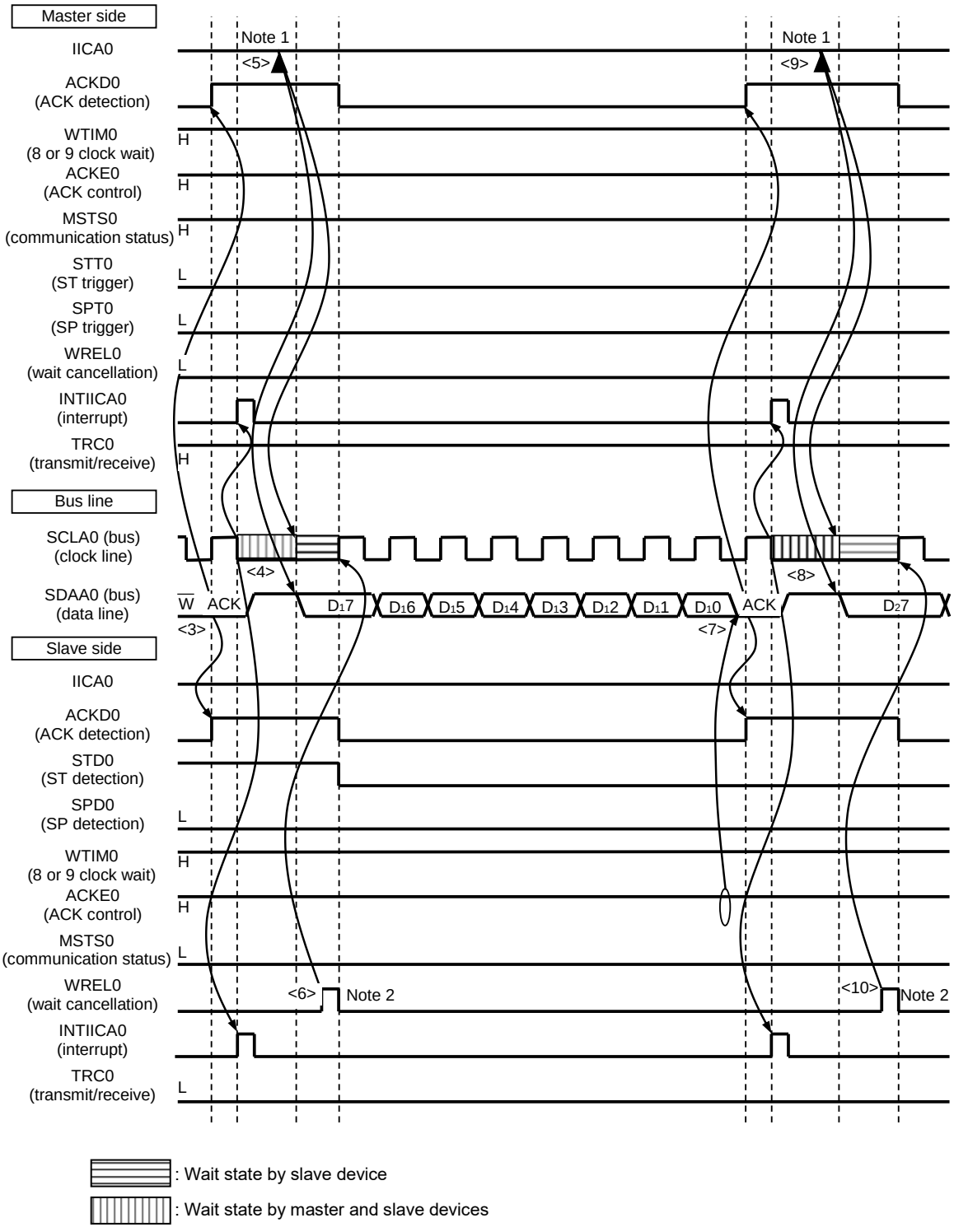
<6> If the slave device releases the wait status (WREL0 = 1), the master device starts transferring data to the slave device.

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 17-32** represent the entire procedure for communicating data using the I^2C bus. **Figure 17-32** (1) Start condition ~ address ~ data shows the processing from <1> to <6>, **Figure 17-32** (2) Address ~ data ~ data shows the processing from <3> to <10>, and **Figure 17-32** (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 17-32. 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 17-32** 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.

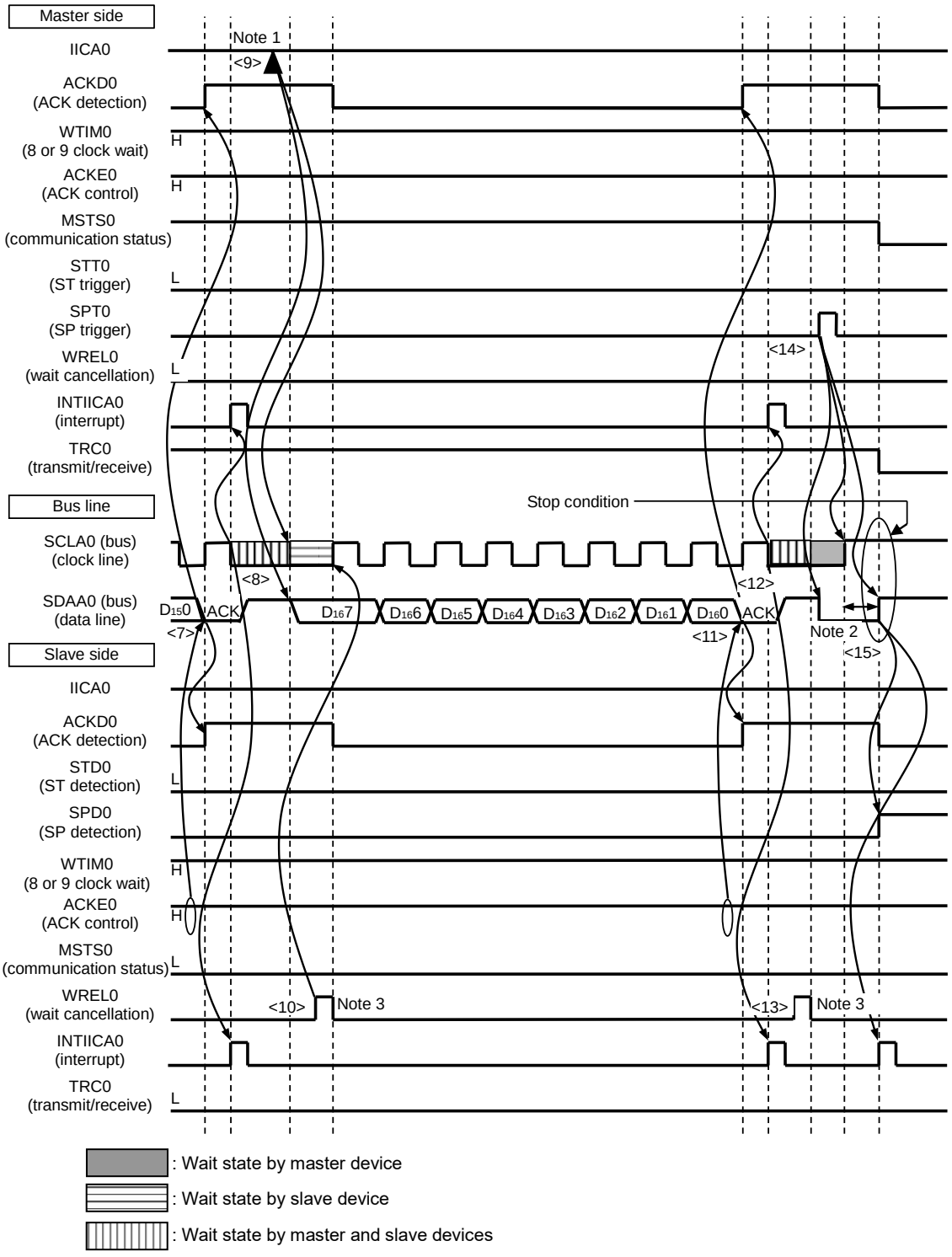
<10> The slave device reads the received data and releases the wait status (WREL0 = 1). The master device then starts transferring data to the slave device.

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 17-32** represent the entire procedure for communicating data using the I^2C bus.
Figure 17-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, **Figure 17-32** (2) Address ~ data ~ data shows the processing from <3> to <10>, and **Figure 17-32** (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 17-32. 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 17-32** 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).

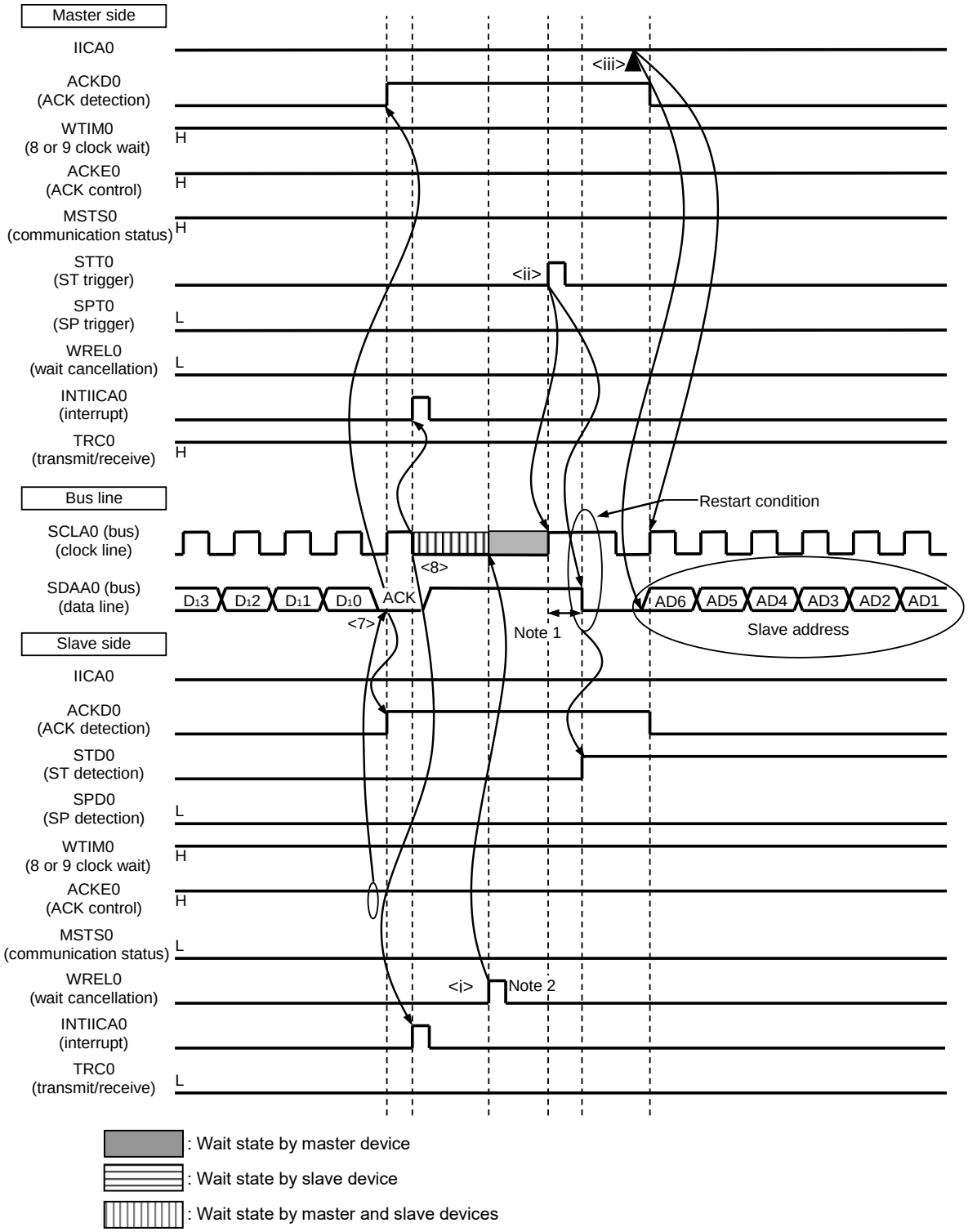
<14> By the master device setting a stop condition trigger (SPT0 = 1), the bus data line is cleared (SDAA0 = 0) and the bus clock line is set (SCLA0 = 1). After the stop condition setup time has elapsed, by setting the bus data line (SDAA0 = 1), the stop condition is then generated (i.e. SCLA0 =1 changes 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 17-32** represent the entire procedure for communicating data using the I^2C bus. **Figure 17-32** (1) Start condition ~ address ~ data shows the processing from <1> to <6>, **Figure 17-32** (2) Address ~ data ~ data shows the processing from <3> to <10>, and **Figure 17-32** (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 17-32. 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 17-32** (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).

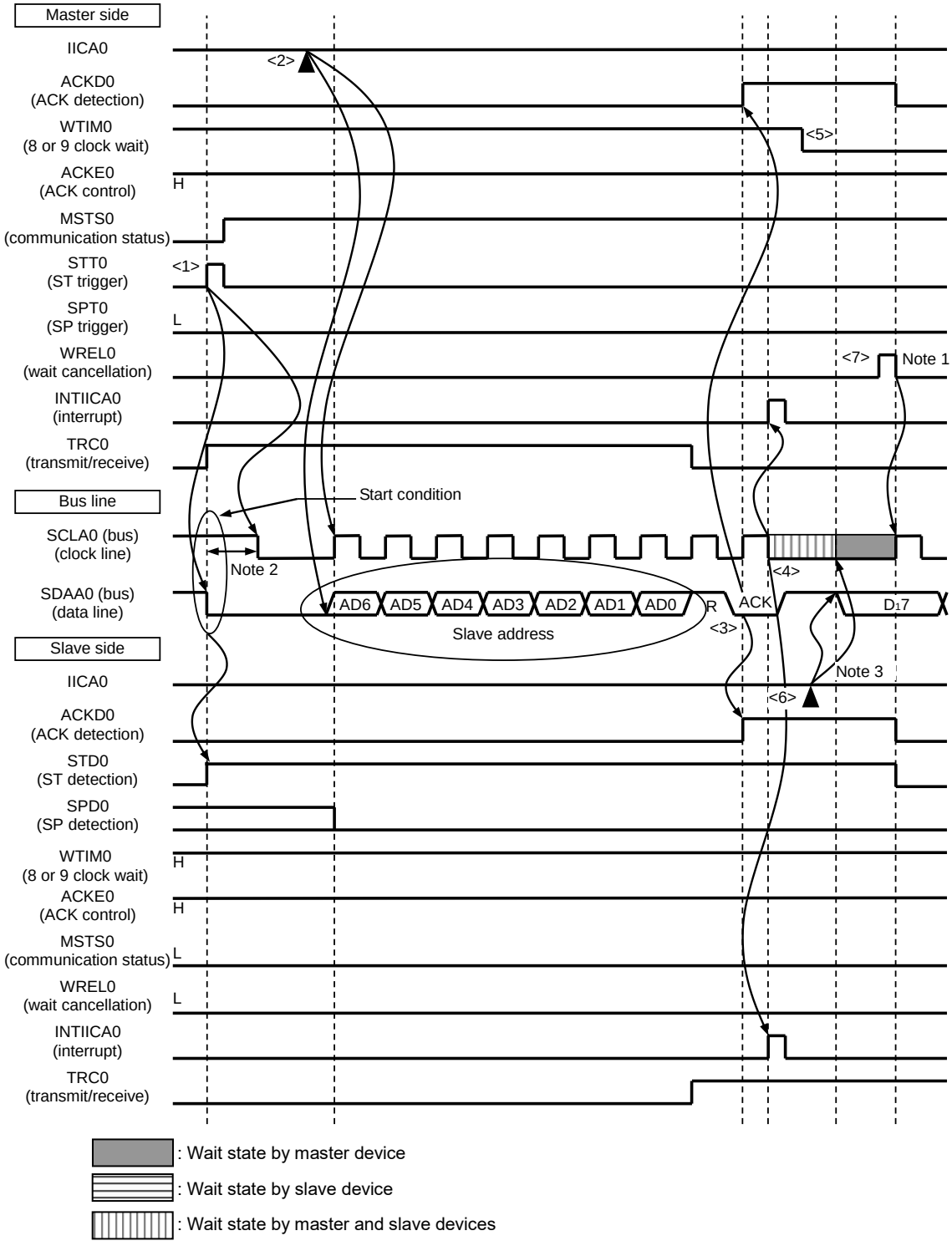
<ii> The start condition trigger is set again by the master device (STT0 = 1) and a start condition (i.e. SCLA0 =1 changes SDAA0 from 1 to 0) is generated once the bus clock line goes high (SCLA0 = 1) and the bus data line goes low (SDAA0 = 0) after the restart condition setup time has elapsed. When 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.

17.6.2 Example Timing Charts of Slave to Master Communications

Figure 17-33. 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 17-33** 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.

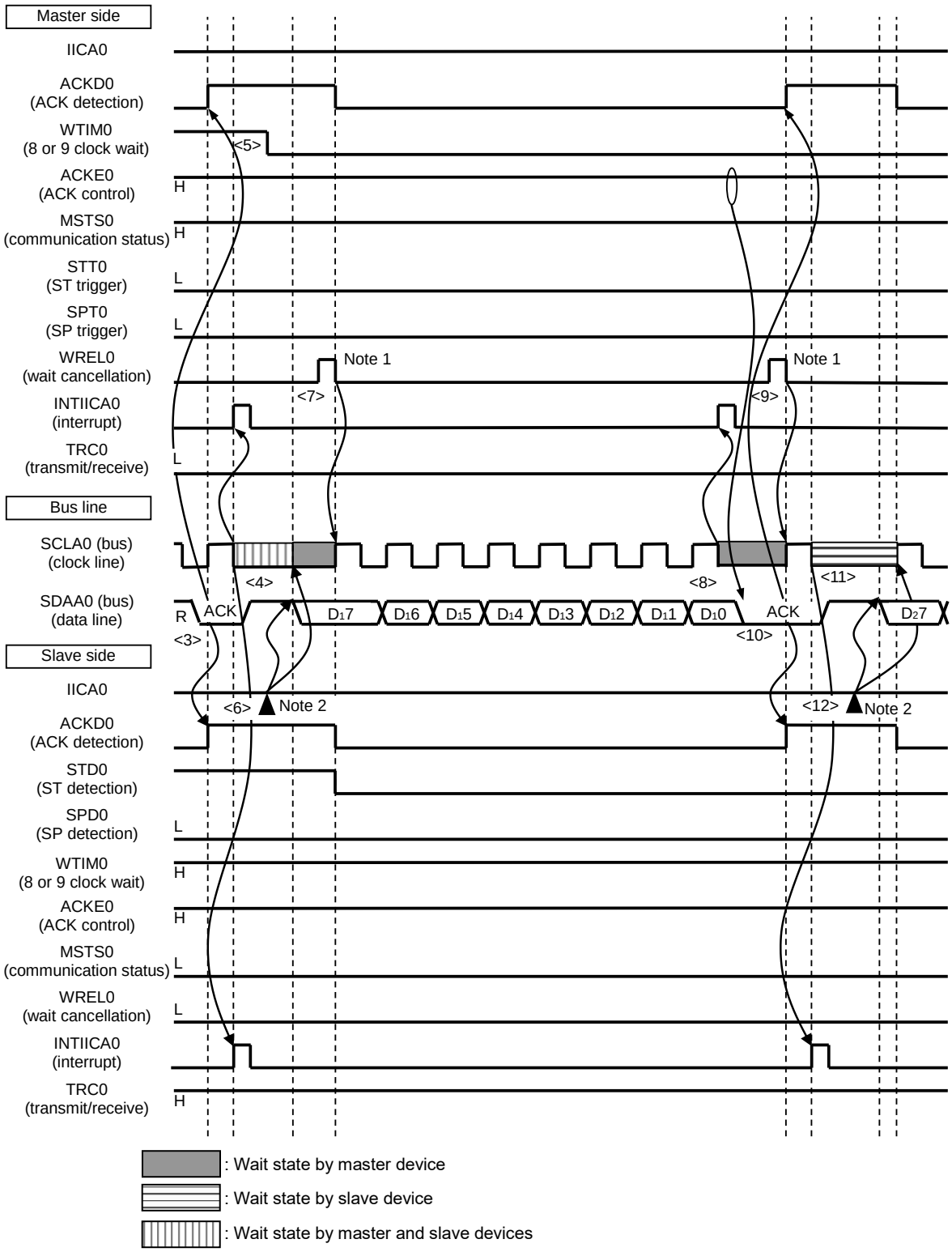
<7> The master device releases the wait status (WREL0 = 1) and starts transferring data from the slave device to the master device.

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 17-33** represent the entire procedure for communicating data using the I^2C bus. **Figure 17-33** (1) Start condition ~ address ~ data shows the processing from <1> to <7>, **Figure 17-33** (2) Address ~ data ~ data shows the processing from <3> to <12>, and **Figure 17-33** (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 17-33. 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 17-33** 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).

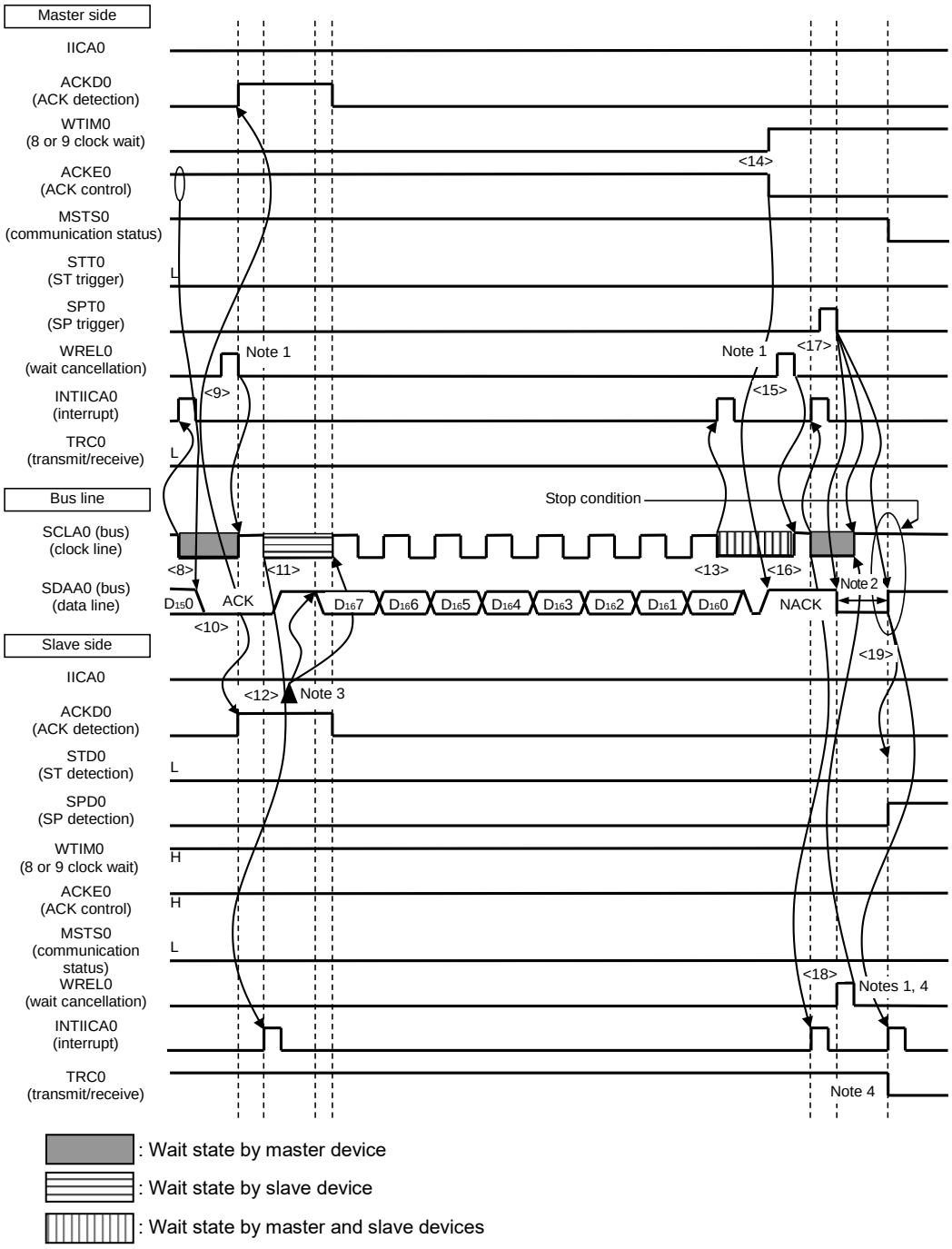
<12> By the slave device writing the data to transmit to the IICA0 register, the wait status set by the slave device is released. The slave device then starts transferring data to the master device.

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 17-33** represent the entire procedure for communicating data using the I^2C bus. **Figure 17-33** (1) Start condition ~ address ~ data shows the processing from <1> to <7>, **Figure 17-33** (2) Address ~ data ~ data shows the processing from <3> to <12>, and **Figure 17-33** (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 17-33. 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 17-33** 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 = 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 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 17-33** represent the entire procedure for communicating data using the I^2C bus. **Figure 17-33** (1) Start condition ~ address ~ data shows the processing from <1> to <7>, **Figure 17-33** (2) Address ~ data ~ data shows the processing from <3> to <12>, and **Figure 17-33** (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

CHAPTER 18 LIN/UART MODULE (RLIN3)

18.1 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 18-1 gives the LIN/UART module specifications and **Figures 18-1**, **18-2**, and **Tabel 18-2** show block diagrams of the LIN/UART module.

Table 18-1. LIN/UART Module Specifications (1/3)

Item		Specifications	
Channel count		RL78/F22: 1 channel RL78/F25: 3 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 • Break delimiter transmission width: 1 to 4 Tbits • Inter-byte space (header): 0 to 7 Tbits (space between Sync field and ID field) Note 1 • Response space: 0 to 7 Tbits Note 1 • Inter-byte space: 0 to 3 Tbits (space between data bytes in response area) • Wake-up: 1 to 16 Tbits
		Slave	• Break reception width: - 9.5 or 10.5 Tbits [for fixed baud rate] - 10 or 11 Tbits [for auto baud rate] • Response space: 0 to 7 Tbits • Inter-byte space: 0 to 3 Tbits (space between data bytes in response area) • Wake-up: 1 to 16 Tbits
	Checksum	• Automatic operation for both transmission and reception • Classic or enhanced selectable (for each frame)	
	Response field data byte count	Variable from 0 to 8 bytes 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 • 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 • 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 • Wake-up transmission (1 to 16 Tbits) • Wake-up reception Low width of input signals measured	
	Status	Master	• Successful frame/wake-up transmission • Successful header transmission • Successful frame/wake-up reception Note 2 • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)

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.

Table 18-1. LIN/UART Module Specifications (2/3)

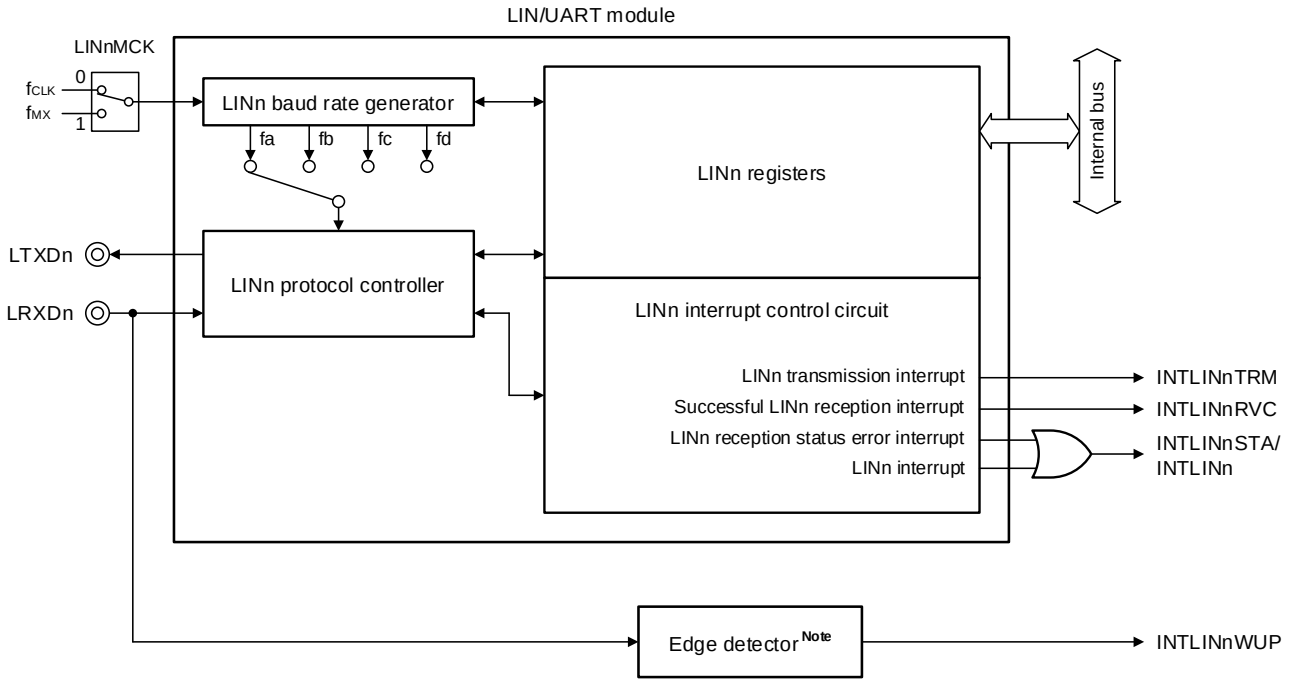
Item		Specifications	
LIN communication function	Status	Slave	• Successful frame/wake-up transmission • Successful frame/wake-up reception Note • Successful header reception • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode) • Detection status on a dominant of 0.5 Tbits or more in a response space. • Detection status on Break reception and Sync field reception.
	Error status	Master	• Bit error • Checksum error • Frame timeout error/response timeout error • Physical bus error • Framing error • Response preparation error
		Slave	• Bit error • Checksum error • Frame timeout error/response timeout error • Sync field error • ID parity error • Framing error • 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 • Successful frame/wake-up reception Note • Error detection
		Slave	• Successful frame/wake-up transmission • Header/frame/wake-up reception Note • Error detection

Note For wake-up reception, the low-level width of the input signal is indicated.

Table 18-1. LIN/UART Module Specifications (3/3)

Item		Specifications
UART communication function	Data buffer	• Transmission data buffer/transmission data buffer for wait (exclusively for transmission; data length is 1, bit lengths 7, 8, and 9 bits are supported) • UART buffer (exclusively for transmission; variable data length from 1 to 9 bytes; bit length of 7 and 8 bits are supported) • Reception data buffer (exclusively for reception; data length is 1, bit lengths 7, 8, and 9 bits are supported)
	Data format	• Character length: 7 or 8 bits 9 bits including the expansion bit supported. • Transmission stop bit: 1 or 2 bits • Parity function: odd, even, 0, or none • LSB- or MSB-first transfer selectable • Reverse input/output of transmission/reception data
	Status	• Transmission status • Reception status • Successful UART buffer transmission • Error SUM • Expansion bit detection • ID match • Reset mode status
	Error status	• Bit error • Framing error • Parity error • 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. • 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 • Successful reception • Status detection

Figure 18-1. LIN/UART Module Block Diagram (Each Channel)

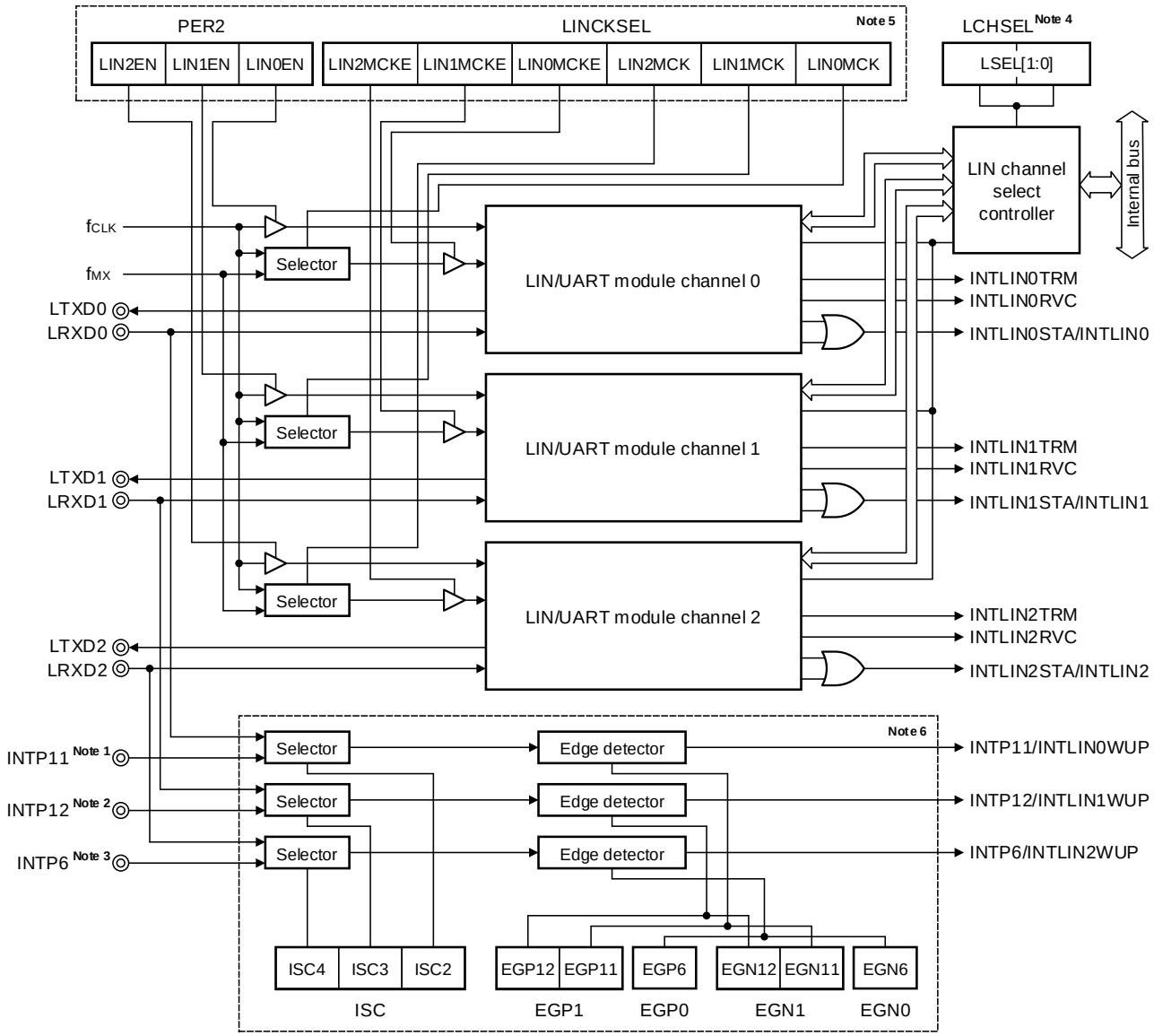


LTXDn, LRXDn: LIN/UART module I/O pins
LINn baud rate generator: Generates the LIN/UART module communication clock signal.
LINn registers: LIN/UART module registers
LINn interrupt controller: Controls interrupt requests generated by the LIN/UART module.

Note For details, see section **22.3.4 External Interrupt Rising/Falling Edge Enable registers (EGP0, EGP1, EGN0, EGN1)** in **CHAPTER 22 INTERRUPT FUNCTIONS**.

Remark n = 0 to 2

Figure 18-2. LIN/UART Module Block Diagram (Whole Module)



Notes 1. INTP11 is mounted only on 64-, 80-, and 100-pin products of RL78/F25.

2. INTP12 is mounted only on 64-, 80-, and 100-pin products of RL78/F25.

3. INTP6 is mounted only on 48-pin products of RL78/F22, and all products of RL78/F25.

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 LSEL[1:0] bits in the LCHSEL register to 00B.

5. For details, see **CHAPTER 5 CLOCK GENERATOR**.

6. For details, see **CHAPTER 22 INTERRUPT FUNCTIONS**.

Table 18-2 shows the I/O pins used in the LIN/UART module.

Table 18-2. LIN/UART Module I/O Pins

Module Symbol	Pin Name	Input/Output	Function
LINn	LRXDn	Input	• LIN communication function • Input pin of the UART communication function
	LTXDn	Output	• LIN communication function • Output pin of the UART communication function

Remark n = 0 to 2

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

18.2 Register Descriptions

Table 18-3 lists the LIN/UART module-related registers.

Table 18-3. List of LIN/UART Module-Related Registers (1/2)

Address	Register Name	Symbol (n = 0 to 2)	After Reset	Access Size	LIN Master	LIN Slave	UART
F02C2H	Peripheral enable register 2	PER2	00H	1, 8	√	√	√
F0073H	Input switch control register	ISC	00H	1, 8	√	√	√
F007BH	LIN channel select register Note	LCHSEL	00H	8	√	√	√
F02C4H	LIN clock select register	LINCKSEL	00H	1, 8	√	√	√
FFF38H	External interrupt rising edge enable register 0	EGP0	00H	1, 8	√	√	√
FFF39H	External interrupt falling edge enable register 0	EGN0	00H	1, 8	√	√	√
FFF3AH	External interrupt rising edge enable register 1	EGP1	00H	1, 8	√	√	√
FFF3BH	External interrupt falling edge enable register 1	EGN1	00H	1, 8	√	√	√
F06C1H	LIN wake-up baud rate select register	LWBRn	00H	8	√	√	√
F06C2H	LIN/UART baud rate prescaler register	LBRPn	0000H	8, 16	–	√	√
	LIN/UART baud rate prescaler 0 register	LBRPn0			√	√	√
F06C3H	LIN/UART baud rate prescaler 1 register	LBRPn1			√	√	√
F06C4H	LIN self-test control register	LSTCn	00H	8	√	√	–
F06C5H	UART stand-by control register	LUSCn	00H	8	–	–	√
F06C8H	LIN/UART mode register	LMDn	00H	8	√	√	√
F06C9H	LIN break field configuration register/ UART configuration register	LBFCn	00H	8	√	√	√
F06CAH	LIN/UART space configuration register	LSCn	00H	8	√	√	√
F06CBH	LIN wake-up configuration register	LWUPn	00H	8	√	√	–
F06CCH	LIN interrupt enable register	LIEn	00H	8	√	√	–
F06CDH	LIN/UART error detection enable register	LEDEn	00H	8	√	√	√
F06CEH	LIN/UART control register	LCUCn	00H	8	√	√	√
F06D0H	LIN/UART transmission control register	LTRCn	00H	8	√	√	√
F06D1H	LIN/UART mode status register	LMSTn	00H	8	√	√	√
F06D2H	LIN/UART status register	LSTn	00H	8	√	√	√
F06D3H	LIN/UART error status register	LESTn	00H	8	√	√	√
F06D4H	LIN/UART data field configuration register	LDFCn	00H	8	√	√	√
F06D5H	LIN/UART ID buffer register	LIDBn	00H	8	√	√	√
F06D6H	LIN checksum buffer register	LCBRn	00H	8	√	√	–
F06D7H	UART data buffer 0 register	LUDBn0	00H	8	–	–	√
F06D8H	LIN/UART data buffer 1 register	LDBn1	00H	8	√	√	√
F06D9H	LIN/UART data buffer 2 register	LDBn2	00H	8	√	√	√
F06DAH	LIN/UART data buffer 3 register	LDBn3	00H	8	√	√	√
F06DBH	LIN/UART data buffer 4 register	LDBn4	00H	8	√	√	√
F06DCH	LIN/UART data buffer 5 register	LDBn5	00H	8	√	√	√
F06DDH	LIN/UART data buffer 6 register	LDBn6	00H	8	√	√	√
F06DEH	LIN/UART data buffer 7 register	LDBn7	00H	8	√	√	√
F06DFH	LIN/UART data buffer 8 register	LDBn8	00H	8	√	√	√
F06E0H	UART operation enable register	LUOERn	00H	8	–	–	√
F06E1H	UART option register 1	LUORn1	00H	8	–	–	√

√: Used, – : Not used (When writing to a register not used, write 00H.)

Note RL78/F25 only.

Caution Registers in the area F06C1H to F06EEH are specified by the LSEL[1:0] bits of LCHSEL register. Channel 1 registers and channel 2 registers are only on RL78/F25 products.

Table 18-3. List of LIN/UART Module-Related Registers (2/2)

Address	Register Name	Symbol (n = 0 to 2)	After Reset	Access Size	LIN Master	LIN Slave	UART
F06E4H	UART transmission data register	LUTDRn	0000H	8, 16	–	–	√
		LUTDRnL			–	–	√
F06E5H		LUTDRnH			–	–	√
F06E6H	UART reception data register	LURDRn	0000H	8, 16	–	–	√
		LURDRnL			–	–	√
F06E7H		LURDRnH			–	–	√
F06E8H	UART wait transmission data register	LUWTDRn	0000H	8, 16	–	–	√
		LUWTDRnL			–	–	√
F06E9H		LUWTDRnH			–	–	√
F06ECH	LIN break and sync field detection status register	LBSSn	00H	8	–	√	–
F06EEH	LIN response space dominant signal detection status register	LRSSn	00H	8	–	√	–

√: Used, – : Not used (When writing to a register not used, write 00H.)

Caution Registers in the area F06C1H to F06EEH are specified by the LSEL[1:0] bits of LCHSEL register. Channel 1 registers and channel 2 registers are only on RL78/F25 products.

18.2.1 LIN Registers for Master Mode

(1) Input Switch Control Register (ISC)

The ISC2, ISC3, and ISC4 bits in the ISC register are used in the LIN/UART module (RLIN3).

Setting bit 2, bit 3, or bit 4 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	ISC4	ISC3	ISC2	0	ISC0

ISC4	Switching inputs for external interrupt INTP6
0	INTP6 pin input signal is set as external interrupt input.
1	LRXD2 pin input signal is set as external interrupt input.

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)

Cautions 1. Bits 7 to 5 and 1 should always be set to 0.

2. Be sure to set the ISC3 and ISC4 bits to 0 in RL78/F22 products.

(2) LIN Channel Select Register (LCHSEL)

Address: F007BH

Symbol	7	6	5	4	3	2	1	0
LCHSEL	0	0	0	0	0	0	LSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 2	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
1, 0	LSEL[1:0]	LIN Channel Select	00B: Selects LIN0. (LIN0 registers can be accessed.) 01B: Selects LIN1. (LIN1 registers can be accessed.) 10B: Selects LIN2. (LIN2 registers can be accessed.) 11B: Setting is prohibited.	R/W

- **LSEL[1:0] bits (LIN channel select bits)**

  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 F06EEH.
  Setting values to the LSEL[1:0] bits map all the registers of the corresponding channel on the register window.
  Setting the LSEL[1:0] bits to 00B maps the LIN0 registers.
  Setting the LSEL[1:0] bits to 01B maps the LIN1 registers.
  Setting the LSEL[1:0] bits to 10B maps the LIN2 registers.

Cautions 1. RL78/F22 products do not have this register.

2. With the RL78/F25 products, set the LSEL[1:0] bits 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.

Writing to the PER2 register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation sets this register to 00H.

Format of Peripheral Enable Register 2 (PER2)

Address: F02C2H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	1	<0>
PER2	0	AAUEN	CAN1EN Note	LIN2EN Note	LIN1EN Note	LIN0EN	0	CAN0EN Note

LIN2EN Note	Control of LIN2 input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by LIN2. • LIN2 is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by LIN2.

LIN1EN Note	Control of LIN1 input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by LIN1. • LIN1 is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by LIN1.

LIN0EN	Control of LIN0 input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by LIN0. • LIN0 is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by LIN0.

Note Only in the RL78/F25 products.

Caution Be sure to clear the following bits to 0.

- **Bits 0, 1, 3, 4, 5, and 7 in the RL78/F22 products**
- **Bits 1 and 7 in the RL78/F25 products**

(4) 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.

Format of LIN Clock Select Register (LINCKSEL)

Address: F02C4H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	3	<2>	<1>	<0>
LINCKSEL	0	LIN2MCKE Note	LIN1MCKE Note	LIN0MCKE	0	LIN2MCK Note	LIN1MCK Note	LIN0MCK

LIN2MCKE Note	Control of supplying or stopping LIN2 communication clock source
0	Stops LIN2 communication clock source supply
1	Enables LIN2 communication clock source supply

LIN1MCKE Note	Control of supplying or stopping LIN1 communication clock source
0	Stops LIN1 communication clock source supply
1	Enables LIN1 communication clock source supply

LIN0MCKE	Control of supplying or stopping LIN0 communication clock source
0	Stops LIN0 communication clock source supply
1	Enables LIN0 communication clock source supply

LIN2MCK Note	Control of selecting LIN2 communication clock source
0	Selects the f_{CLK} clock
1	Selects the f_{MX} clock

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/F25.

Cautions 1. Select the LINn (n = 0 to 2) operating clock with the LINnMCK bit before setting the LINnMCKE bit to 1.

2. When operating LINn in SNOOZE mode, set the LINnMCK bit to 0.

3. In case of LINnMCK is set to 1 (select f_{MX} clock), 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) External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)

For details, see section **22.3.4 External Interrupt Rising/Falling Edge Enable registers (EGP0, EGP1, EGN0, EGN1)** in **CHAPTER 22 INTERRUPT FUNCTIONS.**

(6) LIN Wake-up Baud Rate Select Register (LWBRn)

Address: F06C1H

Symbol	7	6	5	4	3	2	1	0
LWBRn	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
7 to 4	NSPB[3:0]	Bit Sampling Count Select	0000B: 16 sampling 1111B: 16 sampling Settings other than the above are prohibited.	R/W
3 to 1	LPRS[2:0]	Prescaler Clock Select	000B: 1/1 001B: 1/2 010B: 1/4 011B: 1/8 100B: 1/16 101B: 1/32 110B: 1/64 111B: 1/128	R/W
0	LWBR0	Wake-up Baud Rate Select	0: When LIN1.3 is used 1: When LIN 2.x is used	R/W

Set the LWBRn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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).

- **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.

- **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.

(7) LIN/UART Baud Rate Prescaler 0 Register (LBRPn0)

Address: F06C2H

Symbol	7	6	5	4	3	2	1	0
LBRPn0								
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 of the LWBRn register.

(8) LIN/UART Baud Rate Prescaler 1 Register (LBRPn1)

Address: F06C3H

Symbol	7	6	5	4	3	2	1	0
LBRPn1								
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 of the LWBRn register (prescaler clock select bits).

(9) LIN Self-Test Control Register (LSTCn)

Address: F06C4H

Symbol	7	6	5	4	3	2	1	0
LSTCn								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 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 **18.6 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 **18.6 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

Symbol	7	6	5	4	3	2	1	0
LMDn	0	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
7, 6	–	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. 1: The noise filter is disabled.	R/W
4	LIOS	LIN Interrupt Output Select	0: LIN interrupt is used. 1: Transmission interrupt, successful reception interrupt, and reception status interrupt are used.	R/W
3, 2	LCKS[1:0]	LIN System Clock Select	00B: fa (Clock generated by baud rate prescaler 0) 01B: fb (1/2 clock generated by baud rate prescaler 0) 10B: fc (1/8 clock generated by baud rate prescaler 0) 11B: fd (1/2 clock generated by baud rate prescaler 1)	R/W
1, 0	LMD[1:0]	LIN/UART Mode Select	00B: LIN master mode	R/W

Set the LMDn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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.

- **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 **18.9 Interrupts**.

- **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).

- **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.

(11) LIN Break Field Configuration Register/UART Configuration Register (LBFCn)

Address: F06C9H

Symbol	7	6	5	4	3	2	1	0
LBFCn	0	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
7, 6	–	Reserved	These bits are always read as 0. The write value should always be 0.	–
5, 4	BDT[1:0]	Transmission Break Delimiter (High) Width Select	00B: 1 Tbit 01B: 2 Tbits 10B: 3 Tbits 11B: 4 Tbits	R/W
3 to 0	BLT[3:0]	Transmission Break (Low) Width Select	0000B: 13 Tbits 0001B: 14 Tbits 0010B: 15 Tbits : 1110B: 27 Tbits 1111B: 28 Tbits	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.

- **BDT[1:0] 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.

- **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.

(12) LIN/UART Space Configuration Register (LSCn)

Address: F06CAH

Symbol	7	6	5	4	3	2	1	0
LSCn	0	0	IBS[1:0]		0	IBSH[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7, 6	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
5, 4	IBS[1:0]	Inter-Byte Space Select	00B: 0 Tbit 01B: 1 Tbit 10B: 2 Tbits 11B: 3 Tbits	R/W
3	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
2 to 0	IBSH[2:0]	Inter-Byte Space (Header)/ Response Space Select	000B: 0 Tbit 001B: 1 Tbit 010B: 2 Tbits 011B: 3 Tbits 100B: 4 Tbits 101B: 5 Tbits 110B: 6 Tbits 111B: 7 Tbits	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.

- **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.

- **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.

(13) LIN Wake-up Configuration Register (LWUPn)

Address: F06CBH

Symbol	7	6	5	4	3	2	1	0
LWUPn	WUTL[3:0]				0	0	0	0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 4	WUTL[3:0]	Wake-up Transmission Low Width Select	0000B: 1 Tbit 0001B: 2 Tbits 0010B: 3 Tbits 0011B: 4 Tbits : 1100B: 13 Tbits 1101B: 14 Tbits 1110B: 15 Tbits 1111B: 16 Tbits	R/W
3 to 0	–	Reserved	These bits are always read as 0. The write value should always be 0.	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 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).

(14) LIN Interrupt Enable Register (LIEn)

Address: F06CCH

Symbol	7	6	5	4	3	2	1	0
LIEn	0	0	0	0	SHIE	ERRIE	FRCIE	FTCIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 4	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
3	SHIE	Successful Header Transmission Interrupt Enable	0: Disables successful header transmission interrupt. 1: Enables successful header transmission interrupt.	R/W
2	ERRIE	Error Detection Interrupt Enable	0: Disables error detection interrupt. 1: Enables error detection interrupt.	R/W
1	FRCIE	Successful Frame/Wake-up Reception Interrupt Enable	0: Disables successful frame/wake-up reception interrupt. 1: Enables successful frame/wake-up reception interrupt.	R/W
0	FTCIE	Successful Frame/Wake-up Transmission Interrupt Enable	0: Disables successful frame/wake-up transmission interrupt. 1: Enables successful frame/wake-up transmission interrupt.	R/W

Set the LIEn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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.

- **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.

- **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.

- **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.

(15) LIN/UART Error Detection Enable Register (LEDEn)

Address: F06CDH

Symbol	7	6	5	4	3	2	1	0
LEDEn	LTES	0	0	0	FERE	FTERE	PBERE	BERE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7	LTES	Timeout Error Select	0: Frame timeout error 1: Response timeout error	R/W
6 to 4	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
3	FERE	Framing Error Detection Enable	0: Disables framing error detection. 1: Enables framing error detection.	R/W
2	FTERE	Timeout Error Detection Enable	0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.	R/W
1	PBERE	Physical Bus Error Detection Enable	0: Disables physical bus error detection. 1: Enables physical bus error detection.	R/W
0	BERE	Bit Error Detection Enable	0: Disables bit error detection. 1: Enables bit error detection.	R/W

Set the LEDEn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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 **18.4.6 Error Status**.

- **FERE bit (framing error detection enable bit)**

  The FERE bit enables or disables detection of the framing error.
  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 **18.4.6 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 **18.4.6 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 **18.4.6 Error Status**.

- **BERE bit (bit error detection enable bit)**

  The BERE bit enables or disables detection of the bit error.
  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 **18.4.6 Error Status.**

(16) LIN/UART Control Register (LCUCn)

Address: F06CEH

Symbol	7	6	5	4	3	2	1	0
LCUCn	0	0	0	0	0	0	OM1	OM0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 2	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
1	OM1	LIN Mode Select	0: LIN wake-up mode is caused. 1: LIN operation mode is caused.	R/W
0	OM0	LIN Reset	0: LIN reset mode is caused. 1: LIN reset mode is canceled.	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.

- **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 (Frame transmission or wake-up transmission/reception is started).

- **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.

Caution When transitioning from LIN mode with timeout error detection enabled (FTERE bit of LEDEn register is set to 1) to LIN reset mode and then to LIN mode, clear the error flag according to the procedure described in "18.3.1 LIN Reset Mode".

(17) LIN/UART Transmission Control Register (LTRCn)

Address: F06D0H

Symbol	7	6	5	4	3	2	1	0
LTRCn	0	0	0	0	0	0	RTS	FTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 2	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
1	RTS	Response Transmission/ Reception Start	0: Response transmission/reception is stopped in frame separate mode. 1: Response transmission/reception is started in frame separate mode.	R/W
0	FTS	Frame Transmission or Wake-up Transmission/ Reception Start	0: Frame transmission or wake-up transmission/reception is stopped. 1: Frame transmission or wake-up transmission/reception is started.	R/W

- **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.

- **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 cleared to 0 when the frame or wake-up communication is complete. Also, when transitioning to LIN reset mode, this bit becomes 0.

(18) LIN/UART Mode Status Register (LMSTn)

Address: F06D1H

Symbol	7	6	5	4	3	2	1	0
LMSTn	0	0	0	0	0	0	OMM1	OMM0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 2	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
1	OMM1	LIN Mode Status Monitor	0: The module is in LIN wake-up mode. 1: The module is in LIN operation mode.	R
0	OMM0	LIN Reset Status Monitor	0: The module is in LIN reset mode. 1: The module is not in LIN reset mode.	R

- **OMM1 bit (LIN mode status monitor)**
- **OMM0 bit (LIN reset status monitor)**

  The OMM0 and OMM1 bits indicate the current operation mode.

(19) LIN/UART Status Register (LSTn)

Address: F06D2H

Symbol	7	6	5	4	3	2	1	0
LSTn	HTRC	D1RC	0	0	ERR	0	FRC	FTC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7	HTRC	Successful Header Transmission Flag	0: Header transmission has not been completed. 1: Header transmission has been completed.	R/W
6	D1RC	Successful Data 1 Reception Flag	0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.	R/W
4, 5	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
3	ERR	Error Detection Flag	0: No error has been detected. 1: Error has been detected.	R
2	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
1	FRC	Successful Frame/Wake-up Reception Flag	0: Frame or wake-up reception has not been completed. 1: Frame or wake-up reception has been completed.	R/W
0	FTC	Successful Frame/Wake-up Transmission Flag	0: Frame or wake-up transmission has not been completed. 1: Frame or wake-up 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.

- **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.

- **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.

- **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.

- **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.

- **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.

(20) LIN/UART Error Status Register (LESTn)

Address: F06D3H

Symbol	7	6	5	4	3	2	1	0
LESTn	RPER	0	CSER	0	FER	FTER	PBER	BER
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7	RPER	Response Preparation Error Flag	0: Response preparation error has not been detected. 1: Response preparation error has been detected.	R/W
6	–	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. 1: Checksum error has been detected.	R/W
4	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
3	FER	Framing Error Flag	0: Framing error has not been detected. 1: Framing error has been detected.	R/W
2	FTER	Timeout Error Flag	0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.	R/W
1	PBER	Physical Bus Error Flag	0: Physical bus error has not been detected. 1: Physical bus error has been detected.	R/W
0	BER	Bit Error Flag	0: Bit error has not been detected. 1: Bit 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.

- **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.

- **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.

- **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.

- **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.

  When restarting LIN communication after detecting a frame timeout error or response timeout error, do not enter LIN reset mode. To initialize the LIN module (RLIN3), set the LINnEN bit in the PER2 register to 0, and then write 1 again.

- **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.

- **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.

(21) LIN/UART Data Field Configuration Register (LDFCn)

Address: F06D4H

Symbol	7	6	5	4	3	2	1	0
LDFCn	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
7	LSS	Transmission/Reception Continuation Select	0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Checksum is not included.)	R/W
6	FSM	Frame Separate Mode Select	0: Frame separate mode is not set. 1: Frame separate mode is set.	R/W
5	CSM	Checksum Select	0: Classic checksum mode 1: Enhanced checksum mode	R/W
4	RFT	Response Field Communication Direction Select	0: Reception 1: Transmission	R/W
3 to 0	RFDL[3:0]	Response Field Length Select	0000B: 0 byte (+ checksum) 0001B: 1 byte (+ checksum) 0010B: 2 bytes (+ checksum) : 0111B: 7 bytes (+ checksum) 1000B: 8 bytes (+ checksum) Settings other than the above are prohibited.	R/W

- **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).

- **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 **18.4.3 (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.

- **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 **18.4.6 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.

- **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.

- **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.

(22) LIN/UART ID Buffer Register (LIDBn)

Address: F06D5H

Symbol	7	6	5	4	3	2	1	0
LIDBn	IDP[1:0]		ID[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7, 6	IDP[1:0]	Parity Setting	Sets the parity bits (P) to be transmitted in the ID field.	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

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 **18.6 LIN Self-Test Mode**.

- **IDP[1:0] 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.

- **ID[5:0] bits (ID setting bits)**

  The ID bit sets the 6-bit ID value to be transmitted in the ID field of the LIN frame.

(23) LIN Checksum Buffer Register (LCBRn)

Address: F06D6H

Symbol	7	6	5	4	3	2	1	0
LCBRn								
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 **18.6 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

Symbol	7	6	5	4	3	2	1	0
LDBnm								
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 bit in LDFCn register is 1 (transmission)
- FSM bit 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 bit in LDFCn register is 1 (transmission)
- FSM bit in LDFCn register is 1 (frame separate mode)
- RTS bit 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 up to the byte in which the error was detected are 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 bit in LDFCn register is 1 (transmission)
- FSM bit in LDFCn register is 1 (frame separate mode)
- RTS bit 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 **18.6 LIN Self-Test Mode**.

18.2.2 LIN Registers for Slave Mode

(1) Input Switch Control Register (ISC)

The ISC2, ISC3, and ISC4 bits in the ISC register are used in the LIN/UART module (RLIN3).

Setting bit 2, bit 3, or bit 4 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	ISC4	ISC3	ISC2	0	ISC0

ISC4	Switching inputs for external interrupt INTP6
0	INTP6 pin input signal is set as external interrupt input.
1	LRXD2 pin input signal is set as external interrupt input.

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)

Cautions 1. Bits 7 to 5 and 1 should always be set to 0.

2. Be sure to set the ISC3 and ISC4 bits to 0 in RL78/F22 products.

(2) LIN Channel Select Register (LCHSEL)

Address: F007BH

Symbol	7	6	5	4	3	2	1	0
LCHSEL	0	0	0	0	0	0	LSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 2	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
1, 0	LSEL[1:0]	LIN Channel Select	00B: Selects LIN0. (LIN0 registers can be accessed.) 01B: Selects LIN1. (LIN1 registers can be accessed.) 10B: Selects LIN2. (LIN2 registers can be accessed.) 11B: Setting is prohibited.	R/W

- **LSEL[1:0] bits (LIN channel select bits)**

  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 F06EEH.
  Setting values to the LSEL[1:0] bits map all the registers of the corresponding channel on the register window.
  Setting the LSEL[1:0] bits to 00B maps the LIN0 registers.
  Setting the LSEL[1:0] bits to 01B maps the LIN1 registers.
  Setting the LSEL[1:0] bits to 10B maps the LIN2 registers.

Cautions 1. RL78/F22 products do not have this register.

2. With the RL78/F25 products, set the LSEL[1:0] bits 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.

Writing to the PER2 register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation sets this register to 00H.

Format of Peripheral Enable Register 2 (PER2)

Address: F02C2H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	1	<0>
PER2	0	AAUEN	CAN1EN Note	LIN2EN Note	LIN1EN Note	LIN0EN	0	CAN0EN Note

LIN2EN Note	Control of LIN2 input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by LIN2. • LIN2 is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by LIN2.

LIN1EN Note	Control of LIN1 input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by LIN1. • LIN1 is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by LIN1.

LIN0EN	Control of LIN0 input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by LIN0. • LIN0 is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by LIN0.

Note Only in the RL78/F25 products.

Caution Be sure to clear the following bits to 0.

- **Bits 0, 1, 3, 4, 5, and 7 in the RL78/F22 products**
- **Bits 1 and 7 in the RL78/F25 products**

(4) 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.

Format of LIN Clock Select Register (LINCKSEL)

Address: F02C4H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	3	<2>	<1>	<0>
LINCKSEL	0	LIN2MCKE Note	LIN1MCKE Note	LIN0MCKE	0	LIN2MCK Note	LIN1MCK Note	LIN0MCK

LIN2MCKE Note	Control of supplying or stopping LIN2 communication clock source
0	Stops LIN2 communication clock source supply
1	Enables LIN2 communication clock source supply

LIN1MCKE Note	Control of supplying or stopping LIN1 communication clock source
0	Stops LIN1 communication clock source supply
1	Enables LIN1 communication clock source supply

LIN0MCKE	Control of supplying or stopping LIN0 communication clock source
0	Stops LIN0 communication clock source supply
1	Enables LIN0 communication clock source supply

LIN2MCK Note	Control of selecting LIN2 communication clock source
0	Selects the f_{CLK} clock
1	Selects the f_{MX} clock

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/F25.

Cautions 1. Select the LINn (n = 0 to 2) operating clock with the LINnMCK bit before setting the LINnMCKE bit to 1.

2. When operating LINn in SNOOZE mode, set the LINnMCK bit to 0.

3. In case of LINnMCK is set to 1 (select f_{MX} clock), 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) External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)

For details, see section **22.3.4 External Interrupt Rising/Falling Edge Enable registers (EGP0, EGP1, EGN0, EGN1)** in **CHAPTER 22 INTERRUPT FUNCTIONS.**

(6) LIN Wake-up Baud Rate Select Register (LWBRn)

Address: F06C1H

Symbol	7	6	5	4	3	2	1	0
LWBRn	NSPB[3:0]				LPRS[2:0]			0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 4	NSPB[3:0]	Bit Sampling Count Select	0000B: 16 sampling 0011B: 4 sampling 0111B: 8 sampling 1111B: 16 sampling Settings other than the above are prohibited.	R/W
3 to 1	LPRS[2:0]	Prescaler Clock Select	000B: 1/1 001B: 1/2 010B: 1/4 011B: 1/8 100B: 1/16 101B: 1/32 110B: 1/64 111B: 1/128	R/W
0	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

Set the LWBRn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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).

- **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).

(7) LIN/UART Baud Rate Prescaler Register (LBRPn)

Address: F06C3H, F06C2H

	LBRPn1								LBRPn0							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LBRPn																
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

Symbol	7	6	5	4	3	2	1	0
LSTCn								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 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 **18.6 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 **18.6 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

Symbol	7	6	5	4	3	2	1	0
LMDn	0	0	LRDNFS	LIOS	0	0	LMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7, 6	–	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. 1: The noise filter is disabled.	R/W
4	LIOS	LIN Interrupt Output Select	0: LIN interrupt is used. 1: Transmission interrupt, successful reception interrupt, and reception status interrupt are used.	R/W
3, 2	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
1, 0	LMD[1:0]	LIN/UART Mode Select	10B: LIN slave mode (auto baud rate) 11B: LIN slave mode (fixed baud rate)	R/W

Set the LMDn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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.

- **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 **18.9 Interrupts**.

- **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.

(10) LIN Break Field Configuration Register/UART Configuration Register (LBFCn)

Address: F06C9H

Symbol	7	6	5	4	3	2	1	0
LBFCn	0	0	0	0	0	0	0	BLT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 1	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
0	BLT	Reception Break (Low) Width Select	0: Reception break (low width) of 9.5/10 or more Tbits is detected. 1: Reception break (low width) of 10.5/11 or more Tbits is detected.	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

Symbol	7	6	5	4	3	2	1	0
LSCn	0	0	IBS[1:0]		0	RS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7, 6	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
5, 4	IBS[1:0]	Inter-Byte Space Select	00B: 0 Tbit 01B: 1 Tbit 10B: 2 Tbits 11B: 3 Tbits	R/W
3	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
2 to 0	RS[2:0]	Response Space Select	000B: 0 Tbit 001B: 1 Tbit 010B: 2 Tbits 011B: 3 Tbits 100B: 4 Tbits 101B: 5 Tbits 110B: 6 Tbits 111B: 7 Tbits	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.

- **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.

- **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.

(12) LIN Wake-up Configuration Register (LWUPn)

Address: F06CBH

Symbol	7	6	5	4	3	2	1	0
LWUPn	WUTL[3:0]				0	0	0	0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 4	WUTL[3:0]	Wake-up Transmission Low Width Select	0000B: 1 Tbit 0001B: 2 Tbits 0010B: 3 Tbits 0011B: 4 Tbits : 1100B: 13 Tbits 1101B: 14 Tbits 1110B: 15 Tbits 1111B: 16 Tbits	R/W
3 to 0	–	Reserved	These bits are always read as 0. The write value should always be 0.	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

Symbol	7	6	5	4	3	2	1	0
LIEn	0	0	0	0	SHIE	ERRIE	FRCIE	FTCIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 4	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
3	SHIE	Successful Header Reception Interrupt Enable	0: Disables successful header reception interrupt. 1: Enables successful header reception interrupt.	R/W
2	ERRIE	Error Detection Interrupt Enable	0: Disables error detection interrupt. 1: Enables error detection interrupt.	R/W
1	FRCIE	Successful Frame/Wake-up Reception Interrupt Enable	0: Disables successful response/wake-up reception interrupt. 1: Enables successful response/wake-up reception interrupt.	R/W
0	FTCIE	Successful Frame/Wake-up Transmission Interrupt Enable	0: Disables successful response/wake-up transmission interrupt. 1: Enables successful response/wake-up transmission interrupt.	R/W

Set the LIEn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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.

- **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.

- **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.

- **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.

(14) LIN/UART Error Detection Enable Register (LEDEn)

Address: F06CDH

Symbol	7	6	5	4	3	2	1	0
LEDEn	LTES	IPERE	0	SFERE	FERE	TERE	0	BERE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7	LTES	Timeout Error Select	0: Frame timeout error 1: Response timeout error	R/W
6	IPERE	ID Parity Error Detection Enable	0: Disables ID parity error detection. 1: Enables ID parity error detection.	R/W
5	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
4	SFERE	Sync Field Error Detection Enable	0: Disables sync field error detection. 1: Enables sync field error detection.	R/W
3	FERE	Framing Error Detection Enable	0: Disables framing error detection. 1: Enables framing error detection.	R/W
2	TERE	Timeout Error Detection Enable	0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.	R/W
1	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
0	BERE	Bit Error Detection Enable	0: Disables bit error detection. 1: Enables bit error detection.	R/W

Set the LEDEn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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 **18.4.6 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 **18.4.6 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 **18.4.6 Error Status**.

- **FERE bit (framing error detection enable bit)**

  The FERE bit enables or disables detection of the framing error.
  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 **18.4.6 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 **18.4.6 Error Status**.

- **BERE bit (bit error detection enable bit)**

  The BERE bit enables or disables detection of the bit error.
  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 **18.4.6 Error Status**.

(15) LIN/UART Control Register (LCUCn)

Address: F06CEH

Symbol	7	6	5	4	3	2	1	0
LCUCn	0	0	0	0	0	0	OM1	OM0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 2	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
1	OM1	LIN Mode Select	0: LIN wake-up mode is caused. 1: LIN operation mode is caused.	R/W
0	OM0	LIN Reset	0: LIN reset mode is caused. 1: LIN reset mode is canceled.	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.

- **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 (Header reception or wake-up transmission/reception is started).

- **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.

Caution When transitioning from LIN mode with timeout error detection enabled (TERE bit of LEDEn register is set to 1) to LIN reset mode and then to LIN mode, clear the error flag according to the procedure described in "18.3.1 LIN Reset Mode".

(16) LIN/UART Transmission Control Register (LTRCn)

Address: F06D0H

Symbol	7	6	5	4	3	2	1	0
LTRCn	0	0	0	0	0	LNRR	RTS	FTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 3	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
2	LNRR	No-Response Request	0: Response to the reception ID is received/transmitted. 1: Response to the reception ID is not received/transmitted.	R/W
1	RTS	Response Transmission/ Reception Start	0: Response transmission/reception is stopped. 1: Response transmission/reception is started.	R/W
0	FTS	LIN Communication Start	0: Header reception or wake-up transmission/reception is stopped. 1: Header reception or wake-up transmission/reception is started.	R/W

- **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).

- **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.

- **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.

(17) LIN/UART Mode Status Register (LMSTn)

Address: F06D1H

Symbol	7	6	5	4	3	2	1	0
LMSTn	0	0	0	0	0	0	OMM1	OMM0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 2	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
1	OMM1	LIN Mode Status Monitor	0: The module is in LIN wake-up mode. 1: The module is in LIN operation mode.	R
0	OMM0	LIN Reset Status Monitor	0: The module is in LIN reset mode. 1: The module is not in LIN reset mode.	R

- **OMM1 bit (LIN mode status monitor)**
- **OMM0 bit (LIN reset status monitor)**

  The OMM0 and OMM1 bits indicate the current operation mode.

(18) LIN/UART Status Register (LSTn)

Address: F06D2H

Symbol	7	6	5	4	3	2	1	0
LSTn	HTRC	D1RC	0	0	ERR	0	FRC	FTC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7	HTRC	Successful Header Reception Flag	0: Header reception has not been completed. 1: Header reception has been completed.	R/W
6	D1RC	Successful Data 1 Reception Flag	0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.	R/W
5, 4	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
3	ERR	Error Detection Flag	0: No error has been detected. 1: Error has been detected.	R
2	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
1	FRC	Successful Frame/Wake-up Reception Flag	0: Response or wake-up reception has not been completed. 1: Response or wake-up reception has been completed.	R/W
0	FTC	Successful Frame/Wake-up Transmission Flag	0: Response or wake-up transmission has not been completed. 1: Response or wake-up transmission 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.

- **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.

- **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.

- **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.

- **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.

- **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.

(19) LIN/UART Error Status Register (LESTn)

Address: F06D3H

Symbol	7	6	5	4	3	2	1	0
LESTn	RPER	IPER	CSER	SFER	FER	TER	0	BER
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7	RPER	Response Preparation Error Flag	0: Response preparation error has not been detected. 1: Response preparation error has been detected.	R/W
6	IPER	ID Parity Error Flag	0: ID parity error has not been detected. 1: ID parity error has been detected.	R/W
5	CSER	Checksum Error Flag	0: Checksum error has not been detected. 1: Checksum error has been detected.	R/W
4	SFER	Sync Field Error Flag	0: Sync field error has not been detected. 1: Sync field error has been detected.	R/W
3	FER	Framing Error Flag	0: Framing error has not been detected. 1: Framing error has been detected.	R/W
2	TER	Timeout Error Flag	0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.	R/W
1	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
0	BER	Bit Error Flag	0: Bit error has not been detected. 1: Bit 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.

- **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.

- **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.

- **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.

- **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.

- **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.

- **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.

- **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.

(20) LIN/UART Data Field Configuration Register (LDFCn)

Address: F06D4H

Symbol	7	6	5	4	3	2	1	0
LDFCn	LSS	0	LCS	RCDS	RFDL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7	LSS	Transmission/Reception Continuation Select	0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Data transmission/reception is continued without waiting for the next header reception.)	R/W
6	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
5	LCS	Checksum Select	0: Classic checksum mode 1: Enhanced checksum mode	R/W
4	RCDS	Response Field Communication Direction Select	0: Reception 1: Transmission	R/W
3 to 0	RFDL[3:0]	Response Field Length Select	0000B: 0 byte (+ checksum) 0001B: 1 byte (+ checksum) 0010B: 2 bytes (+ checksum) : 0111B: 7 bytes (+ checksum) 1000B: 8 bytes (+ checksum) Settings other than the above are prohibited.	R/W

- **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).

- **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 **18.4.6 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).

- **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.

- **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).

  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.

(21) LIN/UART ID Buffer Register (LIDBn)

Address: F06D5H

Symbol	7	6	5	4	3	2	1	0
LIDBn	IDP[1:0]		ID[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7, 6	IDP[1:0]	Parity	Holds the parity bits (P) received in the ID field.	R/W
5 to 0	ID[5:0]	ID	Holds the 6-bit ID value 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 **18.6 LIN Self-Test Mode**.

- **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.

- **ID[5:0] bits (ID bits)**

  The ID bits hold the 6-bit ID value received in the ID field of the LIN frame.

(22) LIN Checksum Buffer Register (LCBRn)

Address: F06D6H

Symbol	7	6	5	4	3	2	1	0
LCBRn								
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 **18.6 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

Symbol	7	6	5	4	3	2	1	0
LDBnm								
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 of the LTRCn register 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 up to the byte in which the error was detected are 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 **18.6 LIN Self-Test Mode**.

(24) LIN Break and Sync Field Detection Status Register (LBSSn)

Address: F06ECH

Symbol	7	6	5	4	3	2	1	0
LBSSn	0	0	0	0	0	0	SYCC	BRKC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 2	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
1	SYCC	Sync Field Detection Flag	0: Not detect sync field signal 1: Detected sync field signal	R/W
0	BRKC	Break Field Detection Flag	0: Not detect break field signal 1: Detected break field signal	R/W

- **SYCC Bit (Sync Field Detection Flag)**

  Users can not write to this bit if the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
  This bit is cleared by writing “0” to it.
  The SYCC bit is cleared when the LIN/UART module (RLIN3) is transitioning to the reset mode.
  This bit is set when Sync field is successfully reception in LIN Slave mode.

- **BRKC Bit (Break Field Detection Flag)**

  Users can not write to this bit if the OMM0 bit in the LMSTn register is 0 (LIN reset mode).
  This bit is cleared by writing “0” to it.
  The BRKC bit is cleared when the LIN/UART module (RLIN3) is transitioning to the reset mode.
  This bit is set when Break field is successfully reception in LIN Slave mode.

(25) LIN Response Space Dominant Signal Detection Status Register (LRSSn)

Address: F06EEH

Symbol	7	6	5	4	3	2	1	0
LRSSn	0	0	0	0	0	0	0	RSDD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 1	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
0	RSDD	Response Space Dominant Signal Detection Flag	0: Not detect dominant level in response space 1: Detected dominant level in response space	R

- **RSDD bit (Response Space Dominant Signal Detection Flag)**

  Users can not write to this bit.

  This bit is valid only for LIN response transmission in LIN slave mode.

  This bit is invalid when a bit error is occurred in the transmission of the response space.

  This bit is cleared when Sync field is received.

  This bit is cleared when the LIN/UART module (RLIN3) is transitioning to the reset mode.

  This bit is set when a dominant level of 0.5 Tbit or more is detected from the completion of the header reception (the stop bit of the ID field) to the start of transmission **Note** in LIN slave mode.

Note When LSCn.RS[2:0] = 000B, it is the transmission of the start bit of the 1st Data byte.
When LSCn.RS[2:0] = 001B to 111B, it is the transmission of the response space.

18.2.3 Registers for UART

(1) Input Switch Control Register (ISC)

The ISC2, ISC3, and ISC4 bits in the ISC register are used in the LIN/UART module (RLIN3).

Setting bit 2, bit 3, or bit 4 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	ISC4	ISC3	ISC2	0	ISC0

ISC4	Switching inputs for external interrupt INTP6
0	INTP6 pin input signal is set as external interrupt input.
1	LRXD2 pin input signal is set as external interrupt input.

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)

Cautions 1. Bits 7 to 5 and 1 should always be set to 0.

2. Be sure to set the ISC3 and ISC4 bits to 0 in RL78/F22 products.

(2) LIN Channel Select Register (LCHSEL)

Address: F007BH

Symbol	7	6	5	4	3	2	1	0
LCHSEL	0	0	0	0	0	0	LSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 2	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
1, 0	LSEL[1:0]	LIN Channel Select	00B: Selects LIN0. (LIN0 registers can be accessed.) 01B: Selects LIN1. (LIN1 registers can be accessed.) 10B: Selects LIN2. (LIN2 registers can be accessed.) 11B: Setting is prohibited.	R/W

- **LSEL[1:0] bits (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 F06EEH.
  Setting values to the LSEL[1:0] bits map all the registers of the corresponding channel on the register window.
  Setting the LSEL[1:0] bits to 00B maps the LIN0 registers.
  Setting the LSEL[1:0] bits to 01B maps the LIN1 registers.
  Setting the LSEL[1:0] bits to 10B maps the LIN2 registers.

Cautions 1. RL78/F22 products do not have this register.

2. With the RL78/F25 products, set the LSEL[1:0] bits 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.

Writing to the PER2 register is disabled when the GCSC bit of the IAWCTL register is set to 1.

Reset signal generation sets this register to 00H.

Format of Peripheral Enable Register 2 (PER2)

Address: F02C2H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	1	<0>
PER2	0	AAUEN	CAN1EN Note	LIN2EN Note	LIN1EN Note	LIN0EN	0	CAN0EN Note

LIN2EN Note	Control of LIN2 input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by LIN2. • LIN2 is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by LIN2.

LIN1EN Note	Control of LIN1 input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by LIN1. • LIN1 is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by LIN1.

LIN0EN	Control of LIN0 input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by LIN0. • LIN0 is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by LIN0.

Note Only in the RL78/F25 products.

Caution Be sure to clear the following bits to 0.

- **Bits 0, 1, 3, 4, 5, and 7 in the RL78/F22 products**
- **Bits 1 and 7 in the RL78/F25 products**

(4) 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.

Format of LIN Clock Select Register (LINCKSEL)

Address: F02C4H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	3	<2>	<1>	<0>
LINCKSEL	0	LIN2MCKE Note	LIN1MCKE Note	LIN0MCKE	0	LIN2MCK Note	LIN1MCK Note	LIN0MCK

LIN2MCKE Note	Control of supplying or stopping LIN2 communication clock source
0	Stops LIN2 communication clock source supply
1	Enables LIN2 communication clock source supply

LIN1MCKE Note	Control of supplying or stopping LIN1 communication clock source
0	Stops LIN1 communication clock source supply
1	Enables LIN1 communication clock source supply

LIN0MCKE	Control of supplying or stopping LIN0 communication clock source
0	Stops LIN0 communication clock source supply
1	Enables LIN0 communication clock source supply

LIN2MCK Note	Control of selecting LIN2 communication clock source
0	Selects the f_{CLK} clock
1	Selects the f_{MX} clock

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/F25.

Cautions 1. Select the LINn (n = 0 to 2) operating clock with the LINnMCK bit before setting the LINnMCKE bit to 1.

2. When operating LINn in SNOOZE mode, set the LINnMCK bit to 0.

3. In case of LINnMCK is set to 1 (select f_{MX} clock), 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) External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)

For details, see section **22.3.4 External Interrupt Rising/Falling Edge Enable registers (EGP0, EGP1, EGN0, EGN1)** in **CHAPTER 22 INTERRUPT FUNCTIONS**.

(6) LIN Wake-up Baud Rate Select Register (LWBRn)

Address: F06C1H

Symbol	7	6	5	4	3	2	1	0
LWBRn	NSPB[3:0]				LPRS[2:0]			0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 4	NSPB[3:0]	Bit Sampling Count Select	0000B: 16 sampling 0101B: 6 sampling 0110B: 7 sampling 0111B: 8 sampling 1000B: 9 sampling 1001B: 10 sampling 1010B: 11 sampling 1011B: 12 sampling 1100B: 13 sampling 1101B: 14 sampling 1110B: 15 sampling 1111B: 16 sampling Settings other than the above are prohibited.	R/W
3 to 1	LPRS[2:0]	Prescaler Clock Select	000B: 1/1 001B: 1/2 010B: 1/4 011B: 1/8 100B: 1/16 101B: 1/32 110B: 1/64 111B: 1/128	R/W
0	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W

Set the LWBRn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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.

- **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.

(7) LIN/UART Baud Rate Prescaler Register (LBRPn)

Address: F06C3H, F06C2H

	LBRPn1								LBRPn0							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LBRPn																
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

Symbol	7	6	5	4	3	2	1	0
LUSCn	0	0	0	0	0	URDCC	USEC	UWC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 3	–	Reserved	These bits are always read as 0. The write value should always be 0.	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. 1: Enables comparison of the received data and the LIDBn register value in SNOOZE mode.	R/W
1	USEC	UART Standby Error Control	0: Enables error detection interrupt generation. 1: Disables error detection interrupt generation.	R/W
0	UWC	UART Standby Wake-up Control	0: Disables start of reception from STOP mode. 1: Enables start of reception from STOP mode.	R/W

Set the LUSCn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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).

- **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).

- **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.

(9) LIN/UART Mode Register (LMDn)

Address: F06C8H

Symbol	7	6	5	4	3	2	1	0
LMDn	0	0	LRDNFS	0	0	0	LMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7, 6	–	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. 1: The noise filter is disabled.	R/W
4 to 2	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
1, 0	LMD[1:0]	LIN/UART Mode Select	01B: UART mode	R/W

Set the LMDn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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.

- **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.

(10) LIN Break Field Configuration Register/UART Configuration Register (LBFCn)

Address: F06C9H

Symbol	7	6	5	4	3	2	1	0
LBFCn	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
7	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
6	UTPS	UART Output Polarity Select	0: Transmission data is output as is. 1: Transmission data is reversed before being output.	R/W
5	URPS	UART Input Polarity Select	0: Reception data is input as is. 1: Reception data is reversed before being input.	R/W
4, 3	UPS[1:0]	UART Parity Select	00B: No parity 01B: Even parity 10B: 0-parity 11B: Odd parity	R/W
2	USBLS	UART Stop Bit Length Select	0: One stop bit 1: Two stop bits	R/W
1	UBOS	UART Transfer Format Select	0: LSB first 1: MSB first	R/W
0	UBLS	UART Character Length Select	0: 8-bit UART communication 1: 7-bit UART communication	R/W

Set the LBFCn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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.

- **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.

- **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.

- **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.

- **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.

- **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).

(11) LIN/UART Space Configuration Register (LSCn)

Address: F06CAH

Symbol	7	6	5	4	3	2	1	0
LSCn	0	0	IBS[1:0]		0	0	0	0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7, 6	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
5, 4	IBS[1:0]	Inter-Byte Space Select	00B: 0 Tbit 01B: 1 Tbit 10B: 2 Tbits 11B: 3 Tbits	R/W
3 to 0	–	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.
  When transmitting from the transmission buffer (LUTDRn register) and the wait transmission buffer (LUWTDRn register), set 00B to the IBS[1:0] bits.

(12) LIN/UART Error Detection Enable Register (LEDEn)

Address: F06CDH

Symbol	7	6	5	4	3	2	1	0
LEDEn	0	0	0	0	FERE	OERE	0	BERE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 4	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
3	FERE	Framing Error Detection Enable	0: Disables framing error detection. 1: Enables framing error detection.	R/W
2	OERE	Overrun Error Detection Enable	0: Disables overrun error detection. 1: Enables overrun error detection.	R/W
1	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
0	BERE	Bit Error Detection Enable	0: Disables bit error detection. 1: Enables bit error detection.	R/W

Set the LEDEn register when the OMM0 bit in the LMSTn register is 0 (LIN reset mode).

- **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 **18.5.5 Error Status**.

- **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 **18.5.5 Error Status**.

- **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 **18.5.5 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).

(13) LIN/UART Control Register (LCUCn)

Address: F06CEH

Symbol	7	6	5	4	3	2	1	0
LCUCn	0	0	0	0	0	0	0	OM0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 1	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
0	OM0	LIN Reset	0: LIN reset mode is caused. 1: LIN reset mode is canceled.	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

Symbol	7	6	5	4	3	2	1	0
LTRCn	0	0	0	0	0	0	RTS	0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 2	–	Reserved	These bits are 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. 1: UART buffer transmission is enabled.	R/W
0	–	Reserved	This bit is 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

Symbol	7	6	5	4	3	2	1	0
LMSTn	0	0	0	0	0	0	0	OMM0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 1	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
0	OMM0	LIN Reset Status Monitor	0: The module is in LIN reset mode. 1: The module is not in LIN reset mode.	R

- **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

Symbol	7	6	5	4	3	2	1	0
LSTn	0	0	URS	UTS	ERR	0	0	FTC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7, 6	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
5	URS	Reception Status Flag	0: The LIN/UART module is not receiving data. 1: The LIN/UART module is receiving data.	R
4	UTS	Transmission Status Flag	0: The LIN/UART module is not transmitting data. 1: The LIN/UART module is transmitting data.	R
3	ERR	Error Detection Flag	0: No error has been detected. 1: Error has been detected.	R
2, 1	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
0	FTC	Successful Buffer Transmission Flag	0: Data transmission from the UART buffer has not been completed. 1: Data transmission from the UART buffer 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.

- **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.

- **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).

- **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.

- **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.

(17) LIN/UART Error Status Register (LESTn)

Address: F06D3H

Symbol	7	6	5	4	3	2	1	0
LESTn	0	UPER	IDMT	EXBT	FER	OER	0	BER
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
6	UPER	Parity Error Flag	0: Parity error has not been detected. 1: Parity error has been detected.	R/W
5	IDMT	ID match flag	0: Received data does not agree with the ID value. 1: Received data agrees with the ID value.	R/W
4	EXBT	Expansion Bit Detection Flag	0: Expansion bit has not been detected. 1: Expansion bit has been detected.	R/W
3	FER	Framing Error Flag	0: Framing error has not been detected. 1: Framing error has been detected.	R/W
2	OER	Overrun Error Flag	0: Overrun error has not been detected. 1: Overrun error has been detected.	R/W
1	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
0	BER	Bit Error Flag	0: Bit error has not been detected. 1: Bit 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.

- **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.

- **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.

- **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.

- **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.

- **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.

- **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.

(18) LIN/UART Data Field Configuration Register (LDFCn)

Address: F06D4H

Symbol	7	6	5	4	3	2	1	0
LDFCn	0	0	UTSW	0	MDL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7, 6	–	Reserved	These bits are 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. 1: When UART buffer transmission is requested, transmission is started after waiting for stop bit reception to be completed.	R/W
4	–	Reserved	This bit is always read as 0. The write value should always be 0.	R/W
3 to 0	MDL[3:0]	UART Buffer Data Length Select	0000B: 9 data 0001B: 1 data 0010B: 2 data 0011B: 3 data 0100B: 4 data 0101B: 5 data 0110B: 6 data 0111B: 7 data 1000B: 8 data 1001B: 9 data Settings other than the above are prohibited.	R/W

- **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.

- **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).

(19) LIN/UART ID Buffer Register (LIDBn)

Address: F06D5H

Symbol	7	6	5	4	3	2	1	0
LIDBn								
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

Symbol	7	6	5	4	3	2	1	0
LUDBn0								
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 of the LTRCn register is 0 (UART buffer transmission is disabled).

For details of the UART buffer, refer to **18.5.3 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

Symbol	7	6	5	4	3	2	1	0
LDBnm								
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 of the LTRCn register is 0 (UART buffer transmission is disabled).

For details of the UART buffer, refer to **18.5.3 Buffer Processing of Transmission Data**.

(22) UART Operation Enable Register (LUOERn)

Address: F06E0H

Symbol	7	6	5	4	3	2	1	0
LUOERn	0	0	0	0	0	0	UROE	UTOE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 2	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
1	UROE	Reception Enable	0: Disables reception. 1: Enables reception.	R/W
0	UTOE	Transmission Enable	0: Disables transmission. 1: Enables transmission.	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.

- **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.
  This bit must be 0 during transmitting data from UART buffer.

- **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.

(23) UART Option Register 1 (LUORn1)

Address: F06E1H

Symbol	7	6	5	4	3	2	1	0
LUORn1	0	0	0	UECD	UTIGTS	UEBDCE	UEBDL	UEBE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
7 to 5	–	Reserved	These bits are always read as 0. The write value should always be 0.	R/W
4	UECD	Expansion Bit Comparison Disable	0: Enables comparison between the received expansion bit and the UEBDL bit value. 1: Disables comparison between the received expansion bit and the UEBDL bit value.	R/W
3	UTIGTS	Transmission Interrupt Generation Timing Select	0: Transmission interrupt is generated at the start of transmission. 1: Transmission interrupt is generated at the completion of transmission.	R/W
2	UEBDCE	Expansion Bit/Data Comparison Enable	0: Disables data comparison after detection of the expansion bit. 1: Enables data comparison after detection of the expansion bit.	R/W
1	UEBDL	Expansion Bit Detection Level Select	0: Selects expansion bit value 0 as the expansion bit detection level. 1: Selects expansion bit value 1 as the expansion bit detection level.	R/W
0	UEBE	Expansion Bit Enable	0: Disables expansion bit operation. 1: Enables expansion bit operation.	R/W

- **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).

- **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.

- **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).

- **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.

- **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.

(24) UART Transmission Data Register (LUTDRn)

Address: F06E5H, F06E4H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUTDRn	0	0	0	0	0	0	0	[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Bit name	Function	Setting Range	R/W
15 to 9	Reserved	These bits are always read as 0. The write value should always be 0.	–	R/W
8 to 0	–	Sets the data to be transmitted from the transmission buffer.	000H to 1FFH	R/W

Sets the data to be transmitted to the LUTDRn register.

Writing data to this register with the UTOE bit in the LUOERn register set to 1 (enables transmission) 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.

Item	LUTDRn								
	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

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LURDRn	0	0	0	0	0	0	0	[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Bit name	Function	Setting Range	R/W
15 to 9	Reserved	These bits are always read as 0. The write value should always be 0.	–	R/W
8 to 0	–	Allows the reception data to be read from the reception buffer.	000H to 1FFH	R

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.

Item	LURDRn								
	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

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LUWTDRn	0	0	0	0	0	0	0	[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Bit name	Function	Setting Range	R/W
15 to 9	Reserved	These bits are always read as 0. The write value should always be 0.	–	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

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 to switch from reception to transmission in half-duplex communication.
If this register is written while the stop bit is being received, transmission will start after the stop bit has been 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.

Item	LUWTDRn								
	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

18.3 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 18-3 shows mode transitions. **Table 18-4** describes mode transition conditions. **Table 18-5** lists operations available in each mode.

Figure 18-3. Mode Transitions

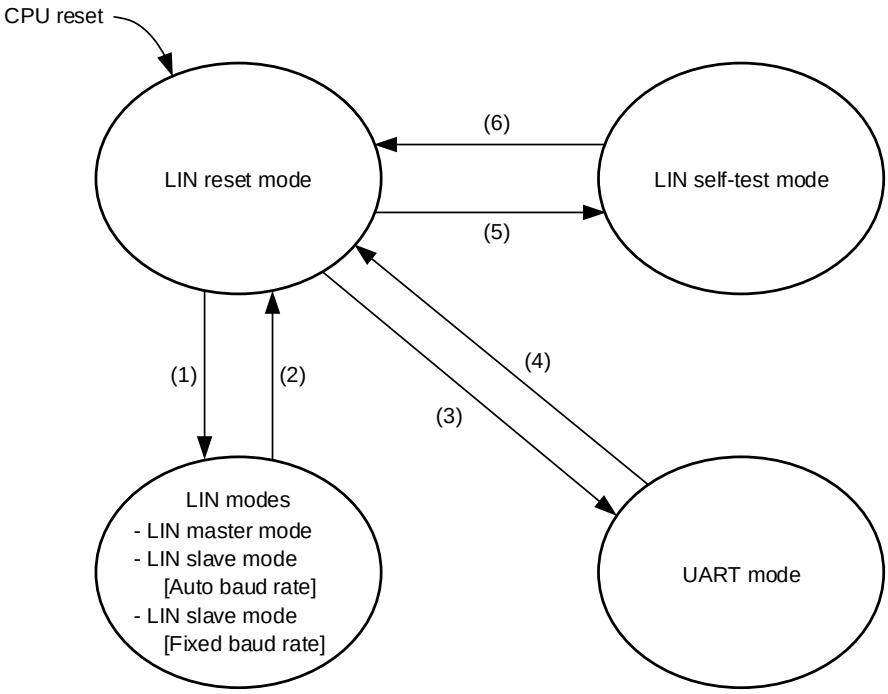


Table 18-4. Mode Transition Conditions

Step	Mode transition			Transition condition
(1)	LIN reset mode	→	LIN mode (LIN master mode)	LMD bits in LMDn register = 00B and OM1 and OM0 bits in LCUCn register = 01B or 11B
	LIN reset mode	→	LIN mode (LIN slave mode[auto baud rate])	LMD bits in LMDn register = 10B and OM1 and OM0 bits in LCUCn register = 01B or 11B
	LIN reset mode	→	LIN mode (LIN slave mode [fixed baud rate])	LMD bits in LMDn register = 11B and OM1 and OM0 bits in LCUCn register = 01B or 11B
(2)	LIN mode	→	LIN reset mode	OM0 bit in LCUCn register = 0
(3)	LIN reset mode	→	UART mode	LMD bits in LMDn register = 01B and OM0 bit in LCUCn register = 1
(4)	UART mode	→	LIN reset mode	OM0 bit in LCUCn register = 0
(5)	LIN reset mode	→	LIN self-test mode	See **18.6 LIN Self-Test Mode**.
(6)	LIN self-test mode	→	LIN reset mode	See **18.6 LIN Self-Test Mode**.

Table 18-5. Operations Available in Each Mode

LIN mode		UART mode	LIN self-test mode
LIN master mode	LIN slave mode [auto baud rate] / LIN slave mode [fixed baud rate]		
Header transmission Response transmission Response reception Wake-up transmission Wake-up reception Error detection	Header reception Response transmission Response reception Wake-up transmission Wake-up reception Error detection	UART transmission UART reception 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 **18.6 LIN Self-Test Mode**.

18.3.1 LIN Reset Mode

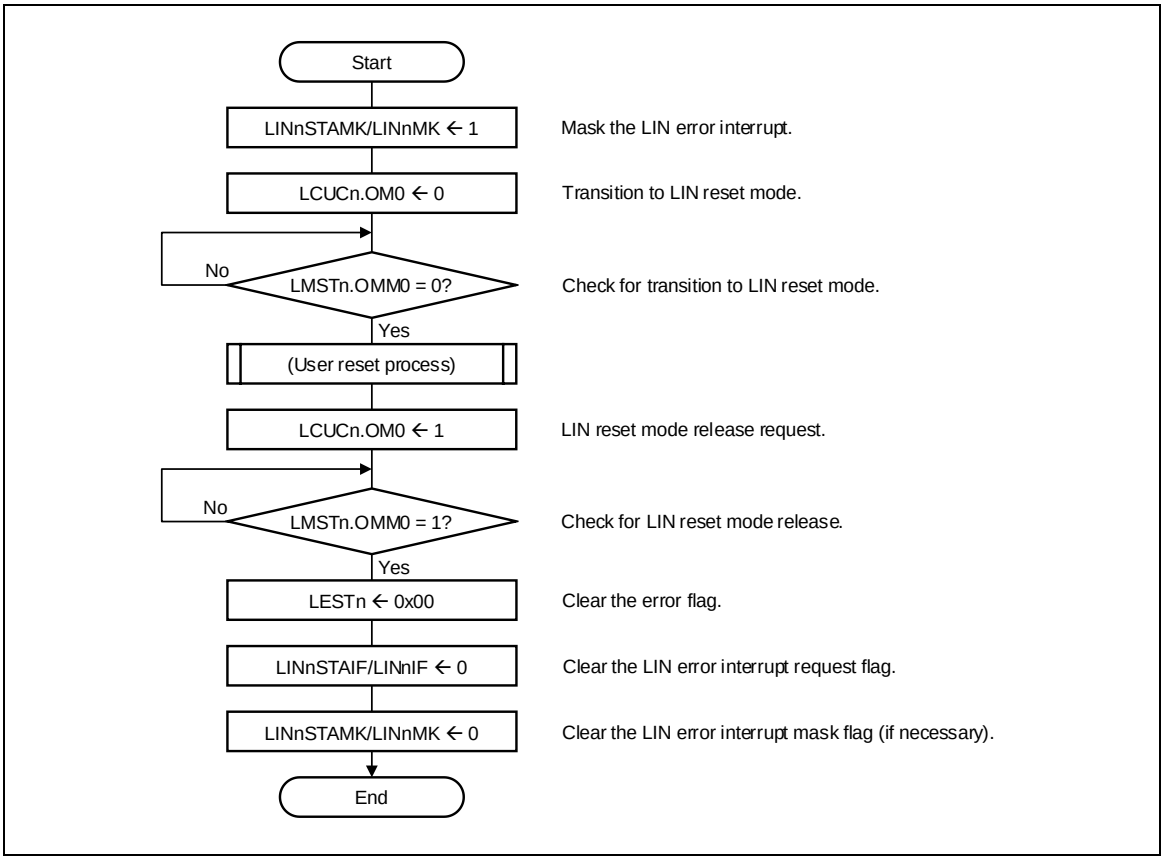
Setting the OM0 bit in the LCUCn register to 0 (LIN reset mode) causes a transition to LIN reset mode. The change to LIN reset mode can be verified by determining that the OMM0 bit in the LMSTn register has been set to 0 (LIN reset mode). In this mode, the LIN communication and the UART communication functions are all halted, and f_{LIN} also stops.

From LIN reset mode, transitions to LIN mode, UART mode, and LIN self-test mode can be made.

- The following registers are initialized to their respective reset values after a transition to the LIN reset mode, and retain their initial values during the LIN reset mode:
  - LTRCn register
  - LSTn register
  - LESTn register
  - LUOERn register
  - LBSSn register
  - LRSSn register

- The following registers retain their previous values after a transition to the LIN reset mode:
  - 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
  - LUORn1 register
  - LUTDRn register
  - LURDRn register
  - LUWTDRn register

Caution When transitioning from LIN mode with timeout error detection enabled (FTERE bit or TERE bit of LEDEn register is set to 1) to LIN reset mode and then to LIN mode, clear the error flags using the procedure shown in the figure below. If LINnEN bit of PER2 register is set to 0 to transition to LIN mode, it is not necessary to clear the error flag.



18.3.2 LIN Mode

LIN mode can operate in the following sub-modes: 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 sub-mode to another sub-mode 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 18-4 shows the transition of operation modes. **Table 18-6** describes the transition conditions of operation modes.

Figure 18-4. Transition of Operation Modes

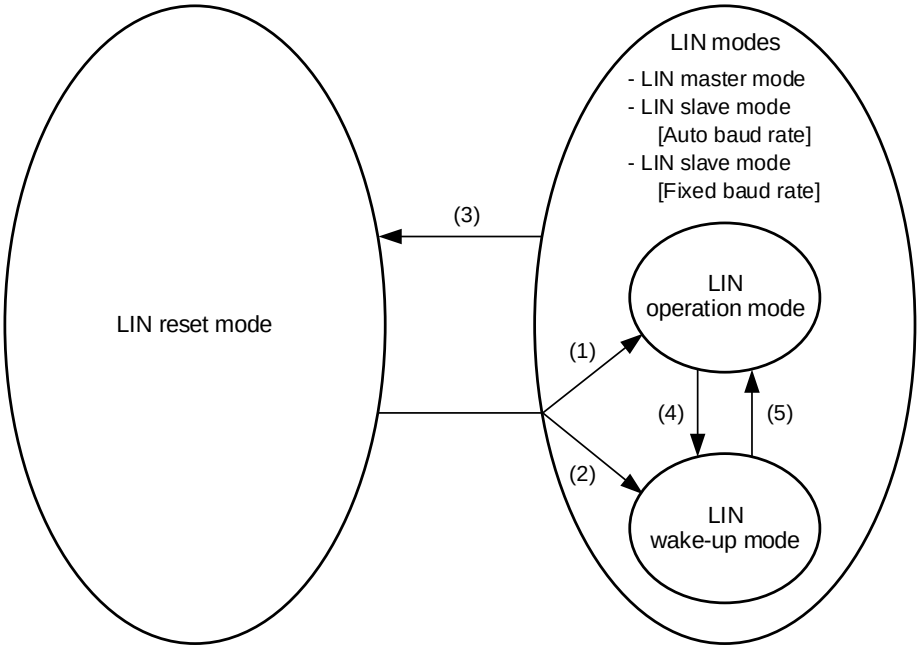


Table 18-6. Transition Conditions of Operation Modes

Step	Operation mode transition			Transition condition
(1)	LIN reset mode	→	LIN mode - LIN operation mode	LMD bits in LMDn register = 00B or 10B or 11B and OM1 and OM0 bits in LCUCn register = 11B
(2)	LIN reset mode	→	LIN mode - LIN wake-up mode	LMD bits in LMDn register = 00B or 10B or 11B and OM1 and OM0 bits in LCUCn register = 01B
(3) Note 1	LIN mode - LIN operation mode - LIN wake-up mode	→	LIN reset mode	OM0 bit in LCUCn register = 0
(4) Note 2	LIN mode - LIN operation mode	→	LIN mode - LIN wake-up mode	OM1 and OM0 bits in LCUCn register = 01B
(5) Note 2	LIN mode - LIN wake-up mode	→	LIN mode - 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.

18.3.3 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 1 changes the mode to UART mode, turning the OMM0 bit in the LMSTn register to 1. Communication settings should be performed after the LMSTn register has become 01B.

18.3.4 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 **18.6 LIN Self-Test Mode**.

18.4 LIN Mode

18.4.1 Operation Overview

(1) LIN Master Mode

(a) Header Transmission

Figure 18-5 shows the operation of the LIN/UART module (LIN master mode) in header transmission. **Table 18-7** provides processing in header transmission.

Figure 18-5. Operation in Header Transmission

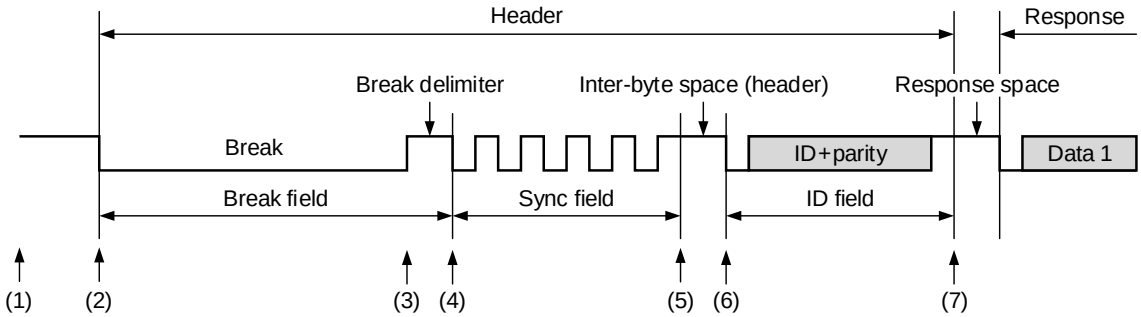


Table 18-7. Processing in Header Transmission

Step	Software processing	LIN/UART module processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Changes the LIN/UART module to the LIN master mode: LIN operation mode • 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 **18.4.6 Error Status**.

(b) Response Transmission

Figure 18-6 shows the operation of the LIN/UART module (LIN master mode) in response transmission. **Table 18-8** provides processing in response transmission.

Figure 18-6. Operation in Response Transmission

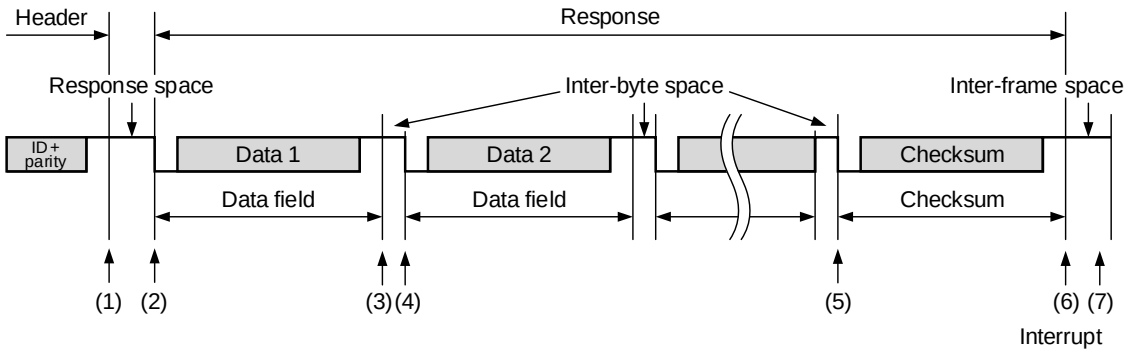


Table 18-8. Processing in Response Transmission

Step	Software processing	LIN/UART module processing
(1)	(When in frame separate mode) • Sets the RTS bit in the LTRCn register to 1 (response transmission/reception started). (When not in frame separate mode) • Waits for an interrupt request.	(When in frame separate mode) • Waits for the setting of the RTS bit in the LTRCn register to 1 by software. • When the bit is set to 1, sends a response space. (When not in frame separate mode) • 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. • Transmits an inter-byte space • Transmits the data 3. • Transmits an inter-byte space (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)). :
(5)		Transmits the checksum.
(6)		• Sets a successful frame/wake-up transmission flag. • Sets the FTS bit in the LTRCn register to 0 (frame transmission or wake-up transmission/reception stopped). (When in frame separate mode) • 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 **18.4.6 Error Status**.

(c) Response Reception

Figure 18-7 shows the operation of the LIN/UART module (LIN master mode) on response reception. **Table 18-9** provides processing in response reception.

Figure 18-7. Operation in Response Reception

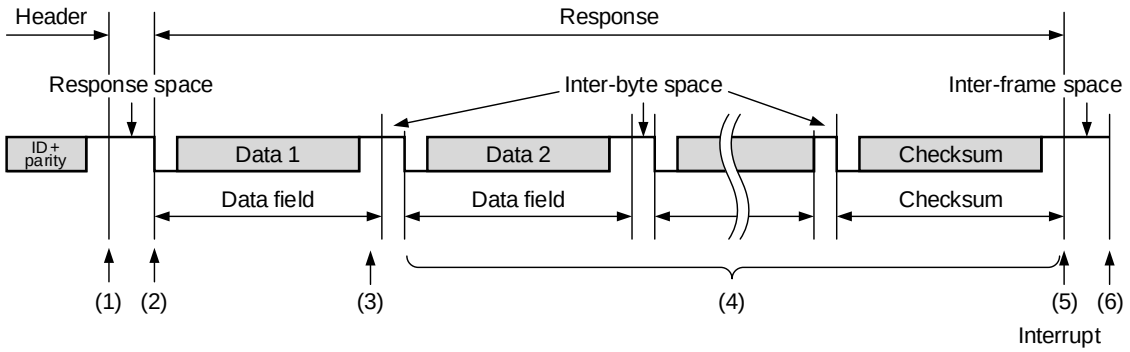


Table 18-9. Processing in Response Reception

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. • Receives the data 3 when the start bit is detected. 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). : • Receives the checksum when the start bit is detected.
(5)		• Determines the checksum. • Sets the successful frame/wake-up reception flag. • Sets the FTS bit in the LTRCn register to 0 (frame transmission or wake-up transmission/reception stopped).
(6)	• Processing after communication. Reads the received data. Checks the LSTn register and clears flags.	Idle

For information about error detection, refer to **18.4.6 Error Status**.

(2) LIN Slave Mode

(a) Header Reception

Figure 18-8 shows the operation of the LIN/UART module (LIN slave mode) in header reception. **Table 18-10** provides processing in header reception.

Figure 18-8. Operation in Header Reception

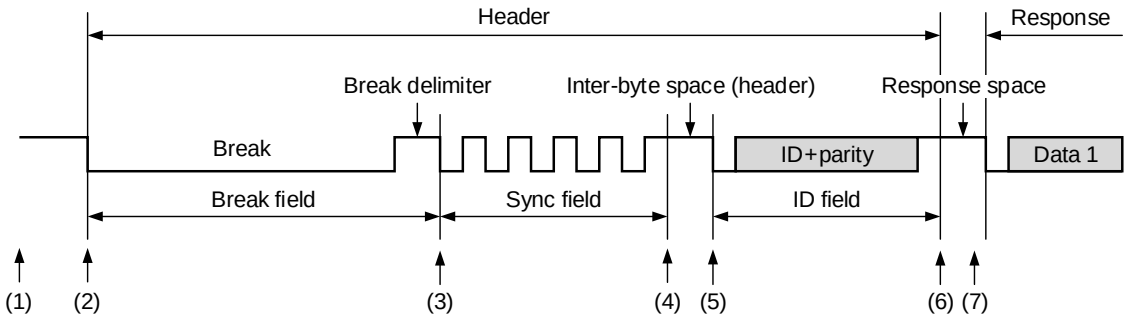


Table 18-10. Processing in Header Reception

Step	Software processing	LIN/UART module processing
(1)	• Sets a baud rate. • Sets noise filter ON/OFF. • Enables interrupt. • Enables error detection. • Sets frame configuration parameters. • Changes the LIN/UART module to the LIN slave mode: LIN operation mode. • 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 **(i) Auto Baud Rate Correction Function** in next page).
(4)		• Detects a sync field (55H) • Sets the baud rate generator (in the case of LIN slave mode [auto baud rate]) • Clears the no-response request bit (LNRR bit).
(5)		• Receives an ID field. • Checks an ID parity bit
(6)		Sets a header reception complete flag.
(7)	• Checks the LSTn register and clears flags. • Checks the LIDBn register and prepares a response.	• Completes a header reception process. • Waits for a response request.

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 **18.4.6 Error Status**.

(i) 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 18-9. Header Reception in LIN Slave Mode [Auto Baud Rate] (in Normal Operation)

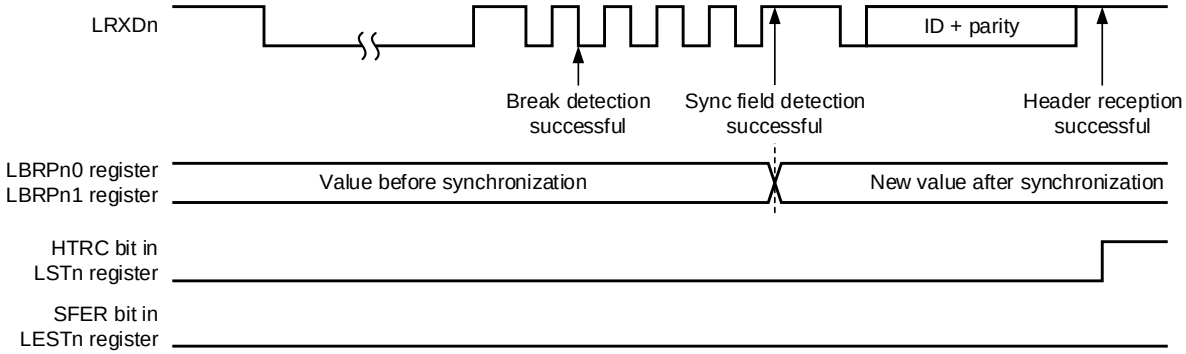
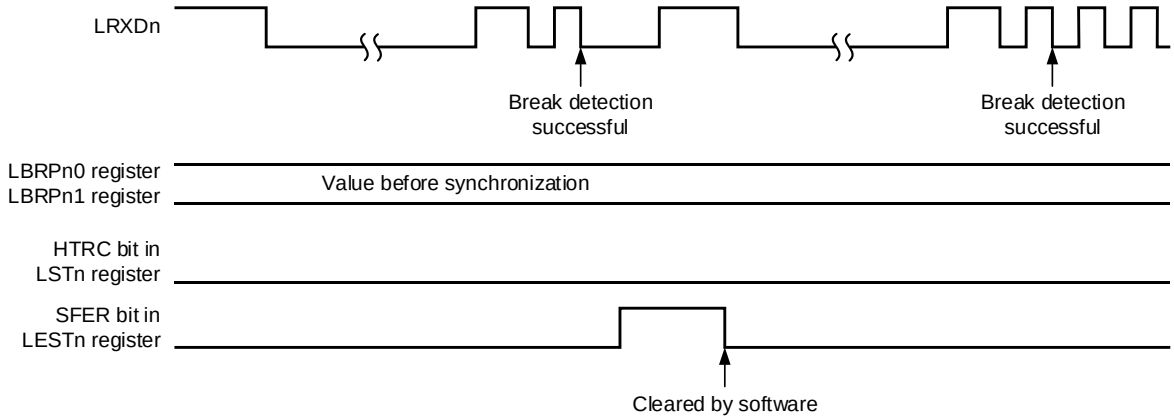


Figure 18-10. Header Reception in LIN Slave Mode [Auto Baud Rate] (Sync Field Error)



(b) Response Transmission

Figure 18-11 shows the operation of the LIN/UART module (in LIN slave mode) in response transmission. **Table 18-11** provides processing in response transmission.

Figure 18-11. Operation in Response Transmission

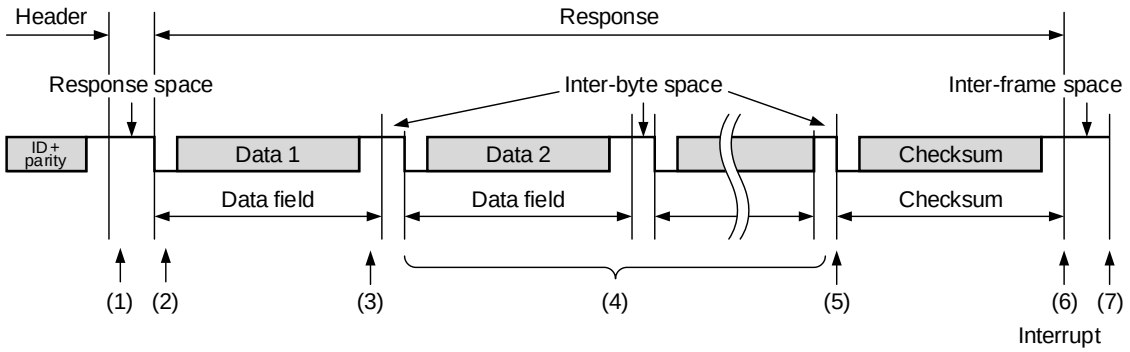


Table 18-11. Processing in Response Transmission

Step	Software processing	LIN/UART module processing
(1)	• Sets the LDFCn register. • Sets the LDBnm register. • 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. • 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. • RSDD bit in the LRSSn register is 1 (dominant level in response space detected) when the bit width of the response space from the completion of the header reception (the stop bit of the ID field) to the start of transmission is 0.5 Tbit or more.
(3)		Transmits the inter-byte space.
(4)		• Transmits the data 2. • Transmits an inter-byte space • Transmits the data 3. • Transmits an inter-byte space (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). :
(5)		Transmits the checksum.
(6)		• Sets a successful frame/wake-up transmission flag or an error flag. • 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.	• Completes the response transmission process. • 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 **18.4.6 Error Status**.

(c) Response Reception

Figure 18-12 shows the operation of the LIN/UART module (LIN slave mode) in response reception. **Table 18-12** provides processing in response reception.

Figure 18-12. Operation in Response Reception

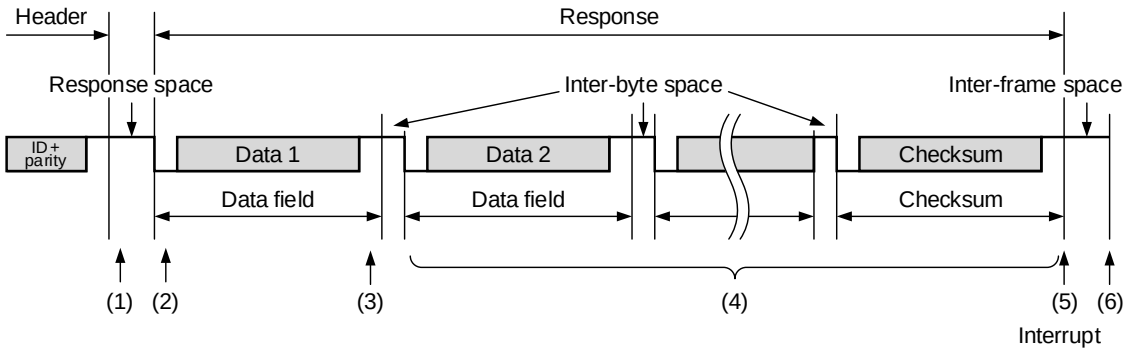


Table 18-12. Processing in Response Reception

Step	Software processing	LIN/UART module processing
(1)	• Sets the LDFCn register. • 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. • 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. • Receives the data 3 when the start bit is detected. 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). : • Receives the checksum when the start bit is detected.
(5)		• Determines the checksum. • Sets a successful frame/wake-up reception flag or an error flag. • Sets the RTS bit in the LTRCn register to 0 (response transmission/reception stopped).
(6)	• Processing after communication Reads the received data. Checks the LSTn register and clears flags.	• Completes the response process. • 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 **18.4.6 Error Status**.

(d) No-Response Request

Figure 18-13 shows the operation of the LIN/UART module (LIN slave mode) when no response is requested. **Table 18-13** shows the processing that occurs when no response is requested.

Figure 18-13. Operation when No Response is Requested

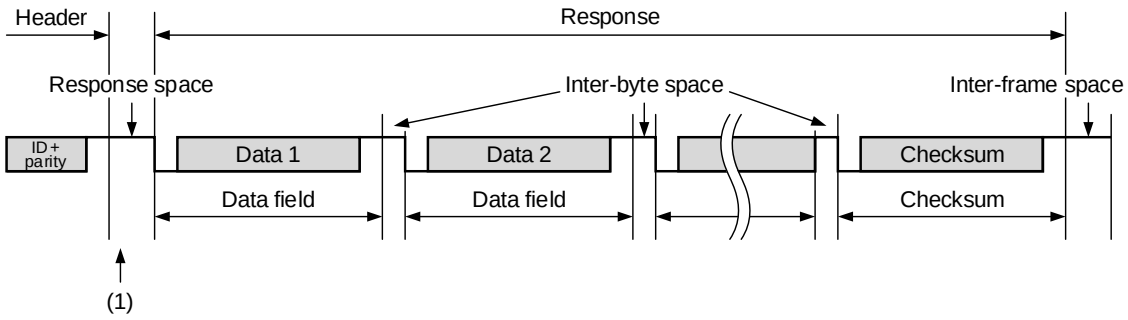


Table 18-13. Processing when No Response is Requested

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. • Completes the frame reception process. • Waits for a new break.

18.4.2 Data Transmission/Reception

(1) Data Transmission

One bit of data is transmitted per 1 Tbit.

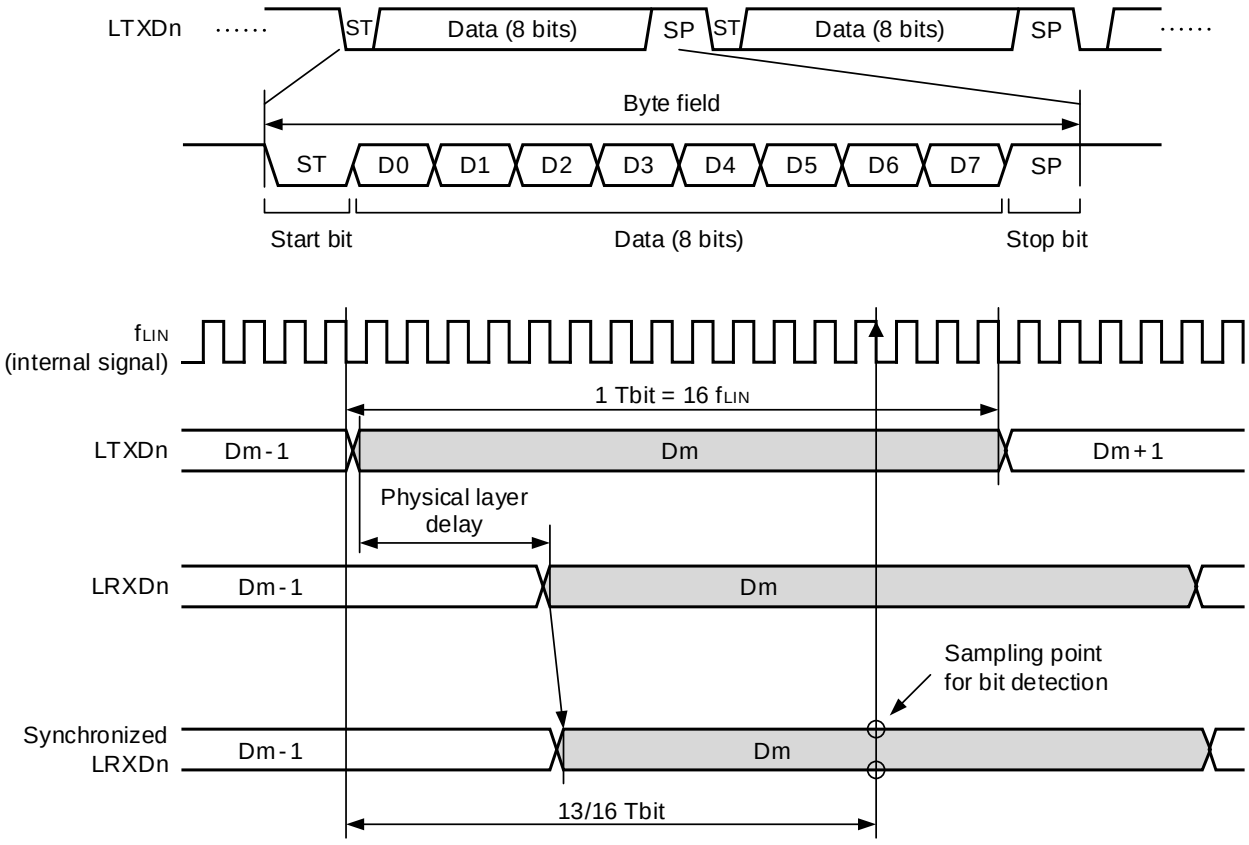
The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data is compared bit by bit, and the results are stored in the BER flag in the LESTn register (see **18.4.6 Error Status**).

In LIN master mode and LIN slave mode [fixed baud rate], 1 Tbit is generated to be 16 f_{LIN}, 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 4f_{LIN}, the sampling point for received data is at the third clock cycle (75% position). If 1 Tbit is generated to be 8f_{LIN}, the sampling point for received data is at the 7th clock cycle (87.5% position).

Figure 18-14 shows an example of data transmission timing.

Figure 18-14. 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.

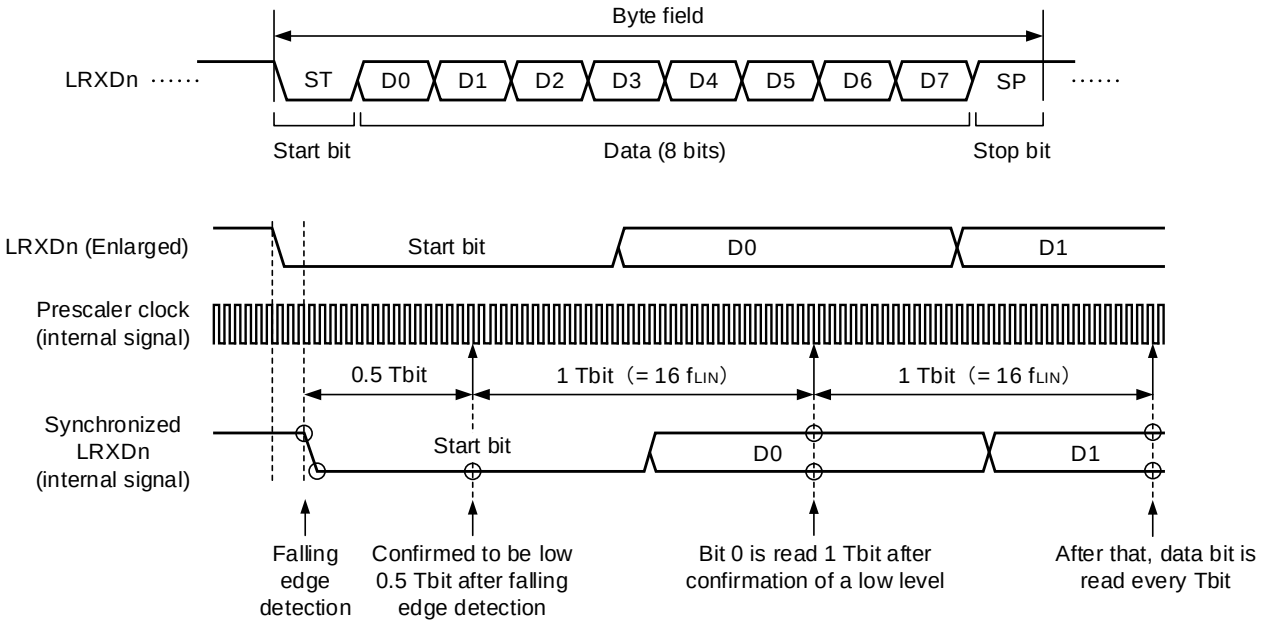
The byte field is synchronized at the falling edge of the start bit for the synchronized LRXDn signal. After the falling edge is detected, sampling is performed again 0.5 Tbit later, and the falling edge is recognized as a start bit if the synchronized LRXDn signal is low. The falling edge is not recognized 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 18-15 shows an example of data reception timing.

Figure 18-15. Example of Data Reception Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])



18.4.3 Transmission/Reception Data Buffering

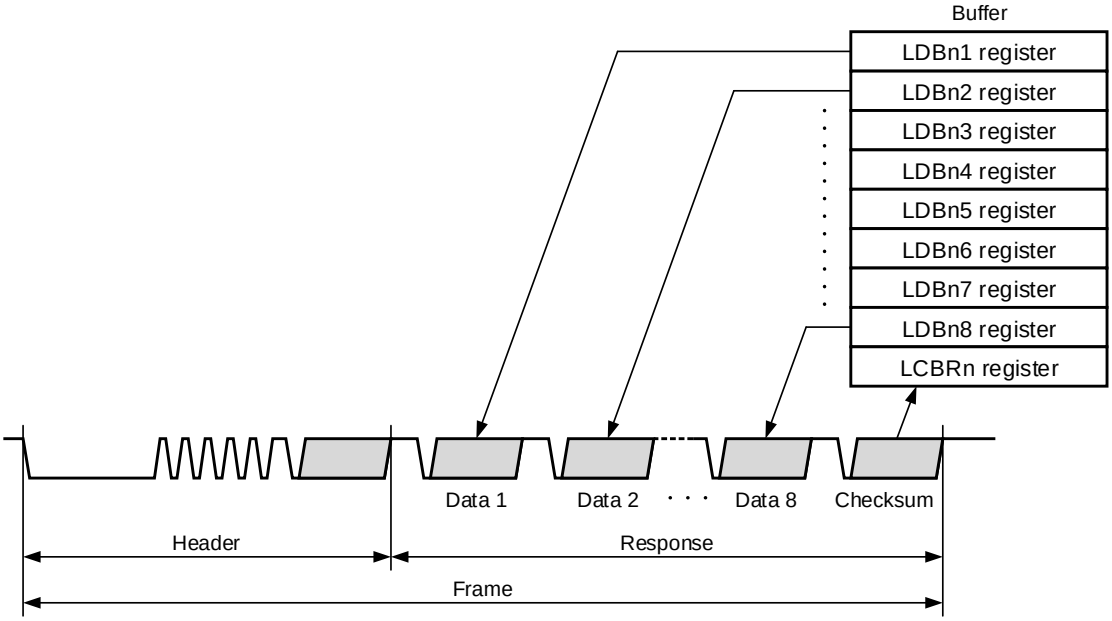
This section explains the buffer processing that takes place when the LIN/UART module sends or receives data continuously.

(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 18-16 depicts the LIN transmission processing and the required buffer.

Figure 18-16. 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).

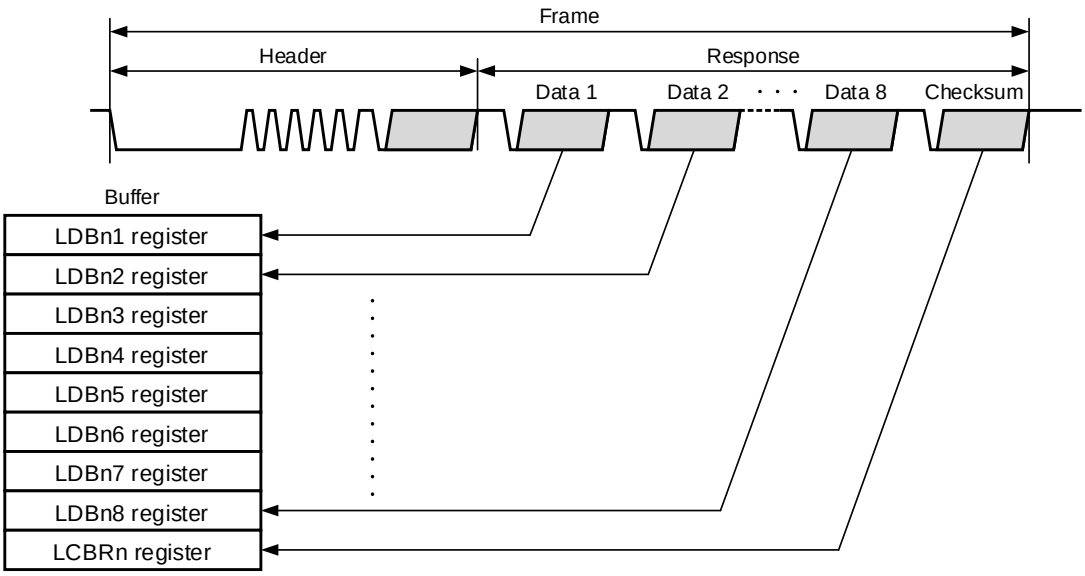
Use frame separate mode when sending or receiving response data of 9 bytes or greater in LIN master mode.

(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 18-17 depicts the LIN reception processing and the required buffer.

Figure 18-17. 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.

18.4.4 Wake-up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

(1) Wake-up Transmission

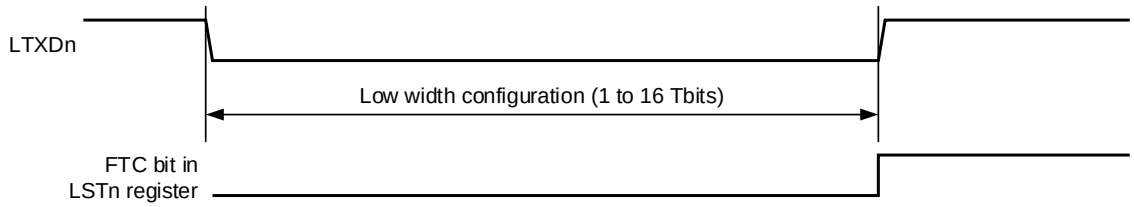
In LIN wake-up mode, setting the RCDS bit in the LDFCn register to 1 (transmission) and the FTS bit in the LTRCn register to 1 (header reception or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low width of the wake-up signal should be set using 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 18-18 shows the wake-up transmission timing.

Figure 18-18. 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 f_{LIN} 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.

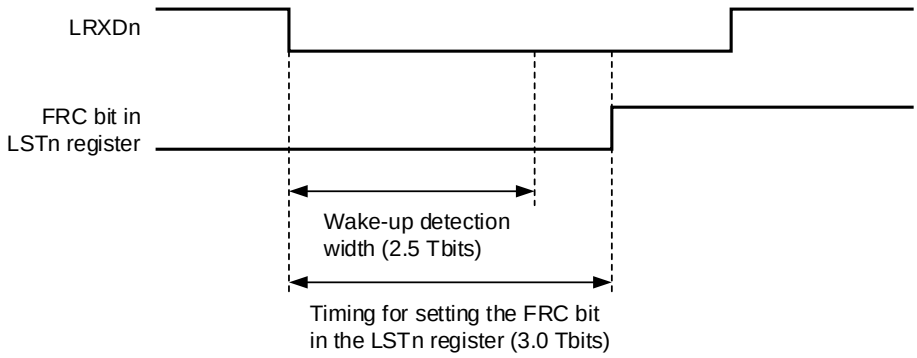
Set the LWBR0 bit in the LWBRn register to 0 when LIN Specification Package Revision 1.3 is used, and set it to 1 when LIN Specification Package Revision 2.x is used. When the LWBR0 bit is set to 1, fa is always selected as the LIN system clock (f_{LIN}) regardless of the setting of LCKS bits in the 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 18-19. Input Signal Low Count Function



18.4.5 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 18-14 shows the types of statuses available in LIN master mode. **Table 18-15** lists the types of statuses available in LIN slave mode [auto baud rate] and in LIN slave mode [fixed baud rate].

Table 18-14. 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 "LIN reset mode is canceled", 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 • 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 • When cleared by software • After transition to LIN reset mode	• LIN operation mode • 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 • When cleared by software • After transition to LIN reset mode	• LIN operation mode • 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 • When cleared by software Note 1 • After transition to LIN reset mode	• LIN operation mode • 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 • When cleared by software • 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 • When cleared by software • After transition to LIN reset mode	LIN operation mode	HTRC flag in LSTn register	Available

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 18-15. 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 “LIN reset mode is canceled”, 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 • 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 • After transition to LIN reset mode	• LIN operation mode • 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 • After transition to LIN reset mode	• LIN operation mode • 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 • After transition to LIN reset mode	• LIN operation mode • 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 • 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 • After transition to LIN reset mode	LIN operation mode	HTRC flag in LSTn register	Available

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).

18.4.6 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 18-16 shows the types of error statuses.

Table 18-16. 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 Notes 1, 2	• LIN operation mode • LIN wake-up mode	Cancel	○	BER flag in LESTn register
Physical bus error	• LIN bus is detected to be high when sending a break • LIN bus is detected to be low when sending a break delimiter • LIN bus is detected to be high when sending a wake-up	• LIN operation mode • LIN wake-up mode	Cancel	○	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	○	FTER flag in LESTn register
Framing error	In response field reception, a stop bit of each data byte is low	LIN operation mode	Cancel	○	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: • The first reception data byte is received after completion of header transmission but before a response transmission/reception request is set • 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-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 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]

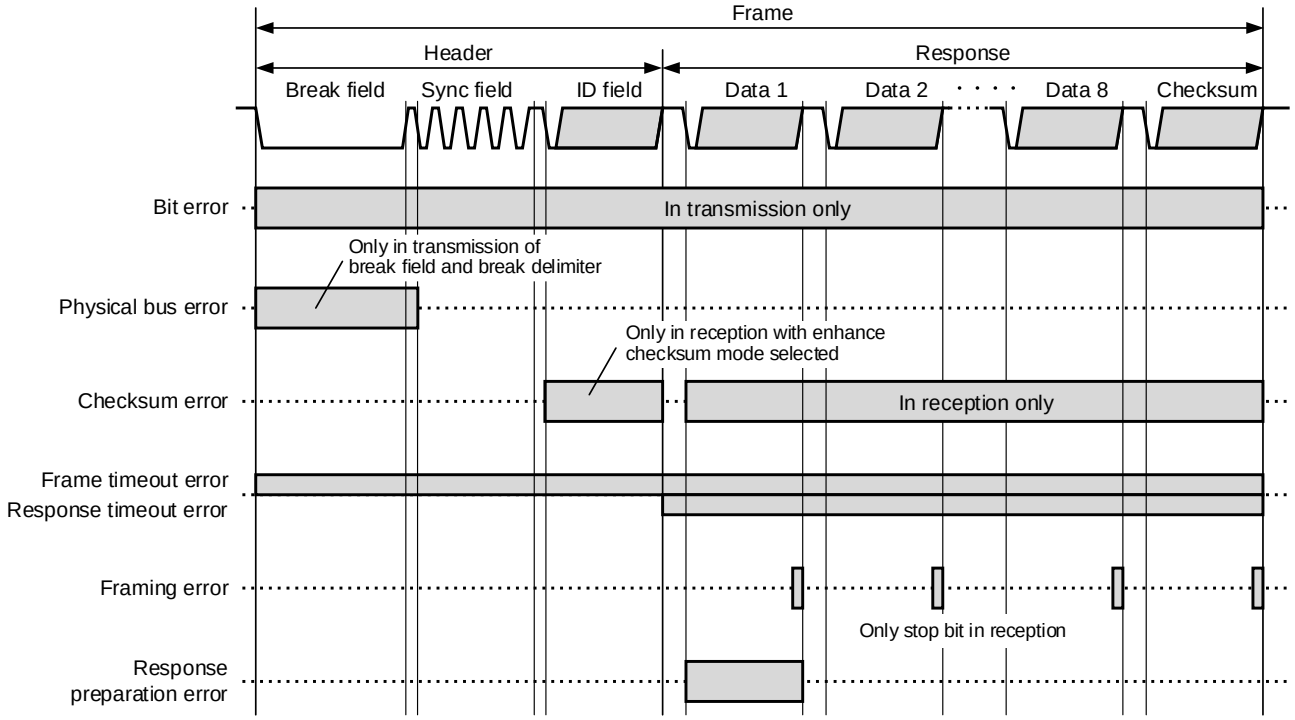
Caution The error status is cleared when another communication is started, when cleared by software, or after transition to LIN reset mode.
When restarting LIN communication after detecting a frame timeout error or response timeout error, do not enter LIN reset mode. To initialize the LIN module (RLIN3), set the LINnEN bit in the PER2 register to 0, and then write 1 again.

(b) Target Time Area for LIN Error Detection

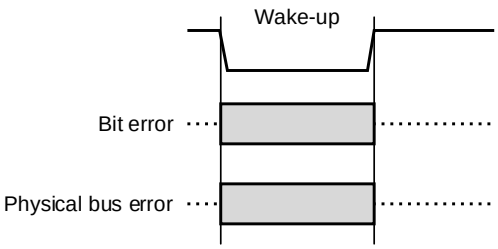
Figure 18-20 shows the time domain in which the LIN/UART module in master mode performs monitoring for error detection.

Figure 18-20. 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 18-17 shows the types of error statuses.

Table 18-17. 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 • LIN wake-up mode	Cancel	○	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	○	TER flag in LESTn register
Framing error	In frame reception, a stop bit of each data byte is low	LIN operation mode	Cancel	○	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	○ 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	○	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. • 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.

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.

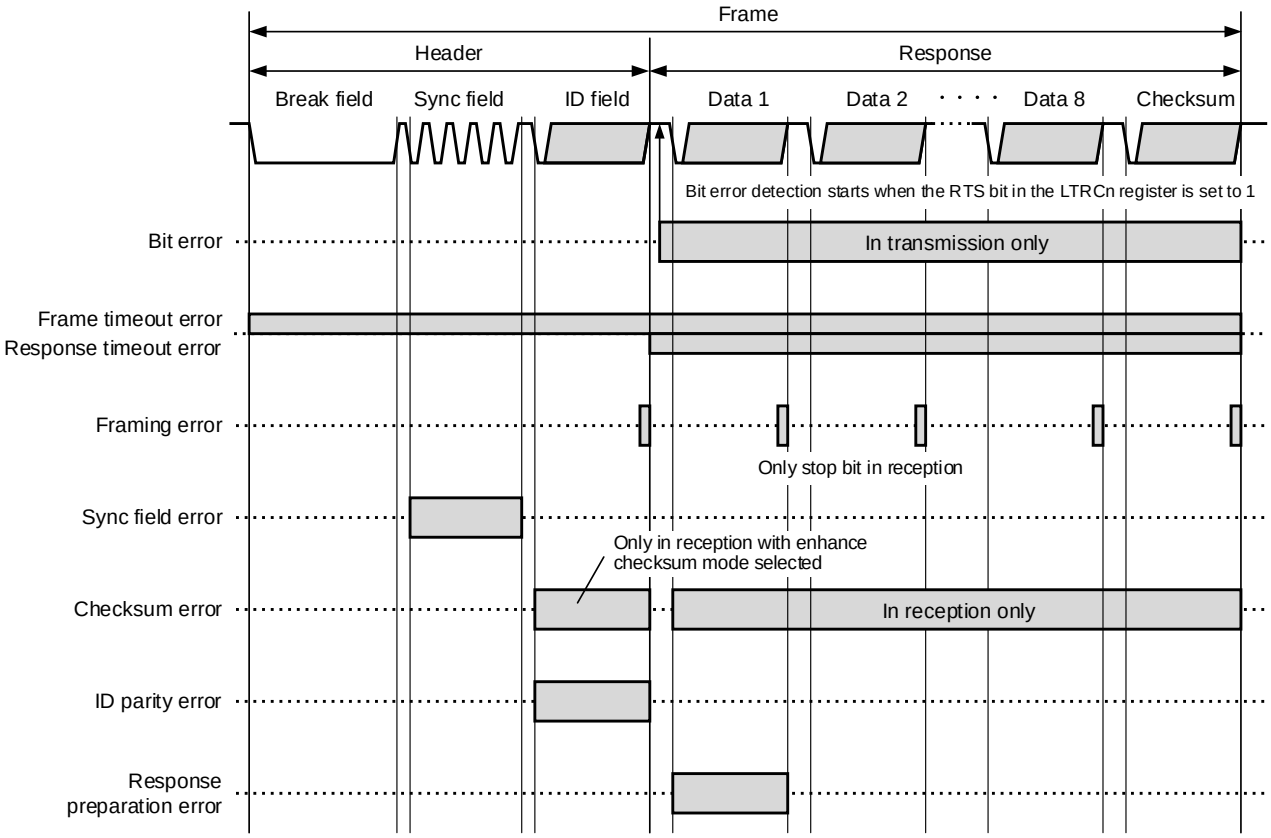
Caution The error status is cleared when cleared by software or after transition to LIN reset mode.

(b) Target Time Area for LIN Error Detection

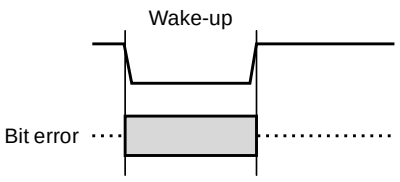
Figure 18-21 shows the time domain in which the LIN/UART module in slave mode performs monitoring for error detection.

Figure 18-21. Target Time Area for LIN Error Detection (LIN Slave Mode)

<Frame transmission/reception>



<Wake-up transmission>



18.5 UART Mode

18.5.1 Operation Overview

(1) Transmission

Figure 18-22 shows LIN/UART module (in UART mode) transmission operations; **Table 18-18** shows LIN/UART module (in UART mode) transmission processing.

Figure 18-22. LIN/UART Module (in UART mode) Transmission Operation

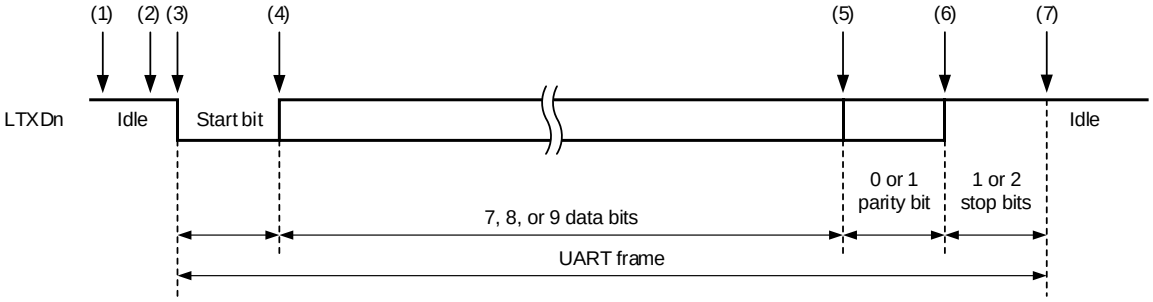


Table 18-18. LIN/UART Module (UART Mode) Transmission Processing

Step	Software processing	LIN/UART module processing
(1)	• Sets a baud rate. • Sets noise filter ON/OFF. • Sets error detection enable. • Sets data format • Sets an interrupt generation timing. • Clears the LIN/UART module from LIN reset mode • 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) or UART wait transmission data register (LUWTDRn).	• Sets the transmission status flag.
(3)	• Waits for an interrupt request. [When the UTIGTS bit is 0 (a transmission interrupt request is output upon start of transmission)] • 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 **18.5.1 (4) Transmission Start Wait Function**). [When the UTIGTS bit is 0 (a transmission interrupt request is output upon start of transmission)] • 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)] • If another piece of transmission data is set, goes to step (3). [When the UTIGTS bit is 1 (a transmission interrupt request is output upon end of transmission)] • When transmitting data continuously, goes to step (2).	[When the UTIGTS bit is 0 (a transmission interrupt request is output upon start of transmission)] • If another piece of transmission data is set, goes to step (3). • If another piece of transmission data is not set, clears the transmission status flag. [When the UTIGTS bit is 1 (a transmission interrupt request is output upon end of transmission)] • Outputs a transmission interrupt. • 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 18-23** shows an operating example where the transmission interrupt generation timing is the start of transmission. **Figure 18-24** shows an operating example where the transmission interrupt generation timing is the end of transmission.

Figure 18-23. LIN/UART Module (in UART mode) Continuous Transmission Operation (when UTIGTS Bit in LUORn1 Register is 0)

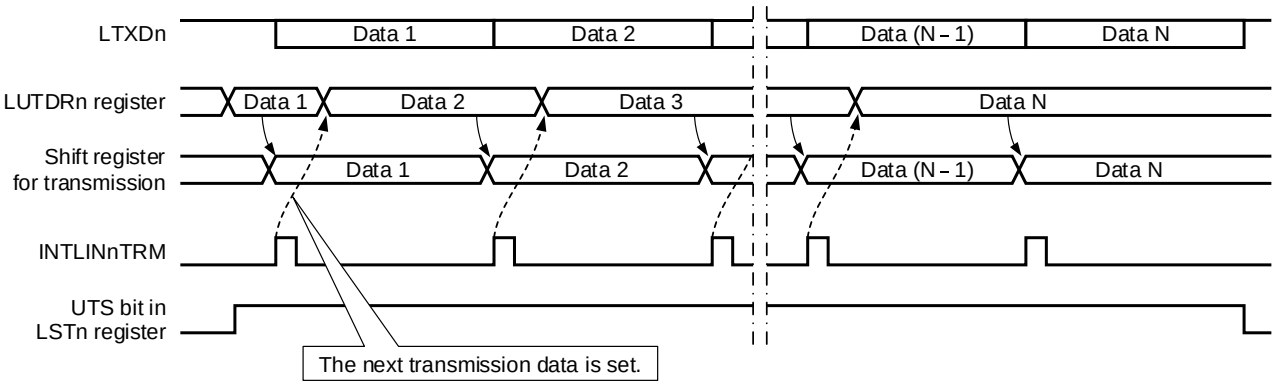
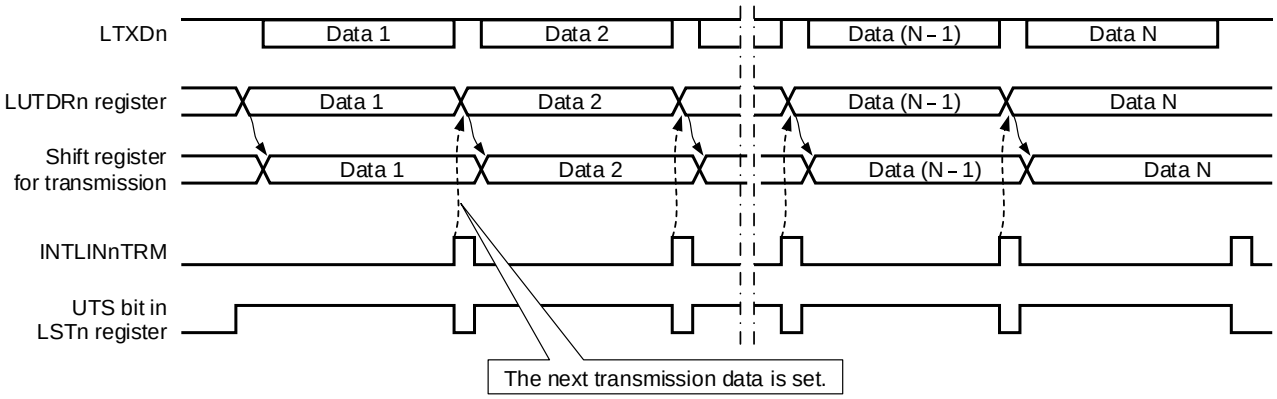


Figure 18-24. LIN/UART Module (in UART mode) Continuous Transmission Operation (when UTIGTS Bit in LUORn1 Register is 1)



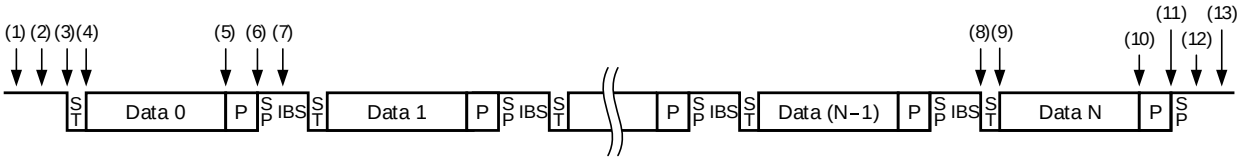
An interrupt can be generated at the end of a transmission by changing the UTIGTS bit in the LUORn1 register from 0 to 1 after the start of transmission of final data, provided only that the transmission interrupt generation timing is the start of transmission and the end of transmission of final data 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 18-25 shows the UART buffer transmission operation in the LIN/UART module (in UART mode). **Table 18-**19 shows the UART buffer transmission processing.

Figure 18-25. UART Buffer Transmission in LIN/UART Module (in UART mode)



ST: Start bit
P: Parity bit
SP: Stop bit
IBS: Inter-byte space

Table 18-19. UART Buffer Transmission Processing in LIN/UART Module (in UART mode)

Step	Software processing	LIN/UART module processing
(1)	• Sets a baud rate. • Sets noise filter ON/OFF. • Sets error detection enable. • Sets data format. • Sets an interrupt generation timing to the end of transmission. • Clears the LIN/UART module from LIN reset mode. • 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. • Sets the transmission data in the UART data buffer 0 register (LUDBn0) and the LIN/UART data buffer m register (LDBnm). • 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 **18.5.1 (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. • Clears the UART buffer transmit start (RTS) bit. • Outputs a transmission interrupt. • Clears the transmission status flag.
(13)	• Checks the LSTn register and clears flags • When transmitting data continuously, goes to step (2).	

(2) Reception

Figure 18-26 shows the LIN/UART module (in UART mode) reception operation. **Table 18-20** shows the LIN/UART module (in UART mode) reception processing.

Figure 18-26. LIN/UART Module (in UART Mode) Reception Operation

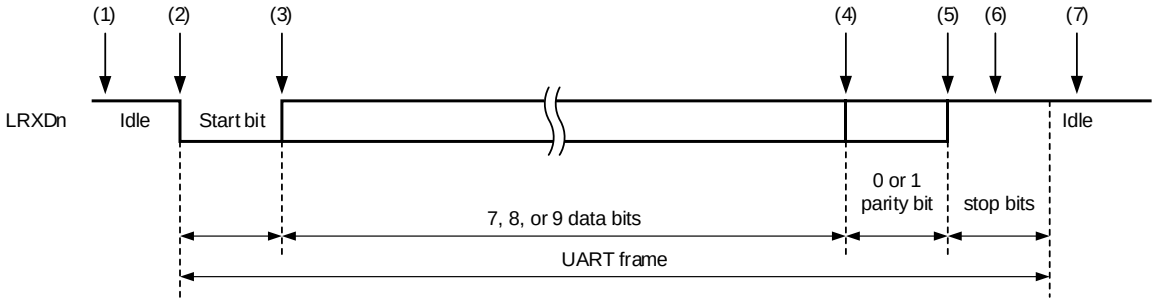


Table 18-20. LIN/UART Module (in UART Mode) Reception Processing

Step	Software processing	LIN/UART module processing
(1)	• Sets a baud rate. • Sets noise filter ON/OFF. • Sets error detection enable. • Sets data format. • Clears the LIN/UART module from LIN reset mode. • Sets the receive enable bit (UROE bit) to 1.	• Waits for reception enable state switching by software. • 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. • 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. • Clears the reception status flag.
(7)	Checks the LSTn register and clears flags.	Waits for a falling edge from the reception pin.

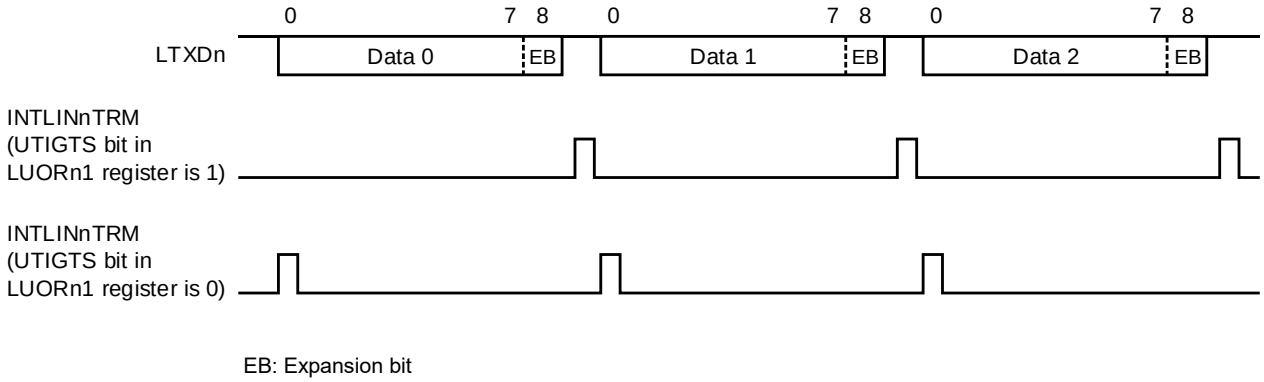
(3) Expansion Bits

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 18-27. Transmission Example When Expansion Bit is Enabled (LSB First)

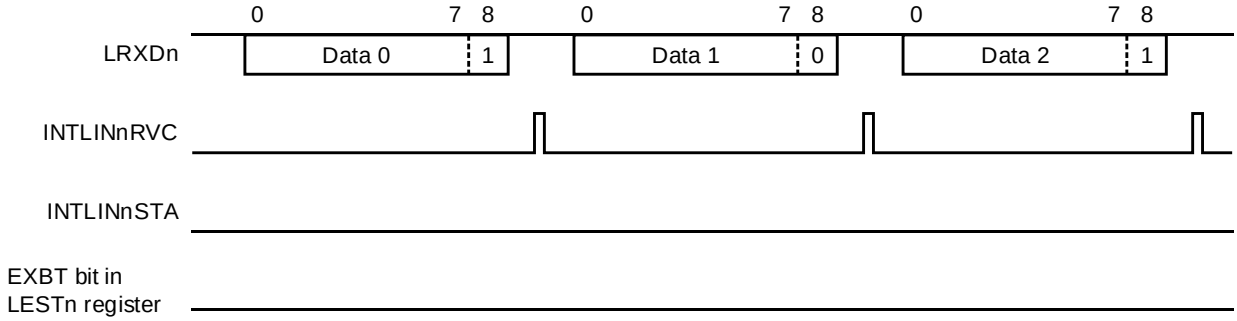


EB: Expansion bit

(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 when 9-bit data is received.

Figure 18-28. Expansion Bit Reception Example (LSB First)



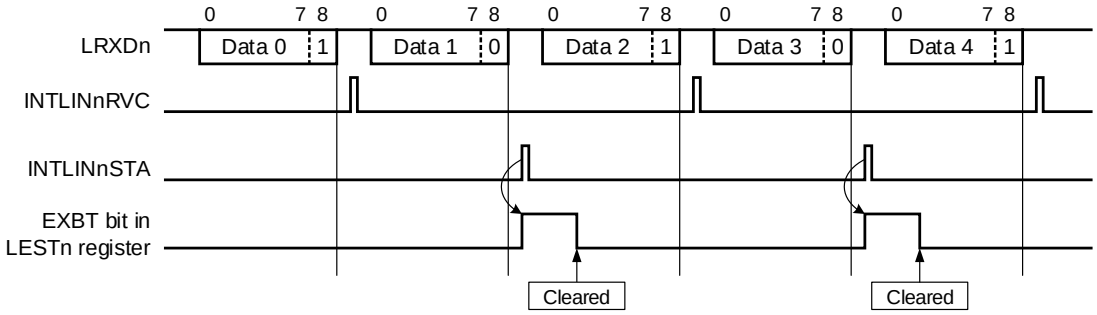
(c) Expansion Bit Reception (with Expansion Bit Comparison)

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 18-29 shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.

Figure 18-29. Expansion Bit Reception Example (with Expansion Bit Comparison) (LSB First, UEBDL = 0)



Cautions 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 to 2).
- 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).

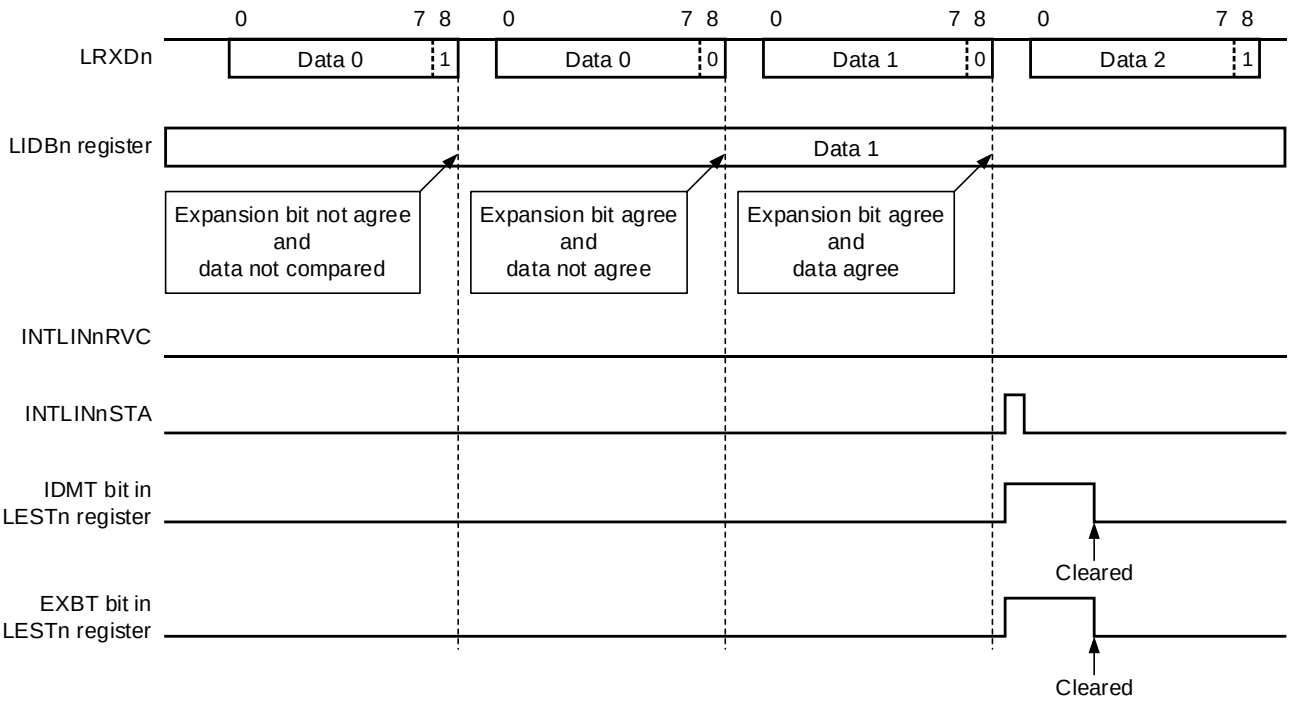
Even if the compared two values agree, a successful LINn reception interrupt is not generated.

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 18-30 shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.

Figure 18-30. Expansion Bit Reception Example (with Data Comparison)
(LSB First, UEBDL = 0)



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.

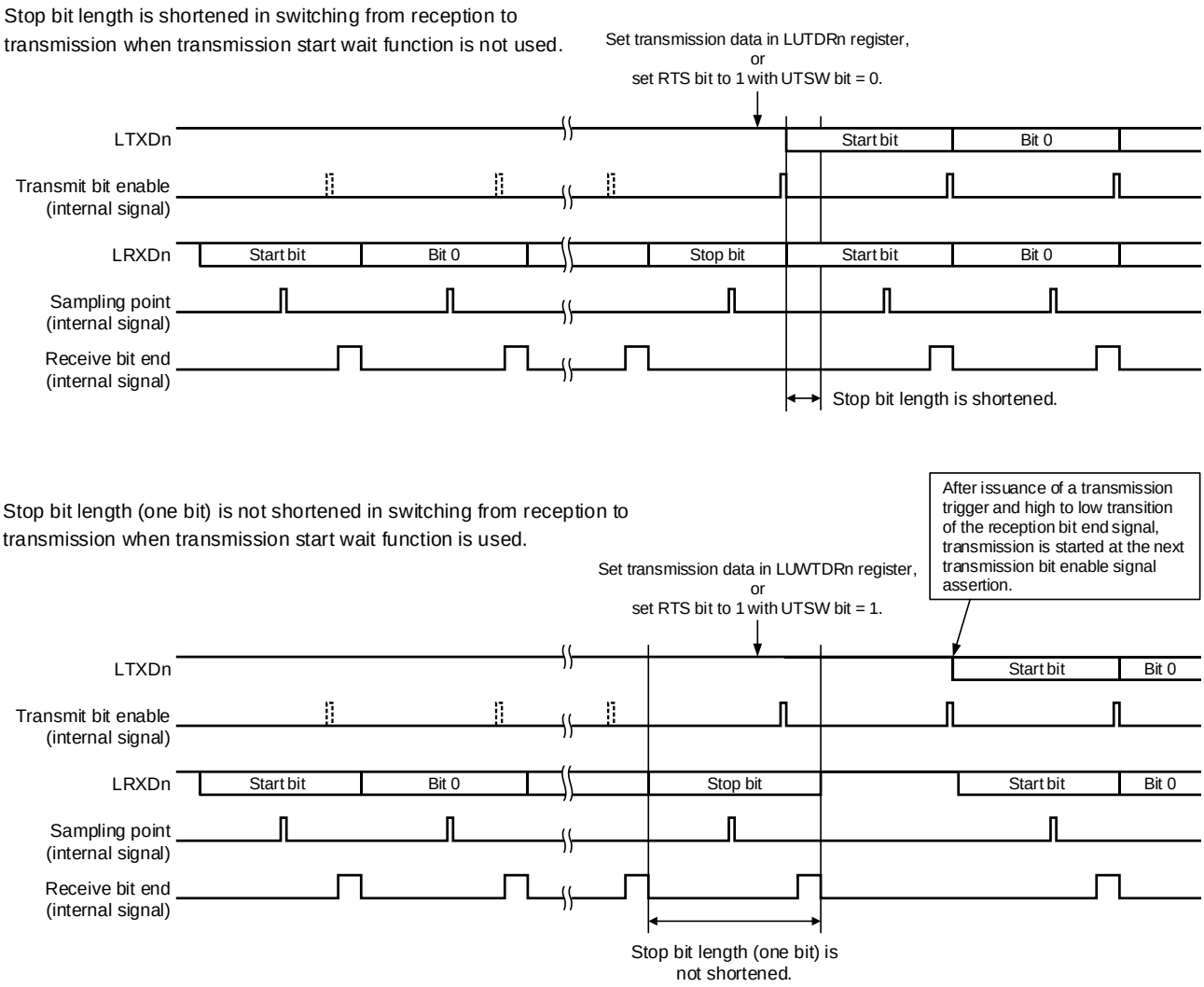
If it is desired to delay the start of transmission until the stop bits for the reception are completed, set data in the LUWTDRn register, which is used only for the wait function, instead of setting transmission data in the LUTDRn register as a start-of-transmission request. When transmitting from the 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 18-31 shows the operation of transmission wait function.

Figure 18-31. 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 **Table 18-21** and **Table 18-22** 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 9600 bps.**
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.**
5. **The CPU enters from the STOP mode to the SNOOZE mode on detecting the falling edge of the LRxDn signal. Note, however, the UART frame reception in the LIN/UART module (UART mode) may not start and the CPU may remain in the SNOOZE mode if an input pulse on the LRxDn pin is too short to be detected as a start bit. In such cases, data in the next UART frame reception may not be received correctly, and this may lead to a reception error.**

Table 18-21. Baud Rate Setting for UART Reception in SNOOZE Mode
(LIN communication clock source = 40 MHz ± 2.0%)

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	1038	2.66	−2.53
2400 bps	1/2	517	2.50	−2.36
4800 bps	1/2	256	1.97	−2.23
9600 bps	1/2	126	1.30	−1.58

Table 18-22. Baud Rate Setting for UART Reception in SNOOZE Mode
(LIN communication clock source = 32 MHz ± 2.0%)

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	830	2.64	−2.55
2400 bps	1/2	413	2.40	−2.46
4800 bps	1/2	205	2.16	−2.04
9600 bps	1/2	101	1.68	−1.19

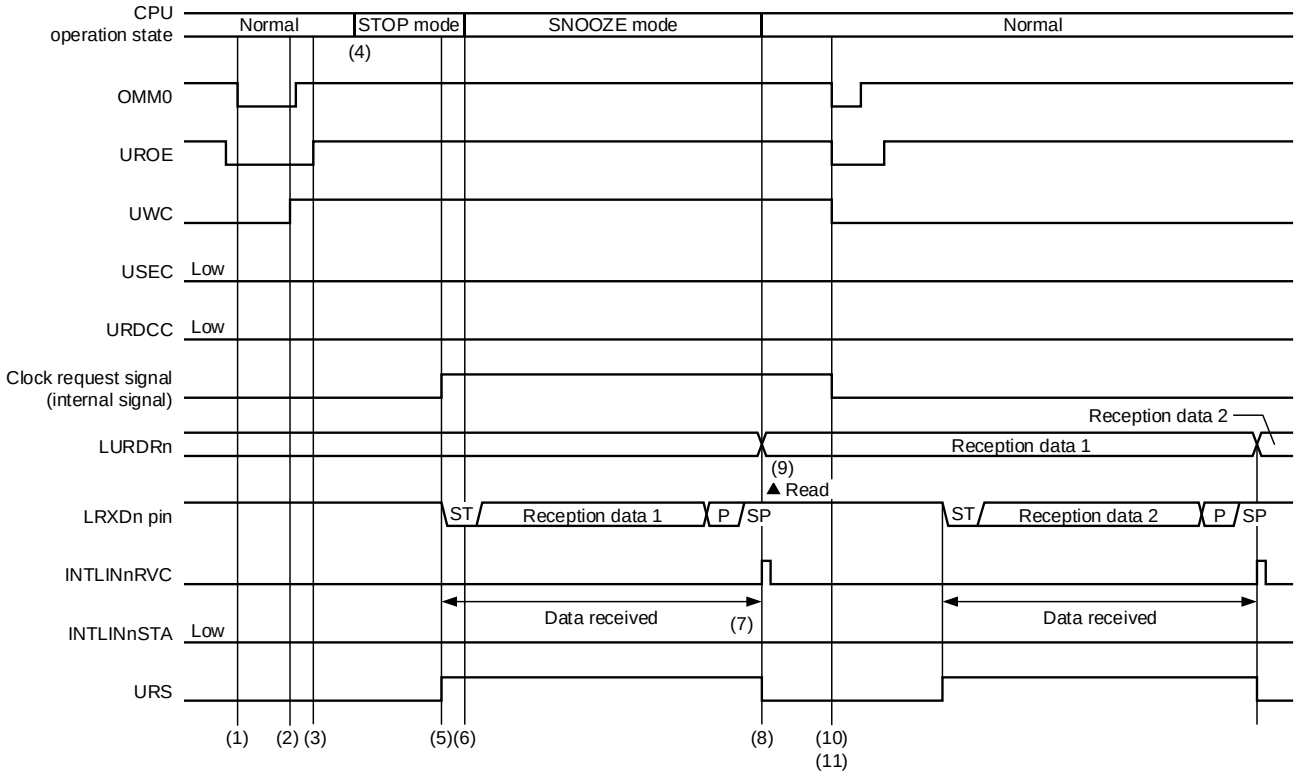
Remarks 1. The maximum and minimum allowable values are applied to the baud rates for UART reception. Set the parameters so that the baud rates for UART transmission should also fall within the allowable range.

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).

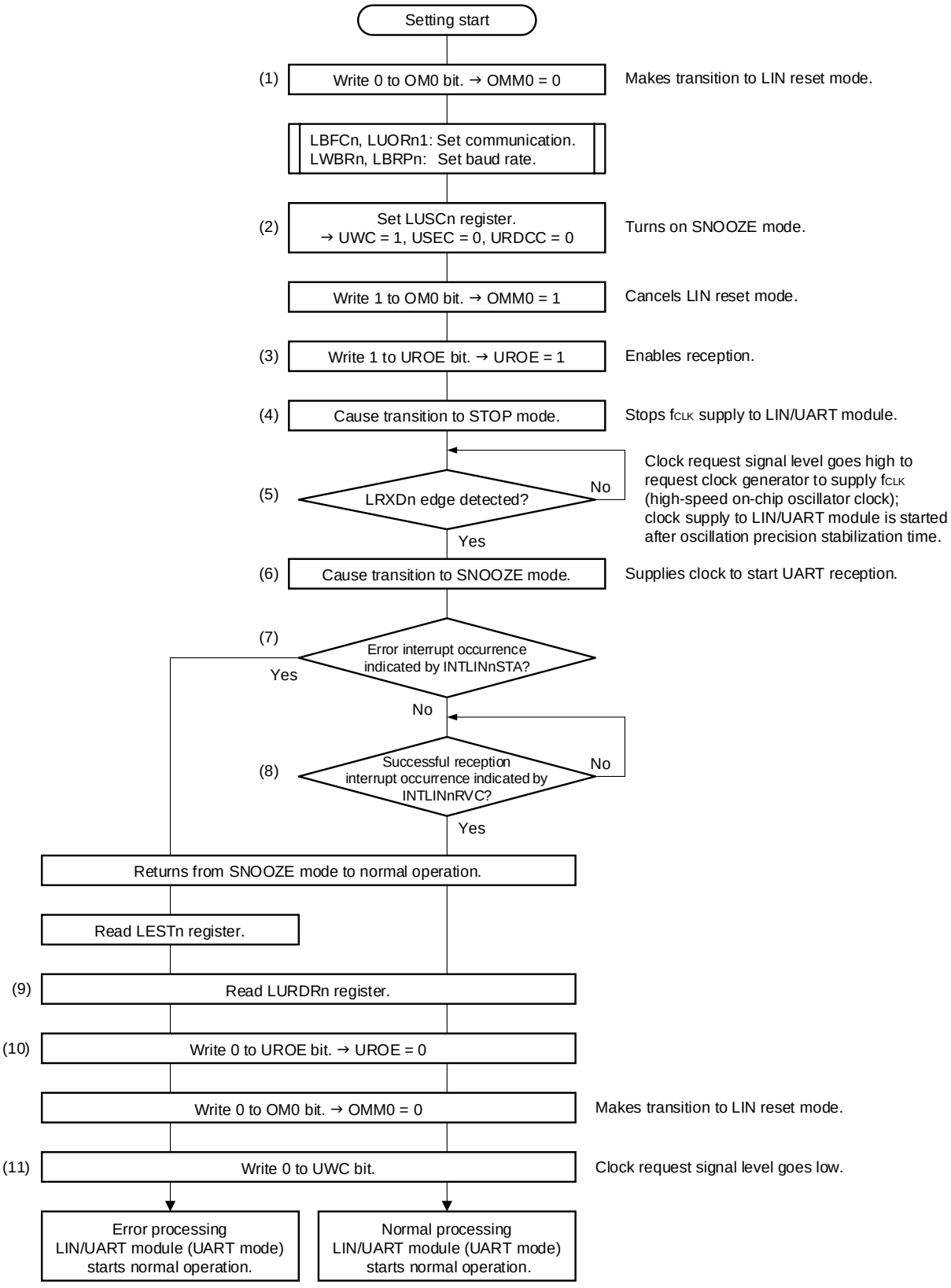
(a) SNOOZE Mode Operation (returning to normal operation upon successful reception; UWC = 1, URDCC = 0)

Figure 18-32. Timing Chart of SNOOZE Mode Operation
(returning to normal operation upon successful reception; UWC = 1, URDCC = 0)



Remark (1) to (11) in **Figure 18-32** correspond to (1) to (11) in **Figure 18-33**.

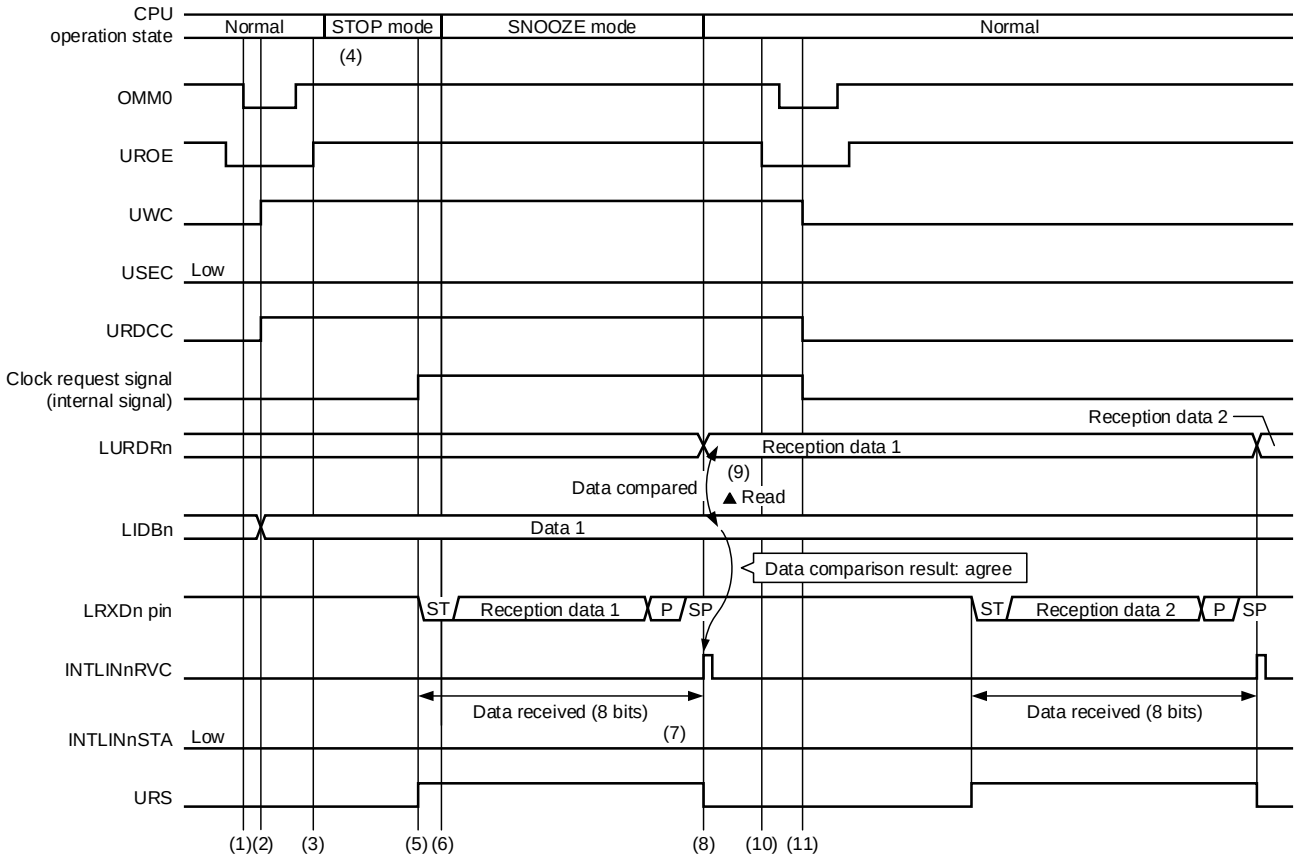
Figure 18-33. Flowchart of SNOOZE Mode Operation
(returning to normal operation upon successful reception; UWC = 1, URDCC = 0)



Remark (1) to (11) in **Figure 18-33** correspond to (1) to (11) in **Figure 18-32**.

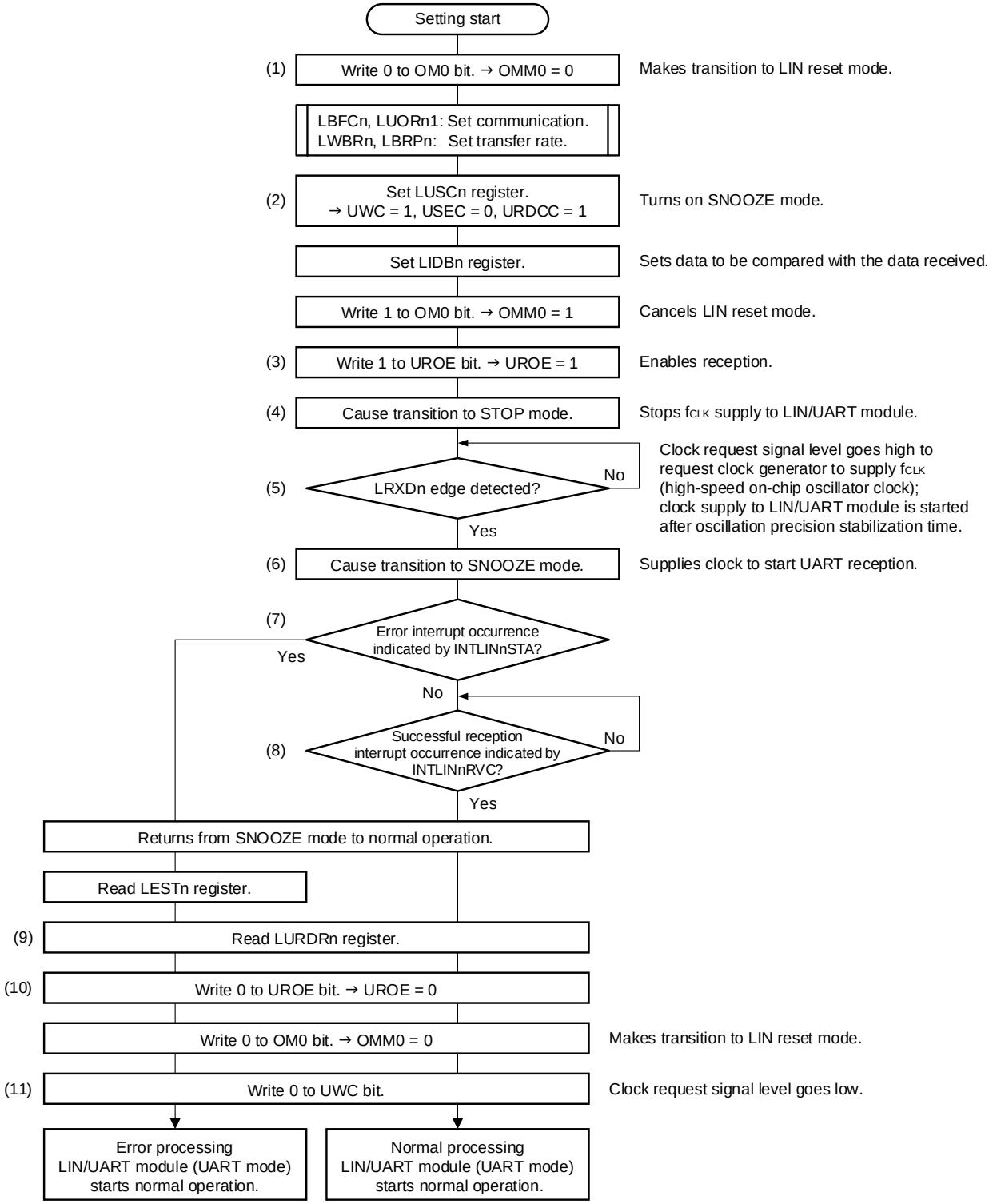
(b) SNOOZE Mode Operation (returning to normal operation upon comparison result agreement of data received; UWC = 1, URDCC = 1)

Figure 18-34. Timing Chart of SNOOZE Mode Operation
(returning to normal operation upon comparison result agreement of data received; UWC = 1, URDCC = 1)



Remark (1) to (11) in **Figure 18-34** correspond to (1) to (11) in **Figure 18-35**.

Figure 18-35. Flowchart of SNOOZE Mode Operation
(returning to normal operation upon comparison result agreement of data received; UWC = 1, URDCC = 1)



Remark (1) to (11) in **Figure 18-35** correspond to (1) to (11) in **Figure 18-34**.

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).

18.5.2 Data Transmission/Reception

(1) Data Transmission

One bit of data is transmitted per Tbit.

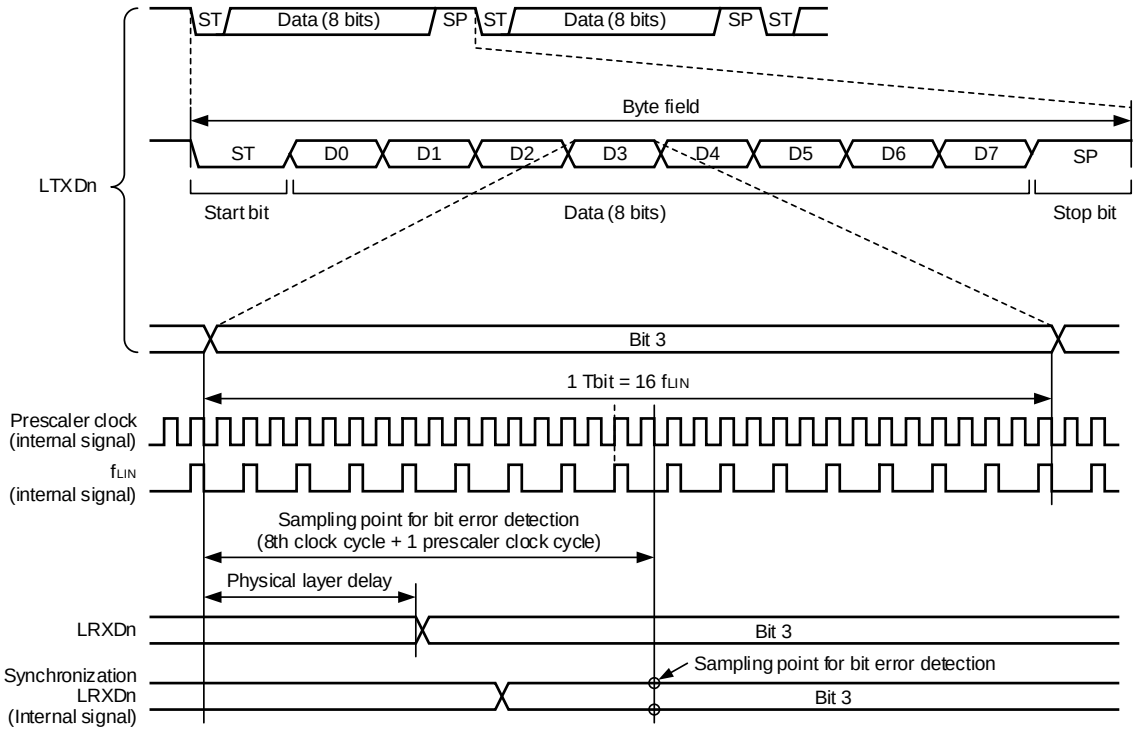
In half-duplex communication, if the BERE bit in the LEDEn register is 1 (bit error detection enabled), the transmission data and the input pin level are compared bit by bit during data transmission, and the results are stored in the BER flag in the LESTn register (see **18.5.5 Error Status**). The timing 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 18-23**.

Table 18-23. 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 18-36. 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.

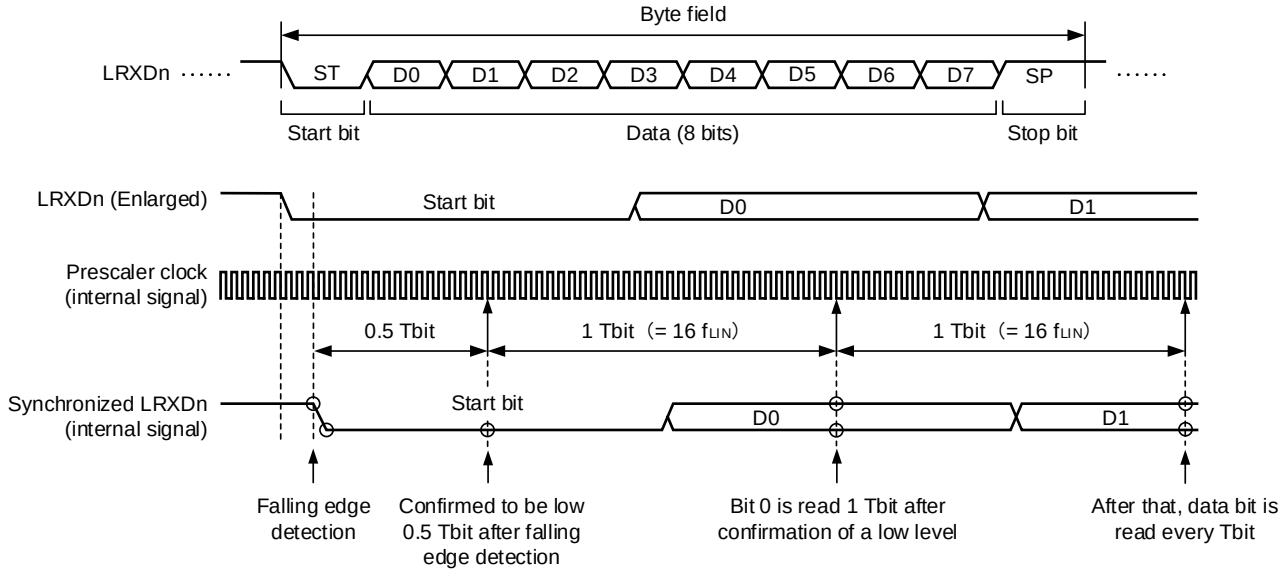
After the start bit is detected, 1 bit is sampled per Tbit.

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 18-37 shows an example of data reception timing.

Figure 18-37. Example of Data Reception Timing (when Sampling Count is 16 in 1 Tbit)



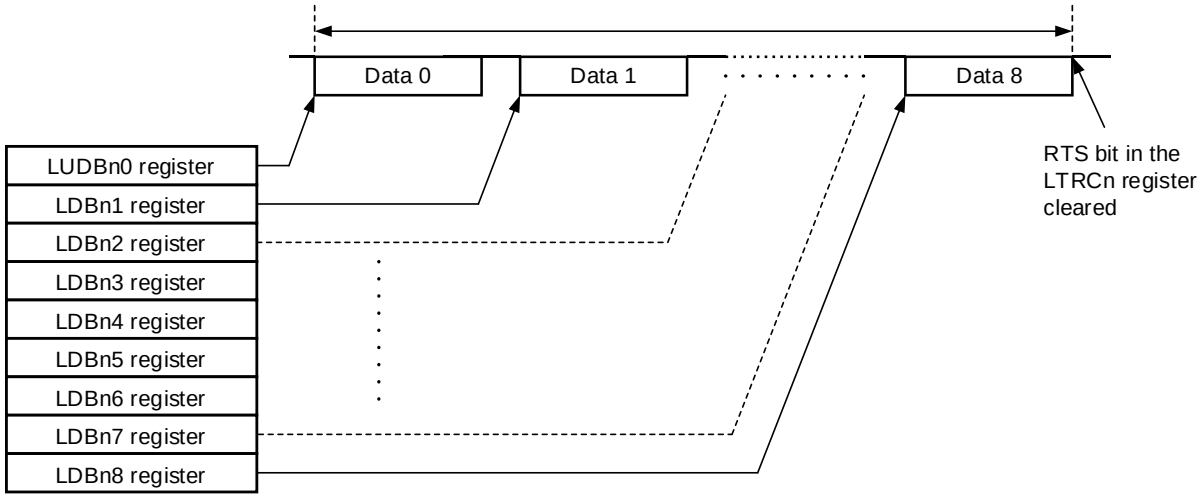
18.5.3 Buffer Processing of Transmission Data

(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. The spaces between transmission data items can be set in the IBS bit in the LSCn register.

Figure 18-38 shows a 9-byte UART buffer and the transmission processing.

Figure 18-38. UART Buffer and Transmission Processing (for 9-Byte Transmission)



18.5.4 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 18-24 shows the types of statuses available in UART mode.

Table 18-24. 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 "LIN reset mode is canceled", 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). • 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 • 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 flag in the LESTn register turns 1 (error detected).	• When cleared by software Note • After transition to LIN reset mode	ERR flag in LSTn register	Available
Transmission status	• When data is written to the LUTDRn or LUWTDRn register. • 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 • The transmission of the data in the UART buffer is complete, and the RTS bit in the LTRCn register is cleared • 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 • After transition to LIN reset mode	URS flag in LSTn register	Not available

Note Writing a 0 to the UPER, IDMT, EXBT, FER, 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.

18.5.5 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 18-25 lists applicable error status types.

Table 18-25. 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.	○	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. (In this case, no data is stored in the LURDRn register).	– (Reception is finished by the time this error is detected)	○	OER flag in LESTn register
Framing error	When the first stop bit at the first bit is low in the reception processing.	– (Reception is finished by the time this error is detected)	○	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.	–	○	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.	–	○	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.

18.6 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 (in the LWBRn register, the NSPB bit must be set to 0000B or 1111B, and the LPRS bit must be set to 000B).

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 18-39. Connection in LIN Reset Mode, LIN Mode and UART Mode

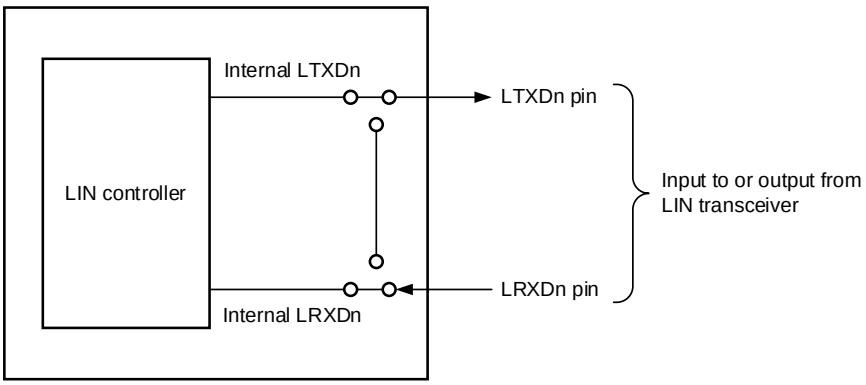
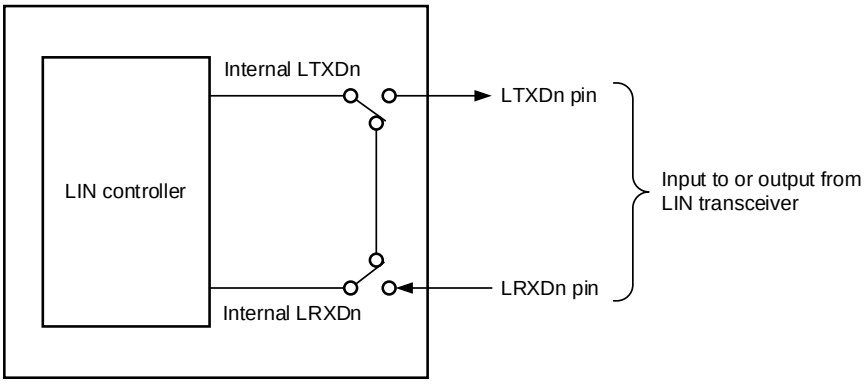


Figure 18-40. Connection in LIN Self-Test Mode



18.6.1 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.

18.6.2 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 = 0000000xB
  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
  Set the FTS bit in the LTRCn register to 1 (frame transmission or wake-up transmission/reception started).

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.
LBRPn0 register, LBRPn1 register, and LCKS bits in LMDn register

2. As necessary, set the related registers in **CHAPTER 22 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 × (number of data bytes + 1) [Tbit]
1 Tbit = LIN communication clock source × 16

Remark x: Any desired value

18.6.3 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 = 0000000xB
  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).

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.
LBRPn0 register, LBRPn1 register, LCKS bits in LMDn register, and IBS bits in LSCn register

2. As necessary, set the related registers in **CHAPTER 22 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 × (number of data bytes + 1) [Tbit]
1 Tbit = LIN communication clock source × 16.

Remark x: Any desired value

18.6.4 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 = 00000000B
  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.)

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.
LBRPn0 register, and LBRPn1 register

2. As necessary, set the related registers in **CHAPTER 22 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

18.6.5 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 = 00000000B
  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.)

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.
LBRPn0 register, LBRPn1 register, and IBS bits in LSCn register

2. As necessary, set the related registers in **CHAPTER 22 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 × (number of data bytes + 1) [Tbit]
1 Tbit = LIN communication clock source × 16

Remark x: Any desired value

18.6.6 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).

18.7 Baud Rate Generator

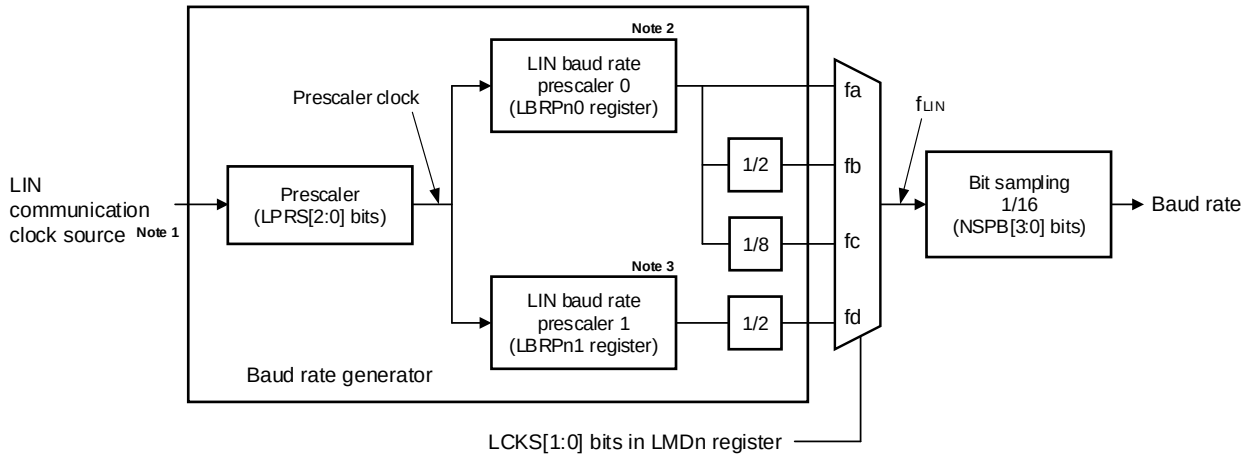
The prescaler clock is obtained by frequency-dividing the LIN communication clock source by the prescaler, and the LIN system clock (f_{LIN}) is obtained by frequency-dividing the prescaler clock by the baud rate generator. The clock obtained by frequency-dividing the LIN system clock (f_{LIN}) by the 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.

18.7.1 LIN Master Mode

Figure 18-41 shows a block diagram of baud rate generation in LIN master mode.

Figure 18-41. 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.

LIN communications clock source = f_{CLK} **Note**

In the range from 4 MHz to 40 MHz

Note 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 18-26 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 18-26. Examples of Baud Rate (19200, 10417, 9600, and 2400 bps) Generation in LIN Master Mode

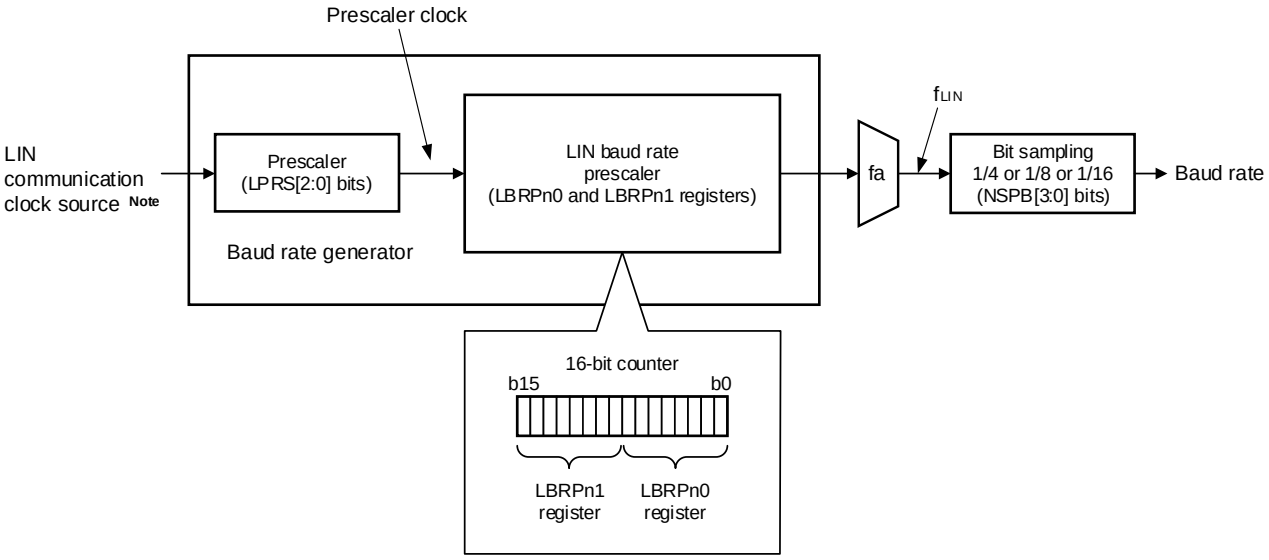
LIN communication clock source	Prescaler	Baud rate generator 0 (N+1) frequency-divided	Baud rate generator 1 (M+1) frequency-divided	LIN system clock	Baud rate	Error
40 MHz	1/1	130	–	fa	19230.77	+0.16%
		–	120	fd	10416.67	−0.003%
		130	–	fb	9615.38	+0.16%
		130	–	fc	2403.85	+0.16%
32 MHz	1/1	104	–	fa	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).

18.7.2 LIN Slave Mode

Figure 18-42 shows a block diagram of baud rate generation in LIN slave mode.

Figure 18-42. 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.
LIN communications clock source = f_{CLK} **Note**
In the range from 4 MHz to 40 MHz

Note 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 18-27 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 18-27. Examples of Baud Rate Generation (19200 bps, 10417 bps, 9600 bps and 2400 bps) in LIN Slave Mode [Fixed Baud Rate]

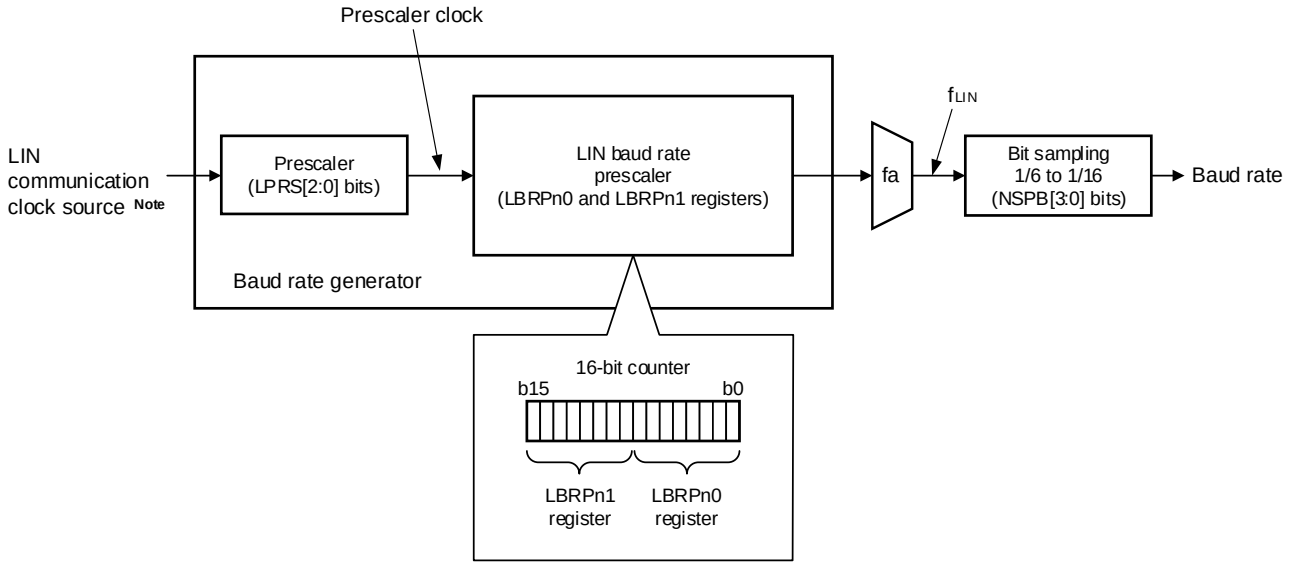
LIN communication clock source	Prescaler	Baud rate generator 0-1 N+1 frequency-divided	Baud rate	Error
40 MHz	1/1	130	19230.77	+0.16%
		240	10416.67	−0.003%
		260	9615.38	+0.16%
		1041	2401.54	+0.06%
32 MHz	1/1	104	19230.77	+0.16%
		192	10416.67	−0.003%
		208	9615.38	+0.16%
		833	2400.96	+0.04%
24 MHz	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).

18.7.3 UART Mode

Figure 18-43 shows a block diagram of baud rate generation in UART mode.

Figure 18-43. 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.
LIN communications clock source = f_{CLK} **Note**
In the range from 4 MHz to 40 MHz

Note 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 18-28. UART Baud Rate Setting Examples (when LIN communication clock source = 40MHz, 32 MHz)

LIN communication clock source	UART Baud Rate (Target Baud Rate)	Prescaler	Baud rate generator 0-1 N+1 frequency-divided	Baud rate	Error
40 MHz	1200 bps	1/2	1041	1200.77	+0.06%
	2400 bps	1/2	521	2399.23	-0.03%
	4800 bps	1/2	260	4807.69	+0.16%
	9600 bps	1/2	130	9615.38	+0.16%
	19200 bps	1/2	65	19230.77	+0.16%
	31250 bps	1/2	40	31250.00	0.00%
	38400 bps	1/1	65	38461.54	+0.16%
32 MHz	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 + (100H × LBRPn1) + 1))
÷ number selected by NSPB[3:0] bps

18.8 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 to 2) 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 18-44 shows the configuration of the noise filter, **Figure 18-45** an example of a noise filter circuit, and **Figure 18-46** the determination of the received data when the noise filter is used.

Figure 18-44. Configuration of Noise Filter

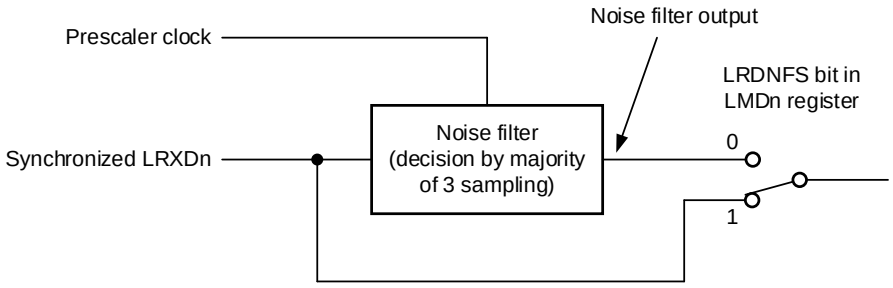


Figure 18-45. Example of Noise Filter Circuit

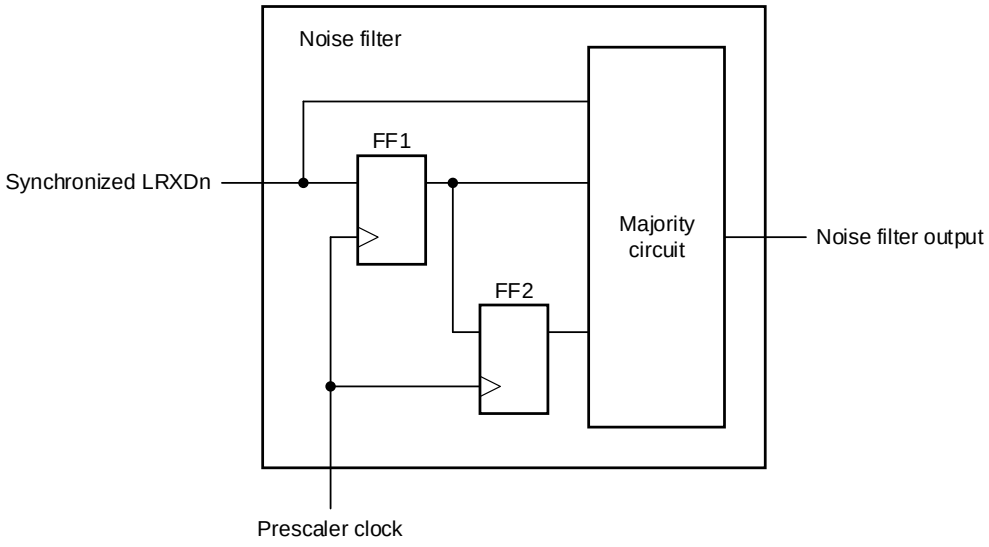
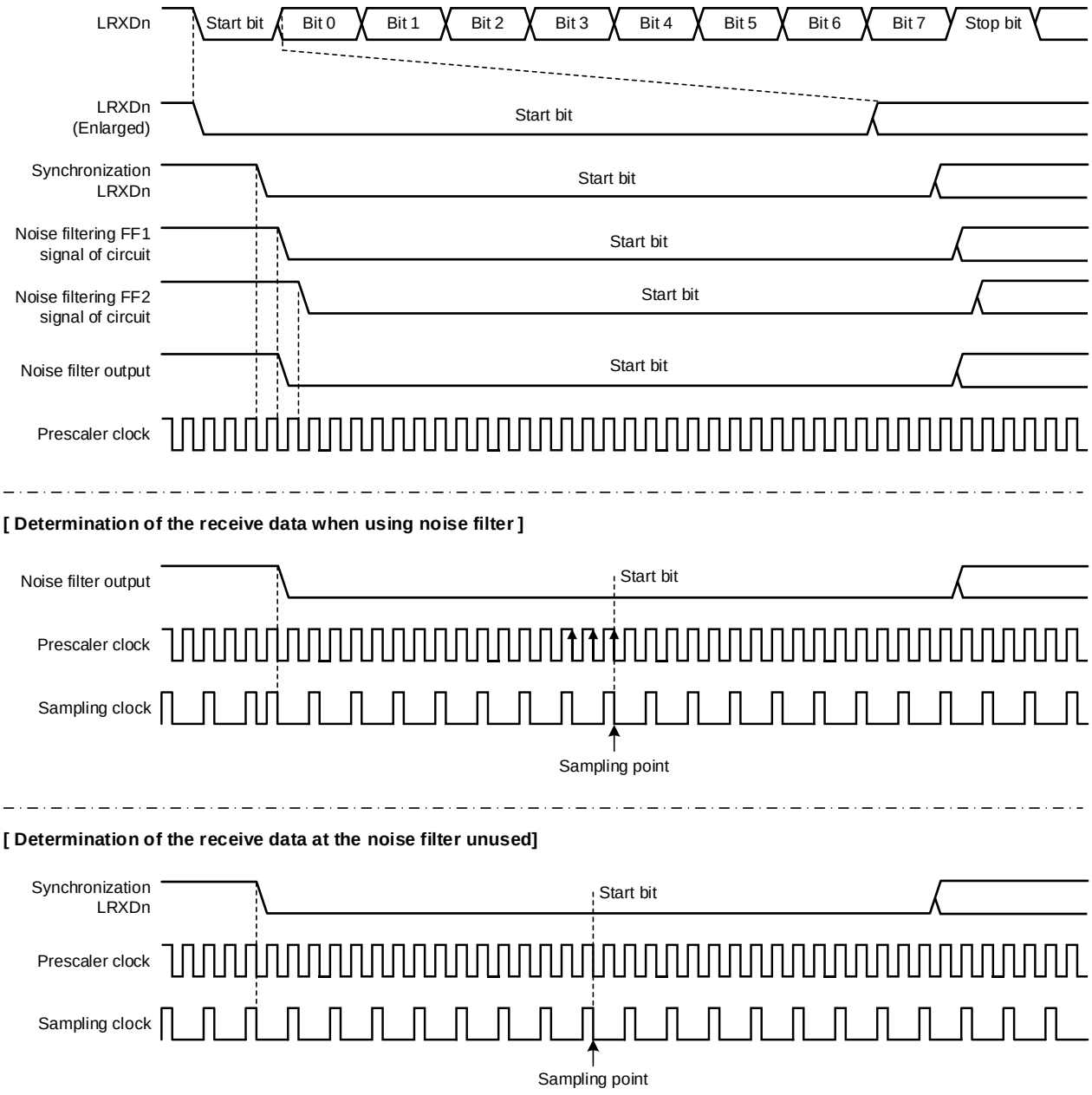


Figure 18-46. Determination of Received Data when Noise Filter is Used



18.9 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 18-29 lists the sources for each interrupt.

Table 18-29. 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 • Successful frame reception • Successful wake-up transmission • Successful wake-up reception • Successful header transmission • Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error	• Successful frame transmission • Successful wake-up transmission • Successful header transmission	• Successful frame reception • Successful wake-up reception	• Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error
	LIN slave mode	• Successful response transmission • Successful response reception • Successful wake-up transmission • Successful wake-up reception • Successful header reception • Bit error • Frame/response timeout error • Framing error • Sync field error • Checksum error • ID parity error • Response preparation error	• Successful response transmission • Successful wake-up transmission	• Successful response reception • Successful wake-up reception • Successful header reception	• Bit error • Frame/response timeout error • Framing error • Sync field error • Checksum error • ID parity error • Response preparation error
UART mode		–	• Transmission start/successful transmission	• Successful reception • Expansion bit mismatch	• Bit error • Overrun error • Framing error • Expansion bit detection • ID match • 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/F22 and RL78/F25, the LINn reception status interrupt and LINn interrupt are assigned to the same interrupt vector.

CHAPTER 19 CAN INTERFACE (RS-CANFD lite) (RL78/F25 Only)

The number of CAN module channels depends on the product.

Product	RL78/F25	RL78/F22
Number of channels	2	–

The frequency relation should be set to $f_{MP}/2 = f_{CLK} \geq f_{CAN}$.

Remark f_{MP}: PLL or MAIN system clock frequency (max 80 MHz)
f_{CLK}: CPU/Periferal hardware clock frequency
f_{CAN}: CAN communication clock

19.1 Overview

The RL78/F25 incorporate 2 channels of the Controller Area Network with Flexible Data Rate (CANFD) module conforming to the ISO11898-1(2015) specifications. **Table 19-1** shows the CAN module specifications. **Figure 19-1** shows the CAN module block diagram.

In this chapter, the following suffix indicates the number of channels or registers.

- s [s = 0, 1]: Channel number
- i [i = 0 to 15]: AFL (Acceptance Filter List) register number
- j [j = 0, 1]: PNF (Pretended Network Filter List) register number
- k [k = 0, 1]: Receive FIFO buffer number
- m [m = 0 to 3]: Transmit buffer number
- n [n = 0 to 15]: Receive buffer number
- p [p = 0 to 15]: Data Field register number
- r [r = 0 to 63]: RAM Test Page register (RPGACCr) number

Remark In register symbol notation in this chapter, L/H notation and channel number “s” are basically omitted.
(Example: **GIPVHs**, **GIPVLs** → **GIPV** register)
However, register names that originally had **CAN0** (or Channel 0) are written as **CANs** with the channel number “s” added.

Table 19-1. RS-CANFD lite Module Specifications (1/2)

Item	Specification
Number of channels	2
Protocol	ISO11898-1(2015) compliant
Communication speed	• Classical-CAN only mode Maximum 1 Mbps Communication speed (CAN bit time clock) = $\frac{1}{\text{(CAN bit time)}}$ CAN bit time = CANTq × Tq count per bit CANTq = $\frac{(\textbf{CsNCFG.NBRP}[\textbf{9:0}] + 1)}{f_{CAN}}$ • CAN-FD mode and CAN-FD only mode Data bit rate: max. 5 Mbps (Nominal bit rate ≤ 1 Mbps) **Note** Note: f_{CAN} max = 40 MHz Transmission rate (CAN nominal bit time clock) = 1 / (CAN nominal bit time) Transmission rate (CAN data bit time clock) = 1/ (CAN data bit time) CAN nominal bit time = CAN Tq (N) × Tq count per nominal bit CAN data bit time = CAN Tq (D) × Tq count per data bit CAN Tq (N) = (**CsNCFG.NBRP[9:0]** + 1) / f_{CAN} CAN Tq (D) = (**CsDCFG.DBRP[7:0]** + 1) / f_{CAN} Tq: Time quantum f_{CAN}: Frequency of CAN communication clock (selected by the **GCFGL.DCS**); max 40 MHz
Buffer	Total 20 buffers per channel • Individual buffers for each channel: 4 buffers per channel Transmit buffer: 4 buffers per channel • Shared buffers between channels: 16 buffers per channel Receive buffer: 0 to 16 buffers per channel Receive FIFO buffer: 2 FIFO buffers per channel (up to 16 buffers allocatable to each) Common FIFO buffer: 1 FIFO buffer per channel (up to 16 buffers allocatable to each) • RAM ECC included
Reception function	• Receives data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be received. • Sets interrupt enable/disable for each FIFO. • Mirror function (to receive messages transmitted from the own CAN node) • Timestamp function (to record message reception time as a 16-bit timer value)
Reception filter function	• Selects receive messages according to 16 receive rules per channel. • Sets the number of receive rules (0 to 16) for each channel. • Acceptance filter processing: Sets ID and mask for each receive rule. • DLC (Data Length Code) 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). Transfer destination: Receive buffer, receive FIFO buffer, and Common FIFO buffer. • Label addition function Stores label information together when storing a message in a receive buffer and FIFO buffer.
Transmission function	• Transmits data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be transmitted. • Sets interrupt enable/disable for each transmit buffer and Common FIFO buffer. • Selects ID priority transmission or transmit buffer number priority transmission. • Transmit request can be aborted (possible to confirm with the flag) • One-shot transmission function

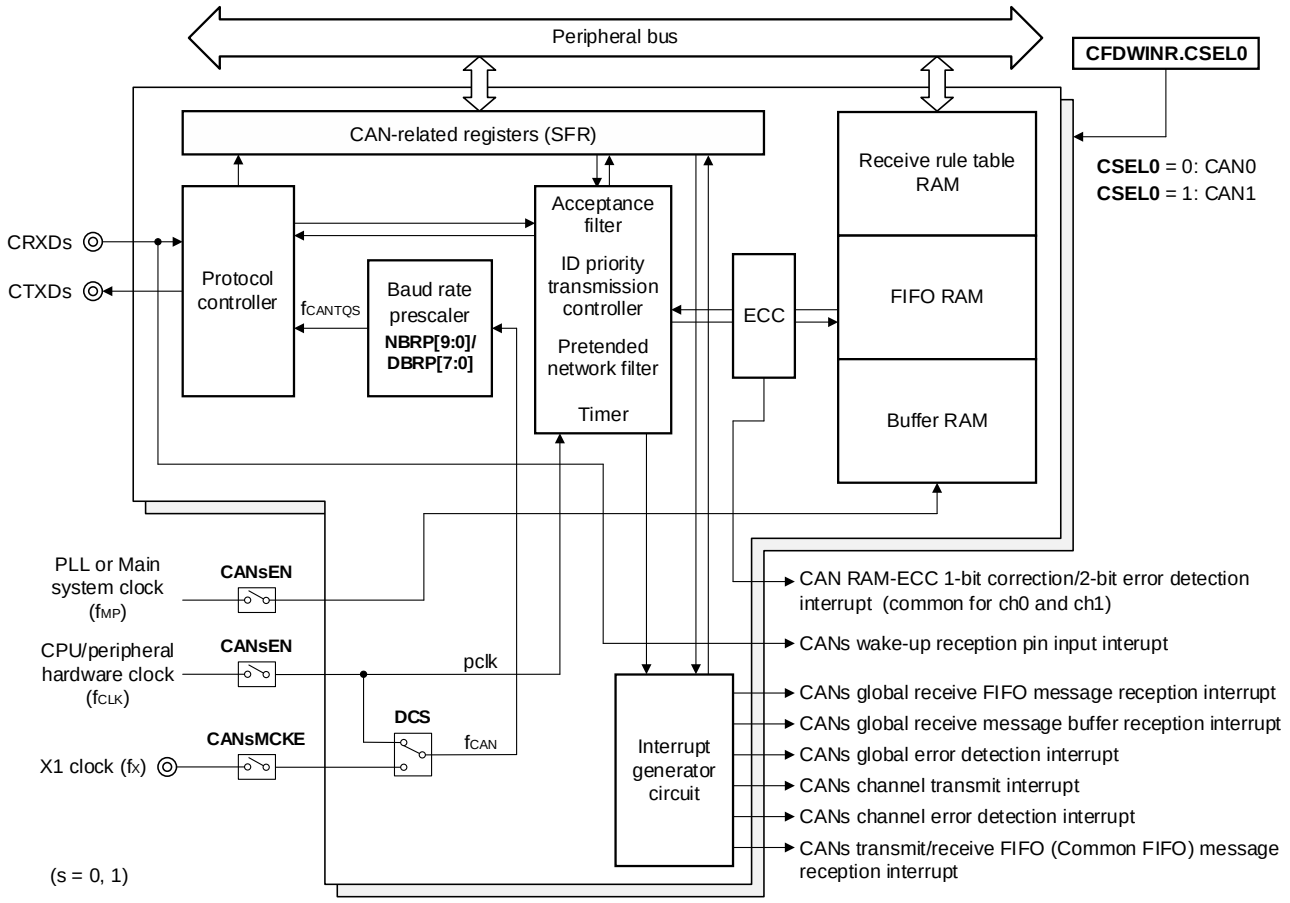
Table 19-1. RS-CANFD lite Module Specifications (2/2)

Item	Specification
Interval transmission function	Sets message transmission interval time (transmit mode of Common FIFO buffers)
Transmit history function	Stores the history information of transmitted messages.
Bus off recovery mode selection	Selects a method of returning from bus off state. • ISO11898-1 compliant • Automatic entry to channel halt mode at bus-off entry • Automatic entry to channel halt mode at bus-off end • Entry to channel halt mode by a program • 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). • Detects error status transitions (error warning, error passive, bus off entry, and bus off recovery) • Reads the error counter. • Monitors DLC errors.
Interrupt source	Total: 15 sources (7 sources × 2 channels + 1 source) • Global (3 sources/channel) - CANs global receive FIFO message reception interrupt - CANs global error detection interrupt - CANs global receive message buffer reception interrupt • Channel (3 sources/channel) - CANs channel transmit CANs transmit complete interrupt CANs transmit abort interrupt CANs transmit/receive FIFO (Common FIFO) transmit complete interrupt CANs transmit history interrupt - CANs transmit/receive FIFO (Common FIFO) message reception interrupt - CANs channel error detection interrupt • Other (2 sources) - CANs wakeup reception pin input interrupt (exists independently for each channel) - CAN RAM-ECC 1-bit correction/2-bit error detection interrupt (common for ch0 and ch1)
CAN stop mode	Reduces power consumption by stopping clock supply to the CAN module.
CAN clock source Note	Selects the clock obtained by frequency-f_{CLK} or the X1 clock (f_X).
Test function	Test function for user evaluation • Basic test mode • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback) • Restricted operation mode • RAM test (read/write test) • Bit flip test [CRC error test enabled]

Note High-speed on-chip oscillator clock (f_{IH}) cannot be selected for f_{CAN} (CAN communication clock source).

Remark s: CAN channel number (s = 0, 1)

Figure 19-1. CAN Module Block Diagram



Remark **NBRP[9:0]**: Bits in the **CsNCFG** register (Channel Nominal Baud Rate Prescaler)
DBRP[7:0]: Bits in the **CsDCFG** register (Channel Data Baud Rate Prescaler)
DCS: Bit in the **GCFG** register (Data Link Controller Clock Select)
f_{CANTQS}: CANs Tq clock
f_{CAN}: CAN communication clock
CANsEN: Bit in the **PER2** register (Refer to **Chapter 5. Clock Generator**)
CANsMCKE: Bit in the **CANCKSEL** register (Refer to **Chapter 5. Clock Generator**)

Caution The frequency relation should be set to $f_{MP}/2 = f_{CLK} \geq f_{CAN}$. (Example: f_{MP} = 80 MHz, f_{CLK} = 40 MHz, X1 = 20 MHz.)

19.2 Input/Output Pins

Table 19-2 lists the I/O pins of the CAN module.

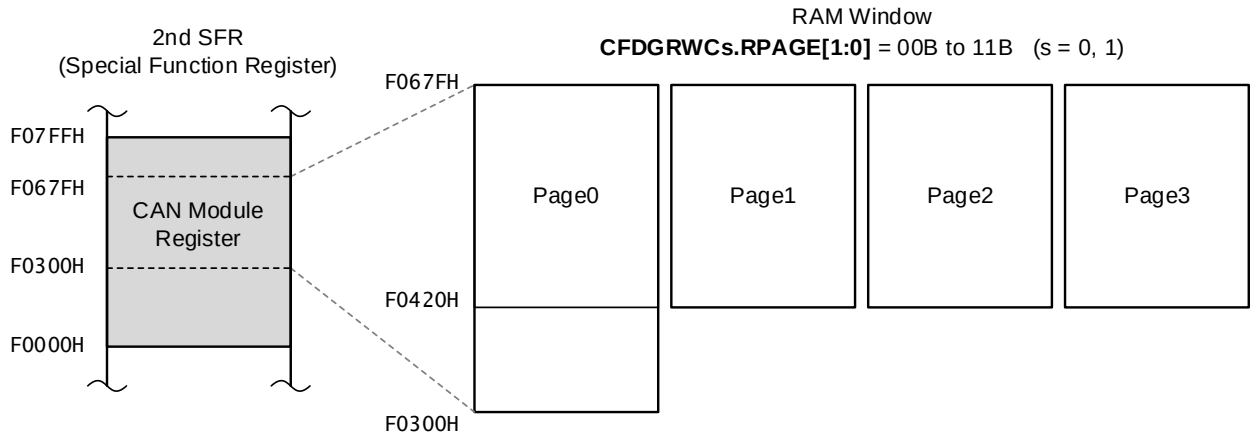
Table 19-2. I/O Pins of the CAN Module

Pin Name	I/O	Description
CRXD0	Input	Receive data input pins of the CAN0 communication function
CTXD0	Output	Transmit data output pins of the CAN0 communication function
CRXD1	Input	Receive data input pins of the CAN1 communication function
CTXD1	Output	Transmit data output pins of the CAN1 communication function

19.3 Register Descriptions

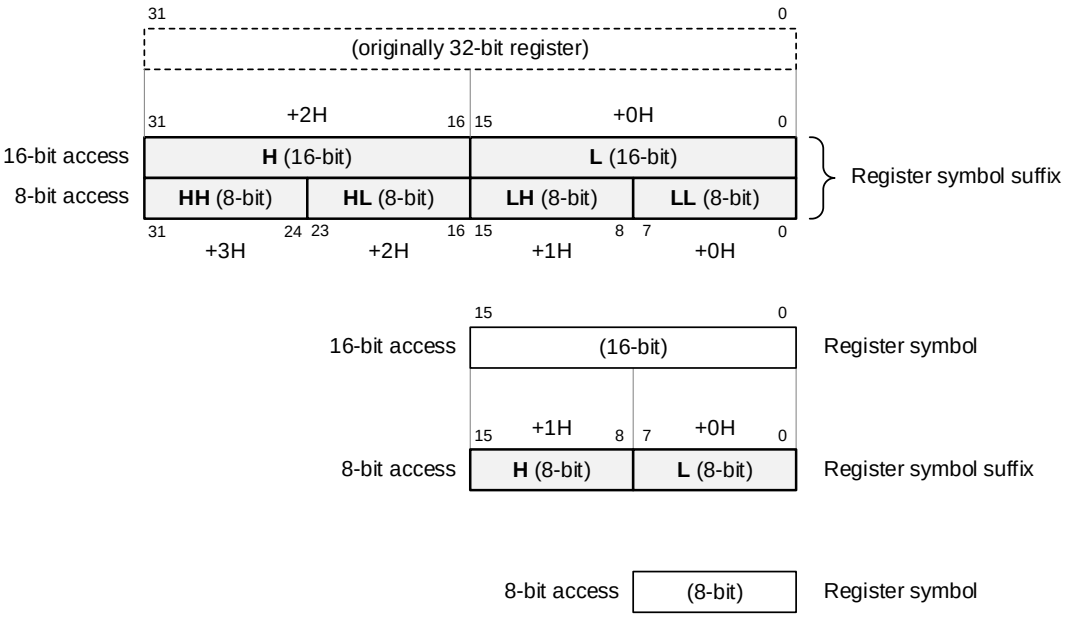
The RS-CANFD lite module registers are located in the 2nd SFR area. The area from address F0420H to address F067FH is switched from page 0 to page 3 by setting the **CFDGRWCs** register.

Figure 19-2. RS-CANFD lite Module Register Address Map



The RS-CANFD lite module has 32-bit length registers. When accessing a 32-bit length register from the CPU, access it as a 16-bit length register (register H, register L) or an 8-bit length register (register HH/HL, register LH/LL).

Figure 19-3. Register Placement and Address



The values after reset of the registers allocated to the CAN RAM area (F0420H to F067FH) are those after the CAN RAM is initialized.

Refer to **19.15.2 CAN Module Configuration After Reset** for details of the initialization process.

Also for the RAM area, if a write access with a size of 8 or 16 bits is done then the RS-CANFD lite module does a read modify write access to the RAM location.

In case of single bit error the correct data will be written back.

In case of multiple bit errors unknown data will be written back.

Detail of ECC is referring to **29.3.4 CAN RAM-ECC Function (RL78/F25 Only)**.

To access the space where the register is not assigned is prohibited.

The read data from the space where the register is not assigned is unknown.

Table 19-3 to **Table 19-7**, starting on the next page, lists the registers of the CAN module.

Table 19-3. List of CAN Module Registers (1/3)

Address	Register Name	Symbol		R/W	Access Size	After Reset Note
F01F7H	CANFD window access control register	CFDWINR		R/W	1, 8	00H
F02C2H	Peripheral enable register 2	PER2		R/W	1, 8	00H
F02C3H	CAN clock select register	CANCKSEL		R/W	1, 8	00H
F0300H	CANs nominal bitrate configuration register L [s = 0, 1]	CsNCFGLL	CsNCFGL	R/W	8, 16	0000H
F0301H		CsNCFGLH				
F0302H	CANs nominal bitrate configuration register H [s = 0, 1]	CsNCFGHL	CsNCFGH	R/W	8, 16	0000H
F0303H		CsNCFGHH				
F0304H	CANs control register L [s = 0, 1]	CsCTRLL	CsCTRL	R/W	8, 16	0005H
F0305H		CsCTRLH				
F0306H	CANs control register H [s = 0, 1]	CsCTRHL	CsCTRH	R/W	8, 16	0000H
F0307H		CsCTRHH				
F0308H	CANs status register L [s = 0, 1]	CsSTSLL	CsSTSL	R/W	8, 16	0005H
F0309H		CsSTSLH				
F030AH	CANs status register H [s = 0, 1]	CsSTSHL	CsSTSH	R/W	8, 16	0000H
F030BH		CsSTSHH				
F030CH	CANs error flag register L [s = 0, 1]	CsERFLLL	CsERFLL	R/W	8, 16	0000H
F030DH		CsERFLLH				
F030EH	CANs error flag register H [s = 0, 1]	CsERFLHL	CsERFLH	R	8, 16	0000H
F030FH		CsERFLHH				
F0310H	Global IP version register L [s = 0, 1]	GIPVLsL	GIPVLs	R	8, 16	8143H
F0311H		GIPVLsH				
F0312H	Global IP version register H [s = 0, 1]	GIPVHsL	GIPVHs	R	8, 16	3C8BH
F0313H		GIPVHsH				
F0314H	Global configuration register L [s = 0, 1]	GCFGLsL	GCFGLs	R/W	8, 16	0000H
F0315H		GCFGLsH				
F0316H	Global configuration register H [s = 0, 1]	GCFGHsL	GCFGHs	R/W	8, 16	0000H
F0317H		GCFGHsH				
F0318H	Global control register L [s = 0, 1]	GCTRLsL	GCTRLs	R/W	8, 16	0000H
F0319H		GCTRLsH				
F031AH	Global control register H [s = 0, 1]	GCTRHsL	GCTRHs	R/W	8, 16	0000H
F031BH		–				
F031CH	Global status register [s = 0, 1]	GSTSsL	GSTSs	R	8, 16	000DH
F031DH		–				
F0320H	Global error flag register L [s = 0, 1]	GERFLLsL	GERFLLs	R/W	8, 16	0000H
F0321H		–				
F0322H	Global error flag register H [s = 0, 1]	GERFLHsL	GERFLHs	R/W	8, 16	0000H
F0323H		–				
F0324H	Global timestamp counter register [s = 0, 1]	GTSCs		R	16	0000H
F0328H	Global acceptance filter list entry control register [s = 0, 1]	–	GAFLECTRs	R/W	8, 16	0000H
F0329H		GAFLECTRsH				
F032EH	Global acceptance filter list configuration register [s = 0, 1]	GAFLCFGsL	GAFLCFGs	R/W	8, 16	0000H
F032FH		–				
F0330H	RX message buffer number register [s = 0, 1]	RMNBsL	RMNBs	R/W	8, 16	0000H
F0331H		RMNBsH				
F0334H	RX message buffer new data register [s = 0, 1]	RMNDsL	RMNDs	R/W	8, 16	0000H
F0335H		RMNDsH				
F0338H	RX message buffer interrupt enable configuration register [s = 0, 1]	RMIECsL	RMIECs	R/W	8, 16	0000H
F0339H		RMIECsH				

Note When the **CANsEN** bit in the **PER2** register is 0, the read value is undefined. When the **CANsEN** bit is 1, the reading from the addresses from F0300H to F0413H is the above initial value. [s = 0, 1]

Table 19-3. List of CAN Module Registers (2/3)

Address	Register Name	Symbol		R/W	Access Size	After Reset Note
F033CH + (4 × k)	RX FIFO configuration / control register k [k = 0, 1], [s = 0, 1]	RFCCksL	RFCCks	R/W	8, 16	0000H
		RFCCksH				
F0344H + (4 × k)	RX FIFO status register k [k = 0, 1], [s = 0, 1]	RFSTSksL	RFSTSks	R/W	8, 16	0000H
		RFSTSksH				
F034CH + (4 × k)	RX FIFO pointer control register k [k = 0, 1], [s = 0, 1]	RFPCTRksL	RFPCTRks	W	8, 16	0000H
		–				
F0354H	Common FIFO configuration / control register L [s = 0, 1]	CFCCLsL	CFCCLs	R/W	8, 16	0000H
F0355H		CFCCLsH				
F0356H	Common FIFO configuration / control register H [s = 0, 1]	CFCCHsL	CFCCHs	R/W	8, 16	0000H
F0357H		CFCCHsH				
F0358H	Common FIFO status register [s = 0, 1]	CFSTSsL	CFSTSs	R/W	8, 16	0000H
F0359H		CFSTSsH				
F035CH	Common FIFO pointer control register [s = 0, 1]	CFPCTRsL	CFPCTRs	W	8, 16	0000H
F035DH		–				
F0360H	FIFO empty status register [s = 0, 1]	FESTSsL	FESTSs	R	8, 16	0103H
F0361H		FESTSsH				
F0364H	FIFO full status register [s = 0, 1]	FFSTSsL	FFSTSs	R	8, 16	0000H
F0365H		FFSTSsH				
F0368H	FIFO message lost status register [s = 0, 1]	FMSTSsL	FMSTSs	R	8, 16	0000H
F0369H		FMSTSsH				
F036CH	RX FIFO interrupt flag status register [s = 0, 1]	RFISTSsL	RFISTSs	R	8, 16	0000H
F036DH		–				
F0370H + m	CANs TX message buffer control register m [m = 0 to 3], [s = 0, 1]	TMCms		R/W	8	00H
F0374H + m	CANs TX message buffer status register m [m = 0 to 3], [s = 0, 1]	TMSTSms		R/W	8	00H
F0378H	CANs TX message buffer transmission request status register [s = 0, 1]	TMTRSTSsL	TMTRSTSs	R	8, 16	0000H
F0379H		–				
F037CH	CANs TX message buffer transmission abort request status register [s = 0, 1]	TMTARSTSsL	TMTARSTSs	R	8, 16	0000H
F037DH		–				
F0380H	CANs TX message buffer transmission completion status register [s = 0, 1]	TMTCSTSsL	TMTCSTSs	R	8, 16	0000H
F0381H		–				
F0384H	CANs TX message buffer transmission abort status register [s = 0, 1]	TMTASTSsL	TMTASTSs	R	8, 16	0000H
F0385H		–				
F0388H	CANs TX message buffer interrupt enable configuration register [s = 0, 1]	TMIECsL	TMIECs	R/W	8, 16	0000H
F0389H		–				
F0398H	CANs TX history list configuration / control register [s = 0, 1]	THLCCsL	THLCCs	R/W	8, 16	0000H
F0399H		THLCCsH				
F039CH	CANs TX history list status register [s = 0, 1]	THLSTSsL	THLSTSs	R/W	8, 16	0001H
F039DH		THLSTSsH				
F03A0H	CANs TX history list pointer control register [s = 0, 1]	THLPCTRsL	THLPCTRs	W	8, 16	0000H
F03A1H		–				
F03A4H	Global TX interrupt status register [s = 0, 1]	GTINTSTSsL	GTINTSTSs	R	8, 16	0000H
F03A5H		–				
F03AAH	Global test configuration register [s = 0, 1]	GTSTCFGsL	GTSTCFGs	R/W	8, 16	0000H
F03ABH		–				
F03ACH	Global test control register [s = 0, 1]	GTSTCTRsL	GTSTCTRs	R/W	8, 16	0000H
F03ADH		–				

Note When the **CANsEN** bit in the **PER2** register is 0, the read value is undefined. When the **CANsEN** bit is 1, the reading from the addresses from F0300H to F0413H is the above initial value. [s = 0, 1]

Table 19-3. List of CAN Module Registers (3/3)

Address	Register Name	Symbol		R/W	Access Size	After Reset Note
F03B0H	Global FD configuration register [s = 0, 1]	GFDCFGsL	GFDCFGs	R/W	8, 16	0000H
F03B1H		GFDCFGsH				
F03B8H	Global lock key register [s = 0, 1]	GLOCKKs		W	16	0000H
F03C0H	Global AFL ignore entry register [s = 0, 1]	GAFLIGENTsL	GAFLIGENTs	R/W	8, 16	0000H
F03C1H		–				
F03C4H	Global AFL ignore control register [s = 0, 1]	GAFLIGCTRs		R/W	16	0000H
F03D0H	Global pretended network filter list entry control register [s = 0, 1]	–	GPFLECTRs	R/W	8, 16	0000H
F03D1H		GPFLECTRsH				
F03D6H	Global pretended network filter list configuration register [s = 0, 1]	–	GPFLCFGs	R/W	8, 16	0000H
F03D7H		GPFLCFGsH				
F03D8H	Global reset control register [s = 0, 1]	GRSTCs		R/W	16	0000H
F03DCH	RS-CANFD lite global RAM window control register [s = 0, 1]	CFDGRWCs		R/W	16	0000H
F0400H	CANs data bit rate configuration register L [s = 0, 1]	CsDCFGLL	CsDCFGL	R/W	8, 16	0000H
F0401H		CsDCFGLH				
F0402H	CANs data bit rate configuration register H [s = 0, 1]	CsDCFGHL	CsDCFGH	R/W	8, 16	0000H
F0403H		CsDCFGHH				
F0404H	CANs CAN-FD configuration register L [s = 0, 1]	CsFDCFGLL	CsFDCFGL	R/W	8, 16	0000H
F0405H		CsFDCFGLH				
F0406H	CANs CAN-FD configuration register H [s = 0, 1]	CsFDCFGHL	CsFDCFGH	R/W	8, 16	0000H
F0407H		CsFDCFGHH				
F0408H	CANs CAN-FD control register L [s = 0, 1]	CsFDCTRLL	CsFDCTRL	R/W	8, 16	0000H
F0409H		–				
F040AH	CANs CAN-FD control register H [s = 0, 1]	CsFDCTRH		W	16	0000H
F040CH	CANs CAN-FD status register L [s = 0, 1]	CsFDSTSLL	CsFDSTSL	R/W	8, 16	0000H
F040DH		CsFDSTSLH				
F040EH	CANs CAN-FD status register H [s = 0, 1]	CsFDSTSHL	CsFDSTSH	R	8, 16	0000H
F040FH		CsFDSTSHH				
F0410H	CANs CAN-FD CRC register L [s = 0, 1]	CsFDCRCL		R	16	0000H
F0412H	CANs CAN-FD CRC register H [s = 0, 1]	CsFDCRCH		R	16	0000H

Note When the **CANsEN** bit in the **PER2** register is 0, the read value is undefined. When the **CANsEN** bit is 1, the reading from the addresses from F0300H to F0413H is the above initial value. [s = 0, 1]

Table 19-4. List of CAN Module Registers on Page.0 Notes 1, 2 (1/2)

Address	Register Name	Symbol		R/W	Access Size	After Reset
F0420H + (10H × i)	Global acceptance filter list ID register iLs [i = 0 to 15], [s = 0, 1]	GAFLIDiLsL	GAFLIDiLs	R/W	8, 16	0000H
		GAFLIDiLsH				
F0422H + (10H × i)	Global acceptance filter list ID register iHs [i = 0 to 15], [s = 0, 1]	GAFLIDiHsL	GAFLIDiHs	R/W	8, 16	0000H
		GAFLIDiHsH				
F0424H + (10H × i)	Global acceptance filter list mask register iLs [i = 0 to 15], [s = 0, 1]	GAFLMiLsL	GAFLMiLs	R/W	8, 16	0000H
		GAFLMiLsH				
F0426H + (10H × i)	Global acceptance filter list mask register iHs [i = 0 to 15], [s = 0, 1]	GAFLMiHsL	GAFLMiHs	R/W	8, 16	0000H
		GAFLMiHsH				
F0428H + (10H × i)	Global acceptance filter list pointer 0 register iLs [i = 0 to 15], [s = 0, 1]	GAFLP0iLsL	GAFLP0iLs	R/W	8, 16	0000H
		GAFLP0iLsH				
F042AH + (10H × i)	Global acceptance filter list pointer 0 register iHs [i = 0 to 15], [s = 0, 1]	GAFLP0iHsL	GAFLP0iHs	R/W	8, 16	0000H
		GAFLP0iHsH				
F042CH + (10H × i)	Global acceptance filter list pointer 1 register iLs [i = 0 to 15], [s = 0, 1]	GAFLP1iLsL	GAFLP1iLs	R/W	8, 16	0000H
		GAFLP1iLsH				
F0520H + (24H × j)	Global pretended network filter list ID register jLs [j = 0, 1], [s = 0, 1]	GPFLIDjLsL	GPFLIDjLs	R/W	8, 16	0000H
		GPFLIDjLsH				
F0522H + (24H × j)	Global pretended network filter list ID register jHs [j = 0, 1], [s = 0, 1]	GPFLIDjHsL	GPFLIDjHs	R/W	8, 16	0000H
		GPFLIDjHsH				
F0524H + (24H × j)	Global pretended network filter list mask register jLs [j = 0, 1], [s = 0, 1]	GPFLMjLsL	GPFLMjLs	R/W	8, 16	0000H
		GPFLMjLsH				
F0526H + (24H × j)	Global pretended network filter list mask register jHs [j = 0, 1], [s = 0, 1]	GPFLMjHsL	GPFLMjHs	R/W	8, 16	0000H
		GPFLMjHsH				
F0528H + (24H × j)	Global pretended network filter list pointer 0 register jLs [j = 0, 1], [s = 0, 1]	GPFLP0jLsL	GPFLP0jLs	R/W	8, 16	0000H
		GPFLP0jLsH				
F052AH + (24H × j)	Global pretended network filter list pointer 0 register jHs [j = 0, 1], [s = 0, 1]	GPFLP0jHsL	GPFLP0jHs	R/W	8, 16	0000H
		GPFLP0jHsH				
F052CH + (24H × j)	Global pretended network filter list pointer 1 register jLs [j = 0, 1], [s = 0, 1]	GPFLP1jLsL	GPFLP1jLs	R/W	8, 16	0000H
		GPFLP1jLsH				
F0530H + (24H × j)	Global pretended network filter list payload type register jLs [j = 0, 1], [s = 0, 1]	GPFLPTjLsL	GPFLPTjLs	R/W	8, 16	0000H
		GPFLPTjLsH				
F0532H + (24H × j)	Global pretended network filter list payload type register jHs [j = 0, 1], [s = 0, 1]	GPFLPTjHsL	GPFLPTjHs	R/W	8, 16	0000H
		GPFLPTjHsH				
F0534H + (24H × j)	Global pretended network filter list payload data 0 register jLs [j = 0, 1], [s = 0, 1]	GPFLPD0jLsL	GPFLPD0jLs	R/W	8, 16	0000H
		GPFLPD0jLsH				
F0536H + (24H × j)	Global pretended network filter list payload data 0 register jHs [j = 0, 1], [s = 0, 1]	GPFLPD0jHsL	GPFLPD0jHs	R/W	8, 16	0000H
		GPFLPD0jHsH				
F0538H + (24H × j)	Global pretended network filter list payload mask 0 register jLs [j = 0, 1], [s = 0, 1]	GPFLPM0jLsL	GPFLPM0jLs	R/W	8, 16	0000H
		GPFLPM0jLsH				
F053AH + (24H × j)	Global pretended network filter list payload mask 0 register jHs [j = 0, 1], [s = 0, 1]	GPFLPM0jHsL	GPFLPM0jHs	R/W	8, 16	0000H
		GPFLPM0jHsH				
F053CH + (24H × j)	Global pretended network filter list payload data 1 register jLs [j = 0, 1], [s = 0, 1]	GPFLPD1jLsL	GPFLPD1jLs	R/W	8, 16	0000H
		GPFLPD1jLsH				
F053EH + (24H × j)	Global pretended network filter list payload data 1 register jHs [j = 0, 1], [s = 0, 1]	GPFLPD1jHsL	GPFLPD1jHs	R/W	8, 16	0000H
		GPFLPD1jHsH				
F0540H + (24H × j)	Global pretended network filter list payload mask 1 register jLs [j = 0, 1], [s = 0, 1]	GPFLPM1jLsL	GPFLPM1jLs	R/W	8, 16	0000H
		GPFLPM1jLsH				
F0542H + (24H × j)	Global pretended network filter list payload mask 1 register jHs [j = 0, 1], [s = 0, 1]	GPFLPM1jHsL	GPFLPM1jHs	R/W	8, 16	0000H
		GPFLPM1jHsH				

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

Table 19-4. List of CAN Module Registers on Page.0 Notes 1, 2 (2/2)

Address	Register Name	Symbol		R/W	Access Size	After Reset
F0580H + (4 × r)	RAM test page access register rLs [r = 0 to 63], [s = 0, 1]	RPGACCrLsL	RPGACCrLs	R/W	8, 16	0000H
		RPGACCrLsH				
F0582H + (4 × r)	RAM test page access register rHs [r = 0 to 63], [s = 0, 1]	RPGACCrHsL	RPGACCrHs	R/W	8, 16	0000H
		RPGACCrHsH				

Notes 1. Set **CFDGRWCs.RPAGE[1:0]** bits to 00B.

2. When the **CANsEN** bit in the **PER2** register is 0 or CAN RAM initialization has not been performed, the read value is undefined. When the **CANsEN** bit is 1 and the initialization of the CAN RAM area is completed, the read value is the initial value listed above. [s = 0, 1]

Table 19-5. List of CAN Module Registers on Page.1 Notes 1, 2 (1/2)

Address	Register Name	Symbol		R/W	Access Size	After Reset
F0420H + (4CH × k)	RX FIFO access ID register kLs [k = 0, 1], [s = 0, 1]	RFIDkLsL	RFIDkLs	R	8, 16	0000H
		RFIDkLsH				
F0422H + (4CH × k)	RX FIFO access ID register kHs [k = 0, 1], [s = 0, 1]	RFIDkHsL	RFIDkHs	R	8, 16	0000H
		RFIDkHsH				
F0424H + (4CH × k)	RX FIFO access pointer register kLs [k = 0, 1], [s = 0, 1]	RFPTRkLsL	RFPTRkLs	R	8, 16	0000H
		RFPTRkLsH				
F0426H + (4CH × k)	RX FIFO access pointer register kHs [k = 0, 1], [s = 0, 1]	–	RFPTRkHs	R	8, 16	0000H
		RFPTRkHsH				
F0428H + (4CH × k)	RX FIFO access CAN-FD status register kLs [k = 0, 1], [s = 0, 1]	RFFDSTSkLsL	RFFDSTSkLs	R	8, 16	0000H
		RFFDSTSkLsH				
F042AH + (4CH × k)	RX FIFO access CAN-FD status register kHs [k = 0, 1], [s = 0, 1]	RFFDSTSkHsL	RFFDSTSkHs	R	8, 16	0000H
		RFFDSTSkHsH				
F042CH + (4CH × k) + (4 × p)	RX FIFO access data field p register kLs [k = 0, 1], [p = 0 to 15], [s = 0, 1]	RFDFk_pLsL	RFDFk_pLs	R	8, 16	0000H
		RFDFk_pLsH				
F042EH + (4CH × k) + (4 × p)	RX FIFO access data field p register kHs [k = 0, 1], [p = 0 to 15], [s = 0, 1]	RFDFk_pHsL	RFDFk_pHs	R	8, 16	0000H
		RFDFk_pHsH				
F04B8H	Common FIFO access ID register L [s = 0, 1]	CFIDLsL	CFIDLs	R/W	8, 16	0000H
F04B9H		CFIDLsH				
F04BAH	Common FIFO access ID register H [s = 0, 1]	CFIDHsL	CFIDHs	R/W	8, 16	0000H
F04BBH		CFIDHsH				
F04BCH	Common FIFO access pointer register L [s = 0, 1]	CFPTRLsL	CFPTRLs	R/W	8, 16	0000H
F04BDH		CFPTRLsH				
F04BEH	Common FIFO access pointer register H [s = 0, 1]	–	CFPTRHs	R/W	8, 16	0000H
F04BFH		CFPTRHHs				
F04C0H	Common FIFO access CAN-FD control / status register L [s = 0, 1]	CFFDCSTSLsL	CFFDCSTSLs	R/W	8, 16	0000H
F04C1H		CFFDCSTSLsH				
F04C2H	Common FIFO access CAN-FD control / status register H [s = 0, 1]	CFFDCSTSHsL	CFFDCSTSHs	R/W	8, 16	0000H
F04C3H		CFFDCSTSHsH				
F04C4H + (4 × p)	Common FIFO access data field p register L [p = 0 to 15], [s = 0, 1]	CFDFpLsL	CFDFpLs	R/W	8, 16	0000H
		CFDFpLsH				
F04C6H + (4 × p)	Common FIFO access data field p register H [p = 0 to 15], [s = 0, 1]	CFDFpHsL	CFDFpHs	R/W	8, 16	0000H
		CFDFpHsH				
F0504H + (4CH × m)	CANs TX message buffer ID register mL [m = 0 to 3], [s = 0, 1]	TMIDmLsL	TMIDmLs	R/W	8, 16	0000H
		TMIDmLsH				
F0506H + (4CH × m)	CANs TX message buffer ID register mH [m = 0 to 3], [s = 0, 1]	TMIDmHsL	TMIDmHs	R/W	8, 16	0000H
		TMIDmHsH				

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

Table 19-5. List of CAN Module Registers on Page.1 Notes 1, 2 (2/2)

Address	Register Name	Symbol		R/W	Access Size	After Reset
F050AH + (4CH × m)	CANs TX message buffer pointer register mH [m = 0 to 3], [s = 0, 1]	–	TMPTRmHs	R/W	8, 16	0000H
		TMPTRmHsH				
F050CH + (4CH × m)	CANs TX message buffer CAN-FD control register mL [m = 0 to 3], [s = 0, 1]	TMFDCTRmLsL	TMFDCTRmLs	R/W	8, 16	0000H
		TMFDCTRmLsH				
F050EH + (4CH × m)	CANs TX message buffer CAN-FD control register mH [m = 0 to 3], [s = 0, 1]	TMFDCTRmHsL	TMFDCTRmHs	R/W	8, 16	0000H
		TMFDCTRmHsH				
F0510H + (4CH × m) + (4 × p)	CANs TX message buffer data field p register m L [m = 0 to 3], [p = 0 to 15], [s = 0, 1]	TMDFm_pLsL	TMDFm_pLs	R/W	8, 16	0000H
		TMDFm_pLsH				
F0512H + (4CH × m) + (4 × p)	CANs TX message buffer data field p register m H [m = 0 to 3],]p = 0 to 15], [s = 0, 1]	TMDFm_pHsL	TMDFm_pHs	R/W	8, 16	0000H
		TMDFm_pHsH				
F0640H	CANs TX history list access register 0L [s = 0, 1]	THLACC0LsL	THLACC0Ls	R	8, 16	0000H
F0641H		–				
F0642H	CANs TX history list access register 0H [s = 0, 1]	THLACC0HsL	THLACC0Hs	R	8, 16	0000H
F0643H		THLACC0HsH				
F0644H	CANs TX history list access register 1L [s = 0, 1]	THLACC1LsL	THLACC1Ls	R	8, 16	0000H
F0645H		THLACC1LsH				
F0646H	CANs TX history list access register 1H [s = 0, 1]	THLACC1HsL	THLACC1Hs	R	8, 16	0000H
F0647H		–				

Notes 1. Set **CFDGRWCs.RPAGE[1:0]** bits to 01B.

2. When the **CANsEN** bit in the **PER2** register is 0 or CAN RAM initialization has not been performed, the read value is undefined. When the **CANsEN** bit is 1 and the initialization of the CAN RAM area is completed, the read value is the initial value listed above. [s = 0, 1]

Table 19-6. List of CAN Module Registers on Page.2 Notes 1, 2

Address	Register Name	Symbol		R/W	Access Size	After Reset
F0420H + (4CH × n)	RX message buffer ID register nLs [n = 0 to 7], [s = 0, 1]	RMIDnLsL	RMIDnLs	R	8, 16	0000H
		RMIDnLsH				
F0422H + (4CH × n)	RX message buffer ID register nHs [n = 0 to 7], [s = 0, 1]	RMIDnHsL	RMIDnHs	R	8, 16	0000H
		RMIDnHsH				
F0424H + (4CH × n)	RX message buffer pointer register nLs [n = 0 to 7], [s = 0, 1]	RMPTRnLsL	RMPTRnLs	R	8, 16	0000H
		RMPTRnLsH				
F0426H + (4CH × n)	RX message buffer pointer register nHs [n = 0 to 7], [s = 0, 1]	-	RMPTRnHs	R	8, 16	0000H
		RMPTRnHsH				
F0428H + (4CH × n)	RX message buffer CAN-FD status register nLs [n = 0 to 7], [s = 0, 1]	RMSTSnLsL	RMSTSnLs	R	8, 16	0000H
		RMSTSnLsH				
F042AH + (4CH × n)	RX message buffer CAN-FD status register nHs [n = 0 to 7], [s = 0, 1]	RMSTSnHsL	RMSTSnHs	R	8, 16	0000H
		RMSTSnHsH				
F042CH + (4CH × n) + (p × 4)	RX message buffer data field p register nLs [n = 0 to 7], [p = 0 to 15], [s = 0, 1]	RMDn_pLsL	RMDn_pLs	R	8, 16	0000H
		RMDn_pLsH				
F042EH + (4CH × n) + (p × 4)	RX message buffer data field p register nHs [n = 0 to 7], [p = 0 to 15], [s = 0, 1]	RMDn_pHsL	RMDn_pHs	R	8, 16	0000H
		RMDn_pHsH				

Notes 1. Set **CFDGRWCs.RPAGE[1:0]** bits to 10B.

2. When the **CANsEN** bit in the **PER2** register is 0 or CAN RAM initialization has not been performed, the read value is undefined. When the **CANsEN** bit is 1 and the initialization of the CAN RAM area is completed, the read value is the initial value listed above. [s = 0, 1]

Table 19-7. List of CAN Module Registers on Page.3 [Notes 1, 2]

Address	Register Name	Symbol		R/W	Access Size	After Reset
F0420H + (4CH × (n -8))	RX message buffer ID register n Ls [n = 8 to 15], [s = 0, 1]	RMIDnLsL	RMIDnLs	R	8, 16	0000H
		RMIDnLsH				
F0422H + (4CH × (n-8))	RX message buffer ID register n Hs [n = 8 to 15], [s = 0, 1]	RMIDnHsL	RMIDnHs	R	8, 16	0000H
		RMIDnHsH				
F0424H + (4CH × (n-8))	RX message buffer pointer register n Ls [n = 8 to 15], [s = 0, 1]	RMPTRnLsL	RMPTRnLs	R	8, 16	0000H
		RMPTRnLsH				
F0426H + (4CH × (n-8))	RX message buffer pointer register n Hs [n = 8 to 15], [s = 0, 1]	-	RMPTRnHs	R	8, 16	0000H
		RMPTRnHsH				
F0428H + (4CH × (n-8))	RX message buffer CAN-FD status register n Ls [n = 8 to 15], [s = 0, 1]	RMSTSnLsL	RMSTSnLs	R	8, 16	0000H
		RMSTSnLsH				
F042AH + (4CH × (n-8))	RX message buffer CAN-FD status register n Hs [n = 8 to 15], [s = 0, 1]	RMSTSnHsL	RMSTSnHs	R	8, 16	0000H
		RMSTSnHsH				
F042CH + (4CH × (n-8)) + (p × 4)	RX message buffer data field p register n Ls [n = 8 to 15], [p = 0 to 15], [s = 0, 1]	RMDn_pLsL	RMDn_pLs	R	8, 16	0000H
		RMDn_pLsH				
F042EH + (4CH × (n-8)) + (p × 4)	RX message buffer data field p register n Hs [n = 8 to 15], [p = 0 to 15], [s = 0, 1]	RMDn_pHsL	RMDn_pHs	R	8, 16	0000H
		RMDn_pHsH				

Notes 1. Set **CFDGRWCs.RPAGE[1:0]** bits to 11B.

2. When the **CANsEN** bit in the **PER2** register is 0 or CAN RAM initialization has not been performed, the read value is undefined. When the **CANsEN** bit is 1 and the initialization of the CAN RAM area is completed, the read value is the initial value listed above. [s = 0, 1]

19.3.1 CANs Nominal Bitrate Configuration Register (CsNCFGH, CsNCFGL) [s = 0, 1]

Address: CsNCFGL: F0300H, CsNCFGH: F0302H
CsNCFGLL: F0300H, CsNCFGLH: F0301H, CsNCFGHL: F0302H, CsNCFGHH: F0303H

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CsNCFGH	NTSEG2[6:0]							NTSEG1[7:0]								NSJW[6]
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CsNCFGL	NSJW[5:0]						NBRP[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
31 to 25	NTSEG2[6:0]	Timing Segment 2	0000000B: Reserved 0000001B: 2 Tq : 1111110B: 127 Tq 1111111B: 128 Tq	R/W
24 to 17	NTSEG1[7:0]	Timing Segment 1	00000000B: Reserved 00000001B: 2 Tq : 11111110B: 255 Tq 11111111B: 256 Tq	R/W
16 to 10	NSJW[6:0]	Resynchronization Jump Width	0000000B: 1 Tq 0000001B: 2 Tq : 1111110B: 127 Tq 1111111B: 128 Tq	R/W
9 to 0	NBRP[9:0]	Channel Nominal Baud Rate Prescaler	Nominal Baud Rate Prescaler division ratio.	R/W

This register is used to configure the transmission / reception nominal baud rate parameters of the channels.

- **NTSEG2[6:0]**
  These bits are used to set the segment TSEG2 to compensate for edges on the CAN Bus with a negative phase error.
  These bits cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write these bits only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **NTSEG1[7:0]**
  These bits are used to set the segment TSEG1 to compensate for edges on the CAN Bus with a positive phase error.
  It also contains the propagation segment.
  These bits cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write these bits only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  See **19.15.1.2 CAN Bit Timing** for more details.

- **NSJW[6:0]**
  These bits set the synchronization jump width. A value from 1 to 128 time quantum can be set.
  These bits cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write these bits only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **NBRP[9:0]**

  When these bits are set to P (0 to 1023), the baud rate prescaler divides f_{CAN} by P + 1.
  The CANs Tq clock (f_{CANTQS}) is obtained by the CAN communication clock (f_{CAN}) and setting the clock division ratio with the NBRP[9:0] bits and one clock cycle of the CANs Tq clock is 1 Time Quantum (Tq).
  These bits cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write these bits only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.2 CANs Control Register (CsCTRH, CsCTRL) [s = 0, 1]

Address: CsCTRL: F0304H, CsCTRH: F0306H
CsCTRLL: F0304H, CsCTRLH: F0305H, CsCTRHL: F0306H, CsCTRHH: F0307H

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CsCTRH	ROM	BFT	–	–	–	CTMS[1:0]		CTME	ERRD	BOM[1:0]		–	TDCV FIE	SOCO IE	EOCO IE	TA IE
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CsCTRL	AL IE	BL IE	OL IE	BOR IE	BOE IE	EP IE	EW IE	BE IE	–	–	–	–	RTBO	CSL PR	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
31	ROM	Restricted Operation Mode	0: Restricted Operation Mode disabled 1: Restricted Operation Mode enabled	R/W
30	BFT	Bit Flip Test	0: First data bit of reception stream not inverted 1: First data bit of reception stream inverted	R/W
29 to 27	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
26, 25	CTMS[1:0]	Channel Test Mode Select	00B: Basic test mode 01B: Listen-Only mode 10B: Self test mode 0 (External Loop back mode) 11B: Self test mode 1 (Internal Loop back mode)	R/W
24	CTME	Channel Test Mode Enable	0: Channel Test Mode disabled 1: Channel Test Mode enabled	R/W
23	ERRD	Channel Error Display	0: Only the 1st set of error codes displayed 1: Accumulated error codes displayed	R/W
22, 21	BOM[1:0]	Channel Bus-Off Mode	00B: Normal mode (comply with ISO 11898-1) 01B: Entry to Halt Mode automatically at Bus-Off start 10B: Entry to Halt Mode automatically at Bus-Off end 11B: Entry to Halt Mode (during Bus-Off Recovery Period) by S/W	R/W
20	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
19	TDCVFIE	Transceiver Delay Compensation Violation Interrupt Enable	0: Transceiver Delay Compensation Violation Interrupt disabled 1: Transceiver Delay Compensation Violation Interrupt enabled	R/W
18	SOCOIE	Successful Occurrence Counter Overflow Interrupt Enable	0: Successful occurrence counter overflow interrupt disabled 1: Successful occurrence counter overflow interrupt enabled	R/W
17	EOCOIE	Error Occurrence Counter Overflow Interrupt Enable	0: Error occurrence counter overflow Interrupt disable 1: Error occurrence counter overflow Interrupt enabled	R/W
16	TAIE	Transmission Abort Interrupt Enable	0: TX abort Interrupt disabled 1: TX abort Interrupt enabled	R/W
15	ALIE	Arbitration Lost Interrupt Enable	0: Arbitration Lost Interrupt disabled 1: Arbitration Lost Interrupt enabled	R/W
14	BLIE	Bus Lock Interrupt Enable	0: Bus Lock Interrupt disabled 1: Bus Lock Interrupt enabled	R/W

Bit	Symbol	Bit Name	Description	R/W
13	OLIE	Overload Interrupt Enable	0: Overload Interrupt disabled 1: Overload Interrupt enabled	R/W
12	BORIE	Bus-Off Recovery Interrupt Enable	0: Bus-Off Recovery Interrupt disabled 1: Bus-Off Recovery Interrupt enabled	R/W
11	BOEIE	Bus-Off Entry Interrupt Enable	0: Bus-Off Entry Interrupt disabled 1: Bus-Off Entry Interrupt enabled	R/W
10	EPIE	Error Passive Interrupt Enable	0: Error Passive Interrupt disabled 1: Error Passive Interrupt enabled	R/W
9	EWIE	Error Warning Interrupt Enable	0: Error Warning Interrupt disabled 1: Error Warning Interrupt enabled	R/W
8	BEIE	Bus Error Interrupt Enable	0: Bus Error Interrupt disabled 1: Bus Error Interrupt enabled	R/W
7 to 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3	RTBO	Return from Bus-Off	0: Channel is not forced to return from Bus-Off 1: Channel is forced to return from Bus-Off	R/W
2	CSLPR	Channel Sleep Request	0: Channel Sleep Request disabled 1: Channel Sleep Request enabled	R/W
1,0	CHMDC[1:0]	Channel Mode Control	00B: Channel Operation Mode request 01B: Channel Reset request 10B: Channel Halt request 11B: Keep current value	R/W

The CANs Control Register is used to control the modes of the related channel. It is used to enable generation of interrupts if errors are detected on the CAN bus connected to this channel. It is also used to configure the channel in test mode.

- **ROM**

  The Restricted Operation Mode is enabled if **CsCTR.ROM** and **CsCTR.CTME** are both 1.
  This bit cannot be set in CH_SLEEP mode.
  Write to this bit only when the related RS-CANFD lite channel is in CH_HALT mode.
  This mode should only be used in Basic Test mode **CsCTR.CTMS** = 00B.
  This bit is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
  Do not set to this bit when Classical-CAN only mode.

- **BFT**

  This bit is used to check the internal CRC generator logic of the protocol controller.
  It inverts the first bit (ID bit) of the CAN message data stream being received, so that the internal generated CRC result will not match the received CRC value of the frame. See the bit stuffing rule when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.
  The internal generated CRC value is always observed in the following registers:

  - **CsERFL.CRCREG[14:0]** (Classical-CAN frames)
  - **CsFDCRC.CRCREG[20:0]** (CAN-FD frames)

  There are some limitations when using this bit:
  Other CAN node will send a reference message and the receiver node(s) can invert one bit of incoming bit stream.
  Note, the transmitter and receiver mode are sharing the same CRC generator, therefore, it is not necessary to consider the modes separately when testing this limitation.
  The Bit Flip Test Mode is enabled if **CsCTR.BFT** (new control signal which is inverting the first bit of the bit stream) and **CsCTR.CTME** are both 1 and **CsCTR.CTMS[1:0]** is 00B.
  If this function is used by a transmitting node, a bit error or an arbitration lost will occur.
  This bit cannot be set in CH_SLEEP mode.
  Do not use this function when the Self test mode 1 (Internal Loop back mode).

Write this bit only when the related RS-CANFD lite channel is in CH_HALT mode.
This bit is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.

- **CTMS[1:0]**

  The **CsCTR.CTMS** bits are used to select the required test mode.
  These bits cannot be set in CH_SLEEP or CH_RESET mode.
  Write these bits only when the related RS-CANFD lite channel is in CH_HALT mode.
  The field is cleared automatically when the related RS-CANFD lite channel moves to CH_RESET mode.

- **CTME**

  This bit is set to enable the channel test modes.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT mode.
  This bit cannot be set in CH_SLEEP mode.
  The bit is cleared automatically when the related RS-CANFD lite channel transits to CH_RESET mode.

- **ERRD**

  This bit controls the display mode of the error flag bits (bits 14 to 8) in the CANs Error Flag Register (**CsERFL**).
  If the **CsCTR.ERRD** bit is 0 and more than one error occurs at the same time, then the error flag bits will set for all the errors that occurred at the same time. No further errors are flagged until **CsERFL** is cleared.
  When the **CsCTR.ERRD** bit is 1, the error flag bits will set for all the errors regardless of occurrence order.
  This bit cannot be set in CH_SLEEP mode.
  Write to this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **BOM[1:0]**

  These bits control the timing of the recovery from Bus-Off mode of the RS-CANFD lite Channel.
  These bits cannot be set in CH_SLEEP mode.
  Write to these bits only when the related RS-CANFD lite channel is in CH_RESET mode.

- **TDCVFIE**

  An error interrupt is generated, when the **CsCTR.TDCVFIE** bit is 1 and the **CsFDSTS.TDCVF** bit belonging to the corresponding CAN channel is 1.
  This bit cannot be set in CH_SLEEP mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET mode.
  Do not set to this bit when Classical-CAN only mode.

- **SOCOIE**

  An error interrupt is generated, when the **CsCTR.SOCOIE** bit is 1 and the **CsFDSTS.SOCO** bit belonging to the corresponding CAN channel is 1.
  This bit cannot be set in CH_SLEEP mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET mode.

- **EOCOIE**

  An error interrupt is generated, when the **CsCTR.EOCOIE** bit is 1 and the **CsFDSTS.EOCO** bit belonging to the corresponding CAN channel is 1.
  This bit cannot be set in CH_SLEEP mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET mode.

- **TAIE**

  An interrupt is generated, when the **CsCTR.TAIE** bit is 1 and a transmission is successfully aborted from a TX MB belonging to the corresponding CAN channel.
  This bit cannot be set in CH_SLEEP mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET mode.

- **ALIE**

  An error interrupt is generated, when the **CsCTR.ALIE** bit and the **CsERFL.ALF** are both 1.
  This bit cannot be set in CH_SLEEP mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET mode.

- **BLIE**

  An error interrupt is generated, when the **CsCTR.BLIE** bit and the **CsERFL.BLF** are both 1.
  This bit cannot be set in CH_SLEEP mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET mode.

- **OLIE**

  An error interrupt is generated, when the **CsCTR.OLIE** bit and the **CsERFL.OVLF** are both 1.
  This bit cannot be set in CH_SLEEP mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET mode.

- **BORIE**

  An error interrupt is generated, when the **CsCTR.BORIE** bit and the **CsERFL.BORF** are both 1.
  This bit cannot be set in CH_SLEEP mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET mode.

- **BOEIE**

  An error interrupt is generated, when the **CsCTR.BOEIE** bit and the **CsERFL.BOEF** are both 1.
  This bit cannot be set in CH_SLEEP mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET mode.

- **EPIE**

  An error interrupt is generated, when the **CsCTR.EPIE** bit and the **CsERFL.EPF** are both 1.
  This bit cannot be set in CH_SLEEP mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET mode.

- **EWIE**

  An error interrupt is generated, when the **CsCTR.EWIE** bit and the **CsERFL.EWF** are both 1.
  This bit cannot be set in CH_SLEEP mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET mode.

- **BEIE**

  An error interrupt is generated, when the **CsCTR.BEIE** bit and the **CsERFL.BEF** are both 1.
  This bit cannot be set in CH_SLEEP mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET mode.

- **RTBO**

  When the protocol controller of the CAN channel enters Bus-off state, the **CsCTR.RTBO** bit can be set to 1 to force recovery from Bus-off state. The error state changes from Bus-Off state to integrating with a maximum delay of 1 CAN Bit time.
  If the **CsCTR.RTBO** bit is set to 1, the REC and TEC registers are initialized and the Bus-Off status bit (Channel Bus-off Status, **CsSTS.BOSTS**) is set to 0.
  The other registers are not initialized by this command. Even if **CsCTR.BORIE** is set, a Bus-Off recovery interrupt is not generated by this recovery from the Bus-Off state.
  This bit cannot be set in CH_SLEEP mode.
  Setting this bit in any state (other than Bus-Off) will have no effect and the bit will be cleared immediately.
  Read value is always 0.
  Return from Bus-Off command should be used only when **CsCTR.BOM[1:0]** is set to 00B.
  Write this bit only when the related RS-CANFD lite channel is in CH_OPERATION mode.
  This bit is automatically cleared once set by the System SW.

- **CSLPR**

  A sleep mode request is generated for the corresponding CAN channel, when the **CsCTR.CSLPR** bit is 1.
  A request to exit sleep mode is generated for the corresponding CAN channel, when the **CsCTR.CSLPR** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_SLEEP mode.

- **CHMDC[1:0]**

  CsCTR.CHMDC[1:0] bits can be used to configure the modes of the CAN channel.
  CAN mode transitions are described in more details in **19.6.2 Channel Modes**.
  Setting **CsCTR.CHMDC[1:0]** to 11B by the CPU has no effect.
  When the RS-CANFD lite module is in GL_HALT mode, these bits can only be set to 00B or 01B.
  These bits cannot be set in CH_SLEEP mode.
  CsCTR.CHMDC[1:0] can change automatically in case of transition to Halt mode due to **CsCTR.BOM[1:0]** setting.
  If CPU write access to **CsCTR.CHMDC[1:0]** happens at the same time when the CAN channel is about to enter Halt Mode (at the start of Bus-Off when **CsCTR.BOM[1:0]** = 01B, or at the end of Bus-Off when **CsCTR.BOM[1:0]** = 10B), then the CPU write access will have the highest priority.
  The CAN channel changes the value of **CsCTR.CHMDC[1:0]** within the Channel Control Registers for the above cases only if the **CsCTR.CHMDC[1.0]** value is 00B (Operation Mode).

19.3.3 CANs Status Register (CsSTSH, CsSTSL) [s = 0, 1]

Address: CsSTSL: F0308H, CsSTSH: F030AH
CsSTSLL: F0308H, CsSTSLH: F0309H, CsSTSHL: F030AH, CsSTSHH: F030BH

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CsSTSH	TEC[7:0]								REC[7:0]							
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CsSTSL	–	–	–	–	–	–	–	ESIF	COM STS	REC STS	TRM STS	BO STS	EP STS	CSLP STS	CHLT STS	CRST STS
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
31 to 24	TEC[7:0]	Transmission Error Count	This register increments or decrements the counter value according to error status of the CAN channel during Transmission.	R/W
23 to 16	REC[7:0]	Reception Error Count	This register increments or decrements the counter value according to error status of the CAN channel during Reception.	R
15 to 9	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
8	ESIF	Error State Indication Flag	0: No CAN-FD message has been received with the ESI flag was set 1: At least 1 CAN-FD message was received where the ESI flag was set	R/W
7	COMSTS	Channel Communication Status	0: Channel is not ready for communication 1: Channel is ready for communication	R
6	RECSTS	Channel Receive Status	0: Channel is not receiving 1: Channel is receiving	R
5	TRMSTS	Channel Transmit Status	0: Channel is not transmitting 1: Channel is transmitting	R
4	BOSTS	Channel Bus-Off Status	0: Channel not in Bus-Off state 1: Channel in Bus-Off state	R
3	EPSTS	Channel Error Passive Status	0: Channel not in Error Passive state. 1: Channel in Error Passive state.	R
2	CSLPSTS	Channel SLEEP Status	0: Channel not in Sleep Mode 1: Channel in Sleep Mode	R
1	CHLTSTS	Channel HALT Status	0: Channel not in Halt Mode 1: Channel in Halt Mode	R
0	CRSTSTS	Channel RESET Status	0: Channel not in Reset Mode 1: Channel in Reset Mode	R

The CANs Status Register shows the mode, error and TX/RX status of the related channel together with its reception and transmission error count values.

- **TEC[7:0]**
  The value of TEC error counter is displayed.
  Write these bits only when in Test Mode or RS-CANFD lite channel is in CH_HALT mode.
  These bits are cleared automatically when RS-CANFD lite module is in GL_RESET or CAN channel is in CH_RESET mode.

- **REC[7:0]**

  The value of the REC error counter is displayed.
  The value in Bus-Off state is indeterminate.
  These bits are cleared automatically when RS-CANFD lite module enters GL_RESET or CAN channel is in CH_RESET mode.

- **ESIF**

  This flag is set when the ESI bit was sampled recessive for a reception CAN message without any error. Note that in case of Loopback or Mirror Mode the self transmitted messages are considered as reception messages.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  The bit is cleared by writing 0 to it.
  This bit is cleared automatically when the related CAN channel is in Reset.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.

- **COMSTS**

  Indicates if the related CAN channel is ready for communication.
  This bit is set automatically when the related CAN channel is ready to perform communication following the detection of eleven consecutive recessive bits after leaving the Reset or Halt Mode.
  This bit is cleared automatically when the related CAN channel is in Reset or Halt Mode.

  Caution This bit is 1 during Bus-Off status.

- **RECSTS**

  Indicates if the related CAN channel is receiving a message.
  This bit is set automatically when the related CAN channel is operating as a receiver node.
  This bit is cleared automatically when the related CAN channel is in the bus-idle state or starts operating as a transmitter node.

- **TRMSTS**

  Indicates if the related CAN channel is transmitting a message.
  This bit is set automatically when the related CAN channel is operating as a transmitter node or is in the Bus-Off state.
  This bit is cleared automatically when the related CAN channel is in the bus-idle state or starts operating as a receiver node.

- **BOSTS**

  Indicates if the related CAN channel has entered the Bus-Off state.
  This bit is set automatically when the value of the related CAN Transmission Error Count bits exceeds 255 and the related CAN channel is in the Bus-Off state (**CsSTSH.TEC[7:0]** > 255).
  This bit is cleared automatically when the related CAN channel exits the Bus-Off state.

- **EPSTS**

  Indicates if the related CAN channel has entered the error passive state. This bit is automatically set upon entering the error passive state (128 ≤ **CsSTSH.TEC[7:0]** < 255, or **CsSTSH.REC[7:0]** ≥ 128).
  This bit is cleared automatically when the related CAN channel exits the error passive state or enters Reset Mode.

- **CSLPSTS**

  Indicates if the related CAN channel is in Sleep Mode.
  This bit is set automatically when the related CAN channel enters Sleep Mode.
  This bit is cleared automatically when the related CAN channel exits Sleep Mode.

- **CHLTSTS**

  Indicates if the related CAN channel is in Halt mode.
  This bit is set automatically when the related CAN channel enters Halt Mode.
  This bit is cleared automatically when the related CAN channel exits Halt Mode.

- **CRSTSTS**

  Indicates if the related CAN channel is in Reset mode.
  This bit is set automatically when the related CAN channel enters Channel Reset Mode. If the mode is changed from Reset Mode to Sleep Mode, **CsSTS.CRSTSTS** remains set to 1.
  This bit is cleared automatically when the related CAN channel exits the Channel Reset Mode, except when changing to Sleep Mode.

19.3.4 CANs Error Flag Register (CsERFLH, CsERFLL) [s = 0, 1]

Address: CsERFLL: F030CH, CsERFLH: F030EH
CsERFLLL: F030CH, CsERFLLH: F030DH, CsERFLHL: F030EH, CsERFLHH: F030FH

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CsERFLH	–	CRCREG[14:0]														
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CsERFLL	–	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
31	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
30 to 16	CRCREG[14:0]	CRC Register value	These bits show the CRC value calculated for the CAN2.0 CAN Frame.	R
15	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
14	ADERR	Acknowledge Delimiter Error	0: Channel Ack Del Error not detected 1: Channel Ack Del Error detected	R/W
13	B0ERR	Bit 0 Error	0: Channel Bit 0 Error not detected 1: Channel Bit 0 Error detected	R/W
12	B1ERR	Bit 1 Error	0: Channel Bit 1 Error not detected 1: Channel Bit 1 Error detected	R/W
11	CERR	CRC Error	0: Channel CRC Error not detected 1: Channel CRC Error detected	R/W
10	AERR	Acknowledge Error	0: Channel Ack Error not detected 1: Channel Ack Error detected	R/W
9	FERR	Form Error	0: Channel Form Error not detected 1: Channel Form Error detected	R/W
8	SERR	Stuff Error	0: Channel stuff Error not detected 1: Channel stuff Error detected	R/W
7	ALF	Arbitration Lost Flag	0: Channel Arbitration Lost not detected 1: Channel Arbitration Lost detected	R/W
6	BLF	Bus Lock Flag	0: Channel Bus Lock not detected 1: Channel Bus Lock detected	R/W
5	OVLF	Overload Flag	0: Channel Overload not detected 1: Channel Overload detected	R/W
4	BORF	Bus-Off Recovery Flag	0: Channel Bus-Off Recovery not detected 1: Channel Bus-Off Recovery detected	R/W
3	BOEF	Bus-Off Entry Flag	0: Channel Bus-Off Entry not detected 1: Channel Bus-Off Entry detected	R/W
2	EPF	Error Passive Flag	0: Channel Error Passive not detected 1: Channel Error Passive detected	R/W
1	EWF	Error Warning Flag	0: Channel Error Warning not detected 1: Channel Error Warning detected	R/W
0	BEF	Bus Error Flag	0: Channel Bus Error not detected 1: Channel Bus Error detected	R/W

The CANs Error Flag Register shows the status of various error conditions detectable regardless of the setting of the related CAN channel Error Interrupt Enable Register. It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) to check when each error condition can occur.

For this register, when only a single bit is to be cleared by the program, do not use bit clear instruction – use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.

An example in assembler language for clearing the **CsERFL.BEF** bit:

```
MOV C0ERFLLL, #0xFE
```

- **CRCREG[14:0]**

  The calculated CRC value can be read from these bits, only when **CsCTR.CTME** is 1 for the channel.
  If **CsCTR.CTME** bit is 0, then these bits are always read as 0.
  These bits show the CAN2.0 CRC value calculated by the RS-CANFD lite channel logic if the CTME bit is enabled.
  The **CsERFL.CRCREG** value is updated in the 1st bit of CRC field of the CAN Frame (reception as well as transmission).
  These bits are cleared automatically when the related channel is in CH_RESET mode.

- **ADERR**

  Indicates a detection of a Acknowledge Delimiter bit error.
  This bit can only be cleared by writing 0 to it when it is 1.
  This bit can only be set by RS-CANFD lite module logic.
  Writing 1 has no influence on the bit value.
  For clearing this bit, the following sequence should be used:

  1. Clear the corresponding flag bit.
  2. Read if the flag bit is cleared.
  3. If yes, continue, else, go back to step 1.

  This bit is set automatically when a form error is detected during the Acknowledge Delimiter state of frame transmission.
  If **CsCTR.ERRD** bit is 1, then this bit will be set if the set and clear conditions occur at the same time for this bit.
  The bit is cleared by writing 0 to it.
  It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

  Caution If CsCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at CsERFL.CRCREG[14:8] is already set and it will be set if CRCREG[14:8] is 0000000B.

- **B0ERR**

  Indicates a detection of a Acknowledge Delimiter bit error.
  This bit can only be cleared by writing 0 to it when it is 1.
  This bit can only be set by RS-CANFD lite module logic.
  Writing 1 has no influence on the bit value.
  For clearing this bit, the following sequence should be used:

  1. Clear the corresponding flag bit.
  2. Read if the flag bit is cleared.
  3. If yes, continue, else, go back to step 1.

  This bit is set automatically when a form error is detected during the Acknowledge Delimiter state of frame transmission.
  If **CsCTR.ERRD** bit is 1, then this bit will be set if the set and clear conditions occur at the same time for this bit.
  The bit is cleared by writing 0 to it.
  It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

Caution If CsCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at CsERFL.CRCREG[14:8] is already set and it will be set if CRCREG[14:8] is 0000000B.

- **B1ERR**

  Indicates a detection of a Recessive bit error.
  This bit can only be cleared by writing 0 to it when it is 1.
  This bit can only be set by RS-CANFD lite module logic.
  Writing 1 has no influence on the bit value.
  For clearing this bit, the following sequence should be used:

  1. Clear the corresponding flag bit.
  2. Read if the flag bit is cleared.
  3. If yes, continue, else, go back to step 1.

  This bit is set automatically when a Recessive bit error (expected recessive bit, sampled as dominant bit) is detected.
  If **CsCTR.ERRD** bit is 1, then this bit will be set if the set and clear conditions occur at the same time for this bit.
  The bit is cleared by writing 0 to it.
  It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

  Caution If CsCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at CsERFL.CRCREG[14:8] is already set and it will be set if CRCREG[14:8] is 0000000B.

- **CERR**

  Indicates a detection of a CAN CRC Error.
  This bit can only be cleared by writing 0 to it when it is 1.
  This bit can only be set by RS-CANFD lite module logic.
  Writing 1 has no influence on the bit value.
  For clearing this bit, the following sequence should be used:

  1. Clear the corresponding flag bit.
  2. Read if the flag bit is cleared.
  3. If yes, continue, else, go back to step 1.

  This bit is set automatically when a CRC error is detected.
  If **CsCTR.ERRD** bit is 1, then this bit will be set if the set and clear conditions occur at the same time for this bit.
  The bit is cleared by writing 0 to it.
  It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

  Caution If CsCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at CsERFL.CRCREG[14:8] is already set and it will be set if CRCREG[14:8] is 0000000B.

- **AERR**

  Indicates a detection of a CAN Acknowledge Error.
  This bit can only be cleared by writing 0 to it when it is 1.
  This bit can only be set by RS-CANFD lite module logic.
  Writing 1 has no influence on the bit value.
  For clearing this bit, the following sequence should be used:

  1. Clear the corresponding flag bit.
  2. Read if the flag bit is cleared.
  3. If yes, continue, else, go back to step 1.

  This bit is set automatically when an Acknowledge error is detected.
  If **CsCTR.ERRD** bit is 1, then this bit will be set if the set and clear conditions occur at the same time for this bit.
  The bit is cleared by writing 0 to it.
  It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

  Caution If CsCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at CsERFL.CRCREG[14:8] is already set and it will be set if CRCREG[14:8] is 0.

- **FERR**

  Indicates a detection of a CAN Form Error.
  This bit can only be cleared by writing 0 to it when it is 1.
  This bit can only be set by RS-CANFD lite module logic.
  Writing 1 has no influence on the bit value.
  For clearing this bit, the following sequence should be used:

  1. Clear the corresponding flag bit.
  2. Read if the flag bit is cleared.
  3. If yes, continue, else, go back to step 1.

  This bit is set automatically when a Form error is detected.
  If **CsCTR.ERRD** bit is 1, then this bit will be set if the set and clear conditions occur at the same time for this bit.
  The bit is cleared by writing 0 to it.
  It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

  Caution If CsCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at CsERFL.CRCREG[14:8] is already set and it will be set if CRCREG[14:8] is 0000000B.

- **SERR**

  Indicates a detection of a CAN Stuff Error.
  This bit can only be cleared by writing 0 to it when it is 1.
  This bit can only be set by RS-CANFD lite module logic.
  Writing 1 has no influence on the bit value.
  For clearing this bit, the following sequence should be used:

  1. Clear the corresponding flag bit.
  2. Read if the flag bit is cleared.
  3. If yes, continue, else, go back to step 1.

  This bit is set automatically when a Stuff error is detected.
  If **CsCTR.ERRD** bit is 1, then this bit will be set if the set and clear conditions occur at the same time for this bit.

The bit is cleared by writing 0 to it.
It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

Caution If CsCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at CsERFL.CRCREG[14:8] is already set and it will be set if CRCREG[14:8] is 0000000B.

- **ALF**

Indicates a detection of a CAN channel Bus Arbitration Lost condition.
This bit can only be cleared by writing 0 to it when it is 1.
This bit can only be set by RS-CANFD lite module logic.
Writing 1 has no influence on the bit value.
The bit is set automatically when an arbitration lost condition is detected on the CAN bus while the CAN channel is in Operation Mode.
If the set condition occurs simultaneously with the clear condition, the bit is set.
The bit is cleared by writing 0 to it.
It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

- **BLF**

Indicates a detection of a CAN channel BusLock condition.
This bit can only be cleared by writing 0 to it when it is 1.
This bit can only be set by RS-CANFD lite module logic.
Writing 1 has no influence on the bit value.
The bit is set automatically when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in Operation Mode.
If the set condition occurs simultaneously with the clear condition, the bit is set.
The bit is cleared by writing 0 to it.
It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

- **OVLF**

Indicates a detection of a CAN channel Overload State.
This bit can only be cleared by writing 0 to it when it is 1.
This bit can only be set by RS-CANFD lite module logic.
Writing 1 has no influence on the bit value.
The bit is set automatically when an overload condition is detected.
If the set condition occurs simultaneously with the clear condition, the bit is set.
The bit is cleared by writing 0 to it.
It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

- **BORF**

Indicates a detection of a CAN channel Bus-Off Recovery State.
This bit can only be cleared by writing 0 to it when it is 1.
This bit can only be set by RS-CANFD lite module logic.
Writing 1 has no influence on the bit value.
The bit is set automatically if CAN channel recovers from Bus-Off state in the following conditions:

- When **CsCTR.BOM[1:0]** is 00B, 10B, or 11B, and normal recovery (11 consecutive recessive bits × 128 times detected) occurs.

The bit is not set if CAN channel recovers from Bus-Off state in the following conditions:

- When CAN Reset Mode is requested.
- When **CsCTR.RTBO** is instructed (the CAN channel returns to error active).
- When **CsCTR.BOM[1:0]** is 01B (Entry to Halt Mode automatically at Bus-Off start).
- When **CsCTR.BOM[1:0]** is 11B and a Halt Request is asserted before the CAN channel reaches the end of the Bus-Off State.

The bit is cleared by writing 0 to it.
It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
If the set condition occurs simultaneously with the clear condition, the bit is set.
Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

- **BOEF**

Indicates a detection of a CAN channel Bus-Off State.
This bit can only be cleared by writing 0 to it when it is 1.
This bit can only be set by RS-CANFD lite module logic.
Writing 1 has no influence on the bit value.
The bit is set automatically when the CAN error state enters the Bus-Off State.
The bit is cleared by writing 0 to it.
It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
If the set condition occurs simultaneously with the clear condition, the bit is set.
Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

- **EPF**

Indicates a detection of a CAN channel Error Passive State.
This bit can only be cleared by writing 0 to it when it is 1.
This bit can only be set by RS-CANFD lite module logic.
Writing 1 has no influence on the bit value.
The bit is set automatically when the CAN error state becomes Error Passive State.
The setting of this bit only occurs when the TEC or REC initially exceed 127. Hence if the TEC or REC remain > 127 and the **CsERFL.EPF** bit is cleared by the system SW, it will not be set again until both the TEC and REC go below 128 and either TEC or REC cross over again from a value ≤ 127 to a value > 127.
If the set condition occurs simultaneously with the clear condition, the bit is set.
The bit is cleared by writing 0 to it.
It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

- **EWF**

This bit indicates if an Error Warning condition has been detected for the CAN channel.
This bit can only be cleared by writing 0 to it when it is 1.
This bit can only be set by RS-CANFD lite module logic.
Writing 1 has no influence on the bit value.
The bit is set automatically when either TEC or REC exceeds 95.
The setting of this bit only occurs when the TEC or REC initially exceed 95. Hence if the TEC or REC remain > 95 and the EWF bit is cleared by the system SW, it will not be set again until both the TEC and REC go below 96 and either TEC or REC cross over again from a value ≤ 95 to a value > 95.
If the set condition occurs simultaneously with the clear condition, the bit is set.
The bit is cleared by writing 0 to it.

It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

- **BEF**

  Indicates a detection of a CAN channel Bus error state, flagged by the bits 14 to 8 in this register.
  This bit can only be cleared by writing 0 to it when it is 1.
  This bit can only be set by RS-CANFD lite module logic.
  Writing 1 has no influence on the bit value.
  The bit is set automatically when a Bus Error is detected.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  The bit is cleared by writing 0 to it.
  It is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.

19.3.5 CANs Data Bitrate Configuration Register (CsDCFGH, CsDCFGL) [s = 0, 1]

Address: CsDCFGL: F0400H, CsDCFGH: F0402H
CsDCFGLL: F0400H, CsDCFGLH: F0401H, CsDCFGHL: F0402H, CsDCFGHH: F0403H

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CsDCFGH	–	–	–	–	DSJW[3:0]				–	–	–	–	DTSEG2[3:0]			
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CsDCFGL	–	–	–	DTSEG1[4:0]					DBRP[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
31 to 28	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
27 to 24	DSJW[3:0]	Resynchronization Jump Width	0000B: 1 Tq 0001B: 2 Tq : 1111B: 16 Tq	R/W
23 to 20	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
19 to 16	DTSEG2[3:0]	Timing Segment 2	0000B: Reserved 0001B: 2 Tq : 1110B: 15 Tq 1111B: 16 Tq	R/W
15 to 13	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
12 to 8	DTSEG1[4:0]	Timing Segment 1	00000B: Reserved 00001B: 2Tq 00010B: 3Tq 00011B: 4Tq : 11110B: 31 Tq 11111B: 32 Tq	R/W
7 to 0	DBRP[7:0]	Channel Data Baud Rate Prescaler	Data Baud Rate Prescaler division ratio	R/W

This register is used to configure the transmission / reception Data Baud Rate parameters of the channels.
The channel of Classical-CAN only mode does not have to perform the configuration of this register.

- **DSJW[3:0]**

  These bits set the synchronization jump width. A value from 1 to 16 time quanta can be set.
  These bits cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write these bits only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **DTSEG2[3:0]**

  These bits are used to set the segment TSEG2 to compensate for edges on the CAN Bus with a negative phase error.
  Any value from 2 to 16 time quanta can be written to these bits.
  These bits cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write these bits only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **DTSEG1[4:0]**

  These bits are used to set the segment TSEG1 to compensate for edges on the CAN Bus with a positive phase error. It also contains the propagation segment.
  Any value from 2 to 32 time quanta can be configured.
  These bits cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write these bits only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  See **19.15.1.2 CAN Bit Timing** for more details.

- **DBRP[7:0]**

  When these bits are set to P (0 to 255), the baud rate prescaler divides f_{CAN} by P + 1.
  The CANs Tq clock (f_{CANTQ0}) is obtained by the CAN communication clock (f_{CAN}) and setting the clock division ratio with the **DBRP[7:0]** bits and one clock cycle of the CANs Tq clock is 1 Time Quantum (Tq).
  These bits cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write these bits only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.6 CANs CAN-FD Configuration Register (CsFDCFGH, CsFDCFGL) [s = 0, 1]

Address: CsFDCFGL: F0404H, CsFDCFGH: F0406H
CsFDCFGLL: F0404H, CsFDCFGLH: F0405H, CsFDCFGHL: F0406H, CsFDCFGHH: F0407H

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CsFDCFGH	–	CLOE	REFE	FDOE	–	–	–	–	TDCO[7:0]							
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CsFDCFGL	–	–	RPNMD[1:0]		–	ESIC	TDCE	TDCOC	–	–	–	–	–	EOCCFG[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
31	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
30	CLOE	Classical-CAN only enable	0: Classical-CAN only mode disabled 1: Classical-CAN only mode enabled	R/W
29	REFE	RX edge filter enable	0: RX edge filter disabled 1: RX edge filter enabled	R/W
28	FDOE	FD only enable	0: FD only mode disabled 1: FD only mode enabled	R/W
27 to 24	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
23 to 16	TDCO[7:0]	Transceiver Delay Compensation Offset	Transceiver delay compensation offset value	R/W
15, 14	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
13, 12	RPNMD[1:0]	Return Pretended Network Filter Mode	00B: Return to Acceptance Filter Mode 01B: Return to Pretended Network Filter ID only and Acceptance Filter Mode 10B: Return to Pretended Network Filter and Acceptance Filter Mode 11B: Return to Pretended Network Filter Mode (Not return)	R/W
11	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
10	ESIC	Error State Indication Configuration	0: The ESI bit in the frame will be representing the Error state of the node itself 1: The ESI bit in the frame will be representing the Error state of message buffer if the node itself is not in error passive. If the node is in Error Passive then the ESI bit will be driven by the node itself	R/W
9	TDCE	Transceiver Delay Compensation Enable	0: Transceiver Delay Compensation disabled 1: Transceiver Delay Compensation enabled	R/W
8	TDCOC	Transceiver Delay Compensation Offset Configuration	0: Measured + offset 1: offset only	R/W
7 to 3	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
2 to 0	EOCCFG[2:0]	Error Occurrence Counter Configuration	000B: All Transmitter or Receiver CAN Frames 001B: All Transmitter CAN Frames 010B: All Receiver CAN Frames 011B: Reserved 100B: Only Transmitter or Receiver CAN-FD Data-Phase (fast bits) 101B: Only Transmitter CAN-FD Data-Phase (fast bits) 110B: Only Receiver CAN-FD Data-Phase (fast bits) 111B: Reserved	R/W

The CANs CAN-FD Configuration Register is used for settings related to CAN-FD mode.

- **CLOE**

  This bit enables the Classical-CAN only mode. If this bit is 1, then the protocol controller can only send Classical-CAN Frames and will react with a Form or CRC error on FD frames.
  Do not set **CsFDCFG.CLOE** and **CsFDCFG.FDOE** simultaneously.

CLOE	FDOE	Channel Mode
0	0	CAN-FD mode
0	1	CAN-FD only mode
1	0	Classical-CAN only mode
1	1	Setting prohibited

  This bit cannot be written in CH_OPERATION, CH_HALT and CH_SLEEP mode.
  Write this bit only when the RS-CANFD lite channel is in CH_RESET mode.

- **REFE**

  This bit enables the RX edge filter during the IDLE detection (bus integration). When it is enabled then 2 consecutive dominant Tq required to detect a synchronization edge.
  This bit cannot be written in CH_OPERATION, CH_HALT and CH_SLEEP mode.
  Do not set this bit when Classical-CAN only mode.

- **FDOE**

  This bit enables the reception and transmission of CAN-FD only frames. If enabled then communication in Classical-CAN frame format is disabled. Transmission of Classical-CAN frames is not possible, the FDF bit of the message buffer is don't care (**CFFDCSTS.CFFDF**/**TMFDCTRm.TMFDF**). If messages with Classical-CAN frame format is received then the protocol controller will treated them as invalid frames and response with error frames.
  In case a Classical-CAN frame is configured for transmitting, the FDF bit is sent as recessive, so a FD frame is sent. If the DLC is configured bigger than 8 the remaining data bytes are padded with CCH.
  Do not set **CsFDCFG.FDOE** and **CsFDCFG.CLOE** simultaneously.
  This bit cannot be written in CH_OPERATION, CH_HALT and CH_SLEEP mode.

- **TDCO[7:0]**

  These bits set the secondary sample point offset. How this value is used, depends on the **CsFDCFG.TDCOC** setting.
  If **CsFDCFG.TDCOC** = 0, the Transceiver Delay Compensation result is equal to the Trv_Delay (measured delay) + the value in **CsFDCFG.TDCO[7:0]**, rounded down to the nearest integer number of time quanta. Otherwise, the transceiver delay compensation result is equal to the value in **CsFDCFG.TDCO[7:0]**.
  Refer to **19.15.1.5 CAN Transmitter Delay Compensation** for details on how **CsFDCFG.TDCO[7:0]** is used.
  The actual offset value is interpreted as TDCO + 1. E.g if 4 is set in **TDCO[7:0]**, the offset is 5 clock cycle.
  Clock cycle is 1 cycle of CAN channel DLL clock.
  This bit cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write these bits only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  Do not set these bits when Classical-CAN only mode.

- **RPNMD[1:0]**

  These bits select the mode which returns from Pretended Network mode (**CsFDSTS.PNSTS[1:0]** = 11B).
  When the received frame passes the Pretended Network filter (ID & Payload) in Pretended Network mode and reception of a frame has completed, RS-CANFD lite changes the mode from Pretended Network mode to selected mode.
  These bits cannot be written in CH_OPERATION or CH_SLEEP mode.

- **ESIC**

  This Error State Indication Configuration, controls the sending of either the own ESI flag information or the message ESI flag information (**CFFDCSTS.CFESI** or **TMFDCTRm.TMESI**).
  This bit cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write this bit only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  Do not set this bit when Classical-CAN only mode.

- **TDCE**

  This bit enables the Transceiver Delay Compensation for the RS-CANFD lite channel.
  This bit cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write this bit only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  Do not set this bit when Classical-CAN only mode.

- **TDCOC**

  This bit selects which offset is used when defining the position of the secondary sample point (SSP) for the RS-CANFD lite channel. If the bit is set to 0 then the position of the SSP is the measured Transceiver delay plus the fixed offset. If the bit is 1 then the position of the SSP is defined only by the offset.
  This bit cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write this bit only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  Do not set this bit when Classical-CAN only mode.

- **EOCCFG[2:0]**

  These bits select which type of CAN frame configuration and direction, protocol errors are counted in.
  These bits cannot be written in CH_OPERATION or CH_SLEEP mode.
  Write these bits only when the RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.7 CANs CAN-FD Control Register (CsFDCTRH, CsFDCTRL) [s = 0, 1]

Address: CsFDCTRL: F0408H, CsFDCTRH: F040AH
CsFDCTRLL: F0408H

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CsFDCTRH	KEY[7:0]								–	–	–	–	–	–	PNMDC[1:0]	
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CsFDCTRL	–	–	–	–	–	–	–	–	–	–	–	–	–	–	SOC CLR	EOC CLR
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
31 to 24	KEY[7:0]	Key Code	These bits control the right or wrong of rewriting of PNMDC bits.	W
23 to 18	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
17, 16	PNMDC[1:0]	Pretended Network Filter Mode Control	00B: Acceptance Filter Mode request 01B: Pretended Network Filter ID only and Acceptance Filter Mode request 10B: Pretended Network Filter and Acceptance Filter Mode request 11B: Pretended Network Filter Mode request	R/W
15 to 2	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
1	SOCCLR	Successful Occurrence Counter Clear	0: No Successful Occurrence Counter clear 1: Clear Successful Occurrence Counter	R/W
0	EOCCLR	Error Occurrence Counter Clear	0: No Error Occurrence Counter clear 1: Clear Error Occurrence Counter	R/W

The CANs CAN-FD Control Register is used to control the error and successful occurrence counters.

- **KEY[7:0]**

  When C4H is written in these bits, the write of **CsFDCTR.PNMDC[1:0]** bits becomes available.
  Read value from these bits is always 00H.
  Be sure to write **CsFDCTR.PNMDC** and **CsFDCTR.KEY** simultaneously.

- **PNMDC[1:0]**

  These bits select the mode about Acceptance Filter or Pretended Network Filter.
  These bits are set the value of **CsFDCFG.RPNMD[1:0]** automatically when the received frame passes the Pretended Network Filter in Pretended Network mode and reception of a frame has completed.
  If this automatic setting by Pretended Network Filter and the set by software occur at the same time, these bits are set by the Pretended Network Filter takes precedence.
  These bits are cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
  Write the value set in **CsFDCFG.RPNMD[1:0]** or the value of Pretended Network mode (11B) to these bits.
  Be sure to write **CsFDCTR.PNMDC[1:0]** and **CsFDCTR.KEY[7:0]** simultaneously.

- **SOCCLR**

  This bit is used to clear the Successful Occurrence Counter.
  This bit cannot be set in CH_SLEEP mode or CH_RESET.

Read value is always 0.
This bit is cleared automatically by the RS-CANFD lite logic.
This bit is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.

- **EOCCLR**

  This bit is used to clear the error occurrence counter.
  This bit cannot be set in CH_SLEEP mode or CH_RESET.
  Read value is always 0.
  This bit is cleared automatically by the RS-CANFD lite logic.
  This bit is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.

19.3.8 CANs CAN-FD Status Register (CsFDSTSH, CsFDSTSL) [s = 0, 1]

Address: CsFDSTSL: F040CH, CsFDSTSH: F040EH
CsFDSTSLL: F040CH, CsFDSTSLH: F040DH, CsFDSTSHL: F040EH, CsFDSTSHH: F040FH

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CsFDSTSH	SOC[7:0]								EOC[7:0]							
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CsFDSTSL	TDCVF	–	PNSTS[1:0]		–	–	SOCO	EOCO	TDCR[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
31 to 24	SOC[7:0]	Successful occurrence counter register	These bits show the successful occurrence counter value.	R
23 to 16	EOC[7:0]	Error occurrence counter register	These bits show the error occurrence counter value.	R
15	TDCVF	Transceiver Delay Compensation Violation Flag	0: Transceiver Delay Compensation Violation has not occurred. 1: Transceiver Delay Compensation Violation has occurred.	R/W
14	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
13, 12	PNSTS[1:0]	Pretended Network Filter State	00B: Acceptance Filter Mode. 01B: Pretended Network Filter ID only and Acceptance Filter Mode. 10B: Pretended Network Filter and Acceptance Filter Mode. 11B: Pretended Network Filter Mode.	R
11, 10	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
9	SOCO	Successful occurrence counter overflow	0: Successful occurrence counter has not overflowed. 1: Successful occurrence counter has overflowed.	R/W
8	EOCO	Error occurrence counter overflow	0: Error occurrence counter has not overflowed. 1: Error occurrence counter has overflowed.	R/W
7 to 0	TDCR[7:0]	Transceiver Delay Compensation Result	Transceiver delay compensation result.	R

The CANs CAN-FD Status Register indicates the transceiver compensation delay result and its related FIFO message lost status.

- **SOC[7:0]**

  These bits are set only by RS-CANFD lite module logic.
  Writing any value has no influence on these bits.
  These bits are updated (incremented) when the occurrence of any error-free messages on the bus is detected. Any means (received or transmitted). When the counter reaches the value of FFH then updating is stopped.
  Note in case of Loopback mode the counter would be incremented twice.
  These bits are cleared by writing 1 to **CsFDCTR.SOCCLR**
  These bits are cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.

- **EOC[7:0]**

  This counter is used together with the **CsFDSTS.SOC[7:0]** counter to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing reduced payload bit length experience significantly higher error rates compared to other messages.
  This higher error rate can be detected depending on the configuration of the **CsFDCFG.EOCCFG[2:0]**.
  These bits are set only by RS-CANFD lite module logic.
  Writing any value has no influence on these bits.
  These bits are updated (incremented) when an error occurs, according to the configuration of the **CsFDCFG.EOCCFG[2:0]** bits. When the counter reaches the value of FFH then updating is stopped.
  These bits are cleared by writing 1 to **CsFDCTR.EOCCLR**.
  These bits are cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.

- **TDCVF**

  The RS-CANFD lite module is capturing internally the transmitted data bit by bit.
  This data is compared against the received CAN-Bus level which is delayed by the Transceiver loop delay.
  The Transceiver delay has some variation depending on physical parameters like temperature. The result flag **CsFDSTS.TDCR[7:0]** will be updated by every message. Hence temporary max. delay violation could be missed. This bit is capturing this violation.
  Writing 1 has no influence on the bit value.
  This bit is set automatically when the Transceiver Delay Compensation is greater than the max. delay compensation (6 data bit times − 2clk_dlc) and the internal Bit is overrun.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  This bit is cleared by writing a 0 to it.
  This bit is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.

- **PNSTS[1:0]**

  These bits indicate the current mode about Acceptance Filter or Pretended Network Filter.
  In the case of under reception, these bits reflect the value of **CsFDCTR.PNMDC[1:0]** after reception.
  In the other case, these bits reflect the value of **CsFDCTR.PNMDC[1:0]** immediately.
  These bits are cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.

- **SOCO**

  Indicates if the related CAN channel successful occurrence counter has overflowed.
  Writing 1 has no influence on the bit value.
  This bit is set automatically when **CsFDSTS.SOC[7:0]** is FFH and, successful message reception or successful message transmission occurs.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  This bit is cleared by writing a 0 to it.
  This bit is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.

- **EOCO**

  Indicates if the related CAN channel error occurrence counter has overflowed.
  Writing 1 has no influence on the bit value.
  This bit is set automatically when **CsFDSTS.EOC[7:0]** is FFH and a CAN bus error is detected based on the configuration defined in **CsFDCFG.EOCCFG[2:0].**
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  This bit is cleared by writing 0.
  This bit is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.

- **TDCR[7:0]**

  The measured delay is a multiple of the CAN channel DLL clock.
  This result depends on the **CsFDCFG.TDCOC** configuration and the offset value in **CsFDCFG.TDCO[7:0]**. Refer to **19.15.1.5 CAN Transmitter Delay Compensation** for details on how this value is derived.
  These bits are set when the transceiver delay has been measured.
  These bits are updated at the falling edge between FDF and res bit if **CsFDCFG.TDCOC** = 0 and the transceiver delay compensation is enabled (**CsFDCFG.TDCE** = 1).
  These bits are cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.

19.3.9 CANs CAN-FD CRC Register (CsFDCRCH, CsFDCRCL) [s =0, 1]

Address: CsFDCRCL: F0410H, CsFDCRCH: F0412H

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CsFDCRCH	–	–	–	–	SCNT[3:0]				–	–	–	CRCREG[20:16]				
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CsFDCRCL	CRCREG[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
31 to 28	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
27 to 24	SCNT[3:0]	Stuff bit count	These bits show the stuff bit count (mod 8) for the CAN-FD frame.	R
23 to 21	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
20 to 0	CRCREG[20:0]	CRC Register value	These bits show the CRC value calculated for the CAN-FD Frame.	R

The CANs CAN-FD CRC Register holds the CRC value calculated for the CAN-FD Frame.

- **SCNT[3:0]**

  These bits show the Stuff Count value of the CAN-FD frame. It shows the number of inserted stuff bits (modulo 8, Gray-coded) for a CAN-FD frame if the **CsCTR.CTME** bit is enabled on **CsFDCRC.SCNT[3:1]**. And the corresponding parity bit to this counter value on **CsFDCRC.SCNT[0].**

  If **CsCTR.CTME** bit is 0, then these bits are always read as 0.

  The **CsFDCRC.SCNT[3:0]** value is updated in the 1st bit of CRC field of the CAN-FD Frame (reception as well as transmission).

  These bits are cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.

- **CRCREG[20:0]**

  The calculated CRC value can be read from these bits, only when **CsCTR.CTME** is 1 for the channel.

  If **CsCTR.CTME** bit is 0, then these bits are always read as 0.

  If CRC_17 (17-bit CRC) is used, then bits 20 to 17 (**CRCREG[20:17]**) are always read as 0000B.

  These bits show the CRC value calculated by the RS-CANFD lite channel logic if the **CsCTR.CTME** bit is enabled.

  The **CsFDCRC.CRCREG[20:0]** value is updated in the 1st bit of CRC field of the CAN-FD Frame (reception as well as transmission).

  These bits are cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.

19.3.10 Global IP Version Register (GIPVHs, GIPVLs) [s = 0, 1]

Address: GIPVLs: F0310H, GIPVHs: F0312H
GIPVLsL: F0310H, GIPVLsH: F0311H, GIPVHsL: F0312H, GIPVHsH: F0313H

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GIPVHs	PSI[15:0]															
After Reset:	0	0	1	1	1	1	0	0	1	0	0	0	1	0	1	1

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GIPVLs	CPU BUS	–	–	–	–	–	IPT[1:0]		IPV[7:0]							
After Reset:	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
31 to 16	PSI[15:0]	Parameter Status Information	These bits show the status of the parameter.	R
15	CPUBUS	CPU Bus Information	This bit shows the CPU Bus type in the product.	R
14 to 10	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
9, 8	IPT[1:0]	IP Type Release Number	These bits show the IP type used in the product.	R
7 to 0	IPV[7:0]	IP Version Release Number	These bits show the IP version used in the product.	R

The Global IP Version Register show the release version of the RS-CANFD lite IP.

- **PSI[15:0]**
  These bits show the status of a parameter. Read data is 3C8BH.

- **CPUBUS**
  This bit shows the CPU Bus type. Read data is 1.

- **IPT[1:0]**
  These bits show the IP type. Read data is 01B.

- **IPV[7:0]**
  These bits show the IP version. Read data is 43H.

19.3.11 Global Configuration Register (GCFGHs, GCFGLs) [s = 0, 1]

Address: GCFGLs: F0314H, GCFGHs: F0316H
GCFGLsL: F0314H, GCFGLsH: F0315H, GCFGHsL: F0316H, GCFGHsH: F0317H

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GCFGHs	ITRCP[15:0]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GCFGLs	–	–	–	TSSS	TSP[3:0]				–	–	CMPOC	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
31 to 16	ITRCP[15:0]	Interval Timer Reference Clock Prescaler	FIFO Interval timer prescaler value	R/W
15 to 13	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
12	TSSS	Timestamp Source Select	0: Source clock for Timestamp counter is peripheral clock 1: Source clock for Timestamp counter is bit time clock	R/W
11 to 8	TSP[3:0]	Timestamp Prescaler	0000B: Timestamp Prescaler = 1 0001B: Timestamp Prescaler = 2 0010B: Timestamp Prescaler = 4 0011B: Timestamp Prescaler = 8 : 1101B: Timestamp Prescaler = 8192 1110B: Timestamp Prescaler = 16384 1111B: Timestamp Prescaler = 32768	R/W
7, 6	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
5	CMPOC	CAN-FD Message Payload Overflow Configuration	0: Message is rejected 1: Message payload is cut to fit to configured message size	R/W
4	DCS	Data Link Controller Clock Select	0: Internal clock 1: External Clock source (X1 clock direct).	R/W
3	MME	Mirror Mode Enable	0: Mirror Mode disabled 1: Mirror Mode enabled	R/W
2	DRE	DLC Replacement Enable	0: DLC replacement disabled 1: DLC replacement enabled	R/W
1	DCE	DLC Check Enable	0: DLC check disabled 1: DLC check enabled	R/W
0	TPRI	Transmission Priority	0: ID Priority 1: Message Buffer Number Priority	R/W

The Global Configuration Register is used to select the transmission priority to be used for all the TX Message Buffers and the clock source for the CAN protocol engine of all CAN channels. It is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

- **ITRCP[15:0]**
  These bits allow the definition of a reference clock for the FIFO interval timer source clock.
  When the **GCFG.ITRCP[15:0]** bits are 0000H, then the timer is disabled.
  These bits cannot be written in GL_SLEEP mode.
  Write these bits only when RS-CANFD lite module is in GL_RESET mode.

- **TSSS**

  This bit allows the selection of the clock source for the Timestamp counter.
  This bit cannot be written in GL_SLEEP mode.
  Write this bit only when RS-CANFD lite module is in GL_RESET mode.
  Do not write 1 in this bit when CAN-FD communication will be used.
  Note bit time clock could be variable depending on the nominal and data rate bit configuration.

- **TSP[3:0]**

  The value configured in these bits defines the period of the clock source used for the Timestamp counter.
  These bits cannot be written in GL_SLEEP mode.
  Write these bits only when RS-CANFD lite module is in GL_RESET mode.

- **CMPOC**

  The received message payload is always compared with the available message payload size in the Message Buffer. This bit controls the message payload acceptance mechanism in the case when the received payload is higher than the Message Buffer payload size **RMNB.RMPLS[2:0], RFCCk.RFPLS[2:0], CFCC.CFPLS[2:0]**.
  This bit cannot be written in GL_SLEEP or GL_OPERATION mode.
  Write this bit only when RS-CANFD lite module is in GL_RESET mode.
  When this bit is set and payload overflow occurs, DLC value is stored in a RXMB or FIFO without changing.

- **DCS**

  This bit selects the clock source for the CAN communications. Internal clean clock has a smaller clock jitter than the Peripheral clock (pclk).
  This bit cannot be written in GL_SLEEP or GL_OPERATION mode.
  Write this bit only when RS-CANFD lite module is in GL_RESET mode.
  Do not select f_{IH} (High-speed on-chip oscillator clock) for source of Data Link Controller Clock.

- **MME**

  This bit enables the Mirror Mode for all CAN channels.
  This bit cannot be written in GL_SLEEP mode.
  Write this bit only when RS-CANFD lite module is in GL_RESET mode.

- **DRE**

  If this bit is 1 and **GCFG.DCE** is 1, then RS-CANFD lite will store the configured value (**GAFLP0i.GAFLDLC[3:0]**) of DLC in the destination RX Message Buffer or FIFO buffer if the DLC check passes. Otherwise the DLC value in the destination RX Message Buffer or FIFO buffer is unchanged.
  This bit cannot be written in GL_SLEEP mode.
  Write this bit only when RS-CANFD lite module is in GL_RESET mode.

- **DCE**

  This bit enables the DLC check for all CAN channels.
  This bit cannot be written in GL_SLEEP mode.
  Write this bit only when RS-CANFD lite module is in GL_RESET mode.

- **TPRI**

  This bit selects the transmission priority for all CAN channels.
  This bit cannot be written in GL_SLEEP mode.
  Write this bit only when RS-CANFD lite module is in GL_RESET mode.

19.3.12 Global Control Register (GCTRHs, GCTRLs) [s = 0, 1]

Address: GCTRLs: F0318H, GCTRHs: F031AH
GCTRLsL: F0318H, GCTRLsH: F0319H, GCTRHsL: F031AH

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GCTRHs	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	TSRST
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GCTRLs	–	–	–	–	CMPOFIE	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
31 to 17	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
16	TSRST	Timestamp Reset	0: Timestamp not reset 1: Timestamp reset	R/W
15 to 12	–	Reserved	There bits are read as 0. The write value should be always 0.	R/W
11	CMPOFIE	CAN-FD message payload overflow Flag Interrupt enable	0: CAN-FD message payload overflow Flag Interrupt Disabled 1: CAN-FD message payload overflow Flag Interrupt Enabled	R/W
10	THLEIE	TX History List Entry Lost Interrupt Enable	0: TX History List Entry Lost Interrupt Disabled 1: TX History List Entry Lost Interrupt Enabled	R/W
9	MEIE	Message lost Error Interrupt Enable	0: Message Lost Error Interrupt Disabled 1: Message Lost Error Interrupt Enabled	R/W
8	DEIE	DLC check Interrupt Enable	0: DLC check Interrupt Disabled 1: DLC check Interrupt Enabled	R/W
7 to 3	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
2	GSLPR	Global Sleep Request	0: Global Sleep Request Disabled 1: Global Sleep Request Enabled	R/W
1, 0	GMDC[1:0]	Global Mode Control	00B: Global Operation Mode Request 01B: Global Reset Mode Request 10B: Global Halt Mode Request 11B: Keep Current Value	R/W

The Global Control Register is used to control the global mode of the RS-CANFD lite module and to control the timestamp function. It is also used to enable and disable the global error interrupts.

- **TSRST**

  When this bit is 1, the Global Timestamp Register is reset to 0000H.
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.
  Do not write this bit when the RS-CANFD lite module is in GL_RESET mode.
  Read value is always 0.
  This bit is cleared automatically by the RS-CANFD lite logic.

- **CMPOFIE**

  If this bit is 1, then an Interrupt will be generated when a CAN-FD message payload overflow condition occurs.
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.

- **THLEIE**

  If this bit is 1, then an Interrupt will be generated when a TX History List Entry Lost condition occurs.
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.

- **MEIE**

  If this bit is 1, then an Interrupt will be generated when a Message Lost condition occurs.
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.

- **DEIE**

  If this bit is 1, then an interrupt will be generated when a DLC error is detected in received frames.
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.

- **GSLPR**

  This bit globally selects the Sleep request for RS-CANFD lite module including CAN channel (Channel Sleep request is set automatically for channel).
  Write this bit only when RS-CANFD lite module is in GL_RESET or GL_SLEEP mode.

- **GMDC[1:0]**

  These bits can be used to configure the modes for the RS-CANFD lite module.
  Additionally, if **GCTR.GSLPR** is 1 when the RS-CANFD lite module is in Reset Mode, then the RS-CANFD lite module transits to Global Sleep Mode.
  Setting **GCTR.GMDC[1:0]** to 11B has no effect. Mode transition is described in detail later in **19.6.1 Global Modes**.
  These bits cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.

19.3.13 Global Status Register (GSTSs) [s = 0, 1]

Address: GSTSs: F031CH
GSTSsL: F031CH

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GSTSs	–	–	–	–	–	–	–	–	–	–	–	–	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 read as 0. The write value should be always 0.	R/W
3	GRAMINIT	Global RAM Initialization	0: RAM initialization is finished 1: RAM initialization ongoing	R
2	GSLPSTS	Global Sleep Status	0: Not in Sleep Mode 1: In Sleep Mode	R
1	GHLTSTS	Global Halt Status	0: Not in Halt Mode 1: In Halt Mode	R
0	GRSTSTS	Global Reset Status	0: Not in Reset Mode 1: In Reset Mode	R

The Global Status Register indicates the global status of the RS-CANFD lite module.

- **GRAMINIT**

  Indicates the Global RS-CANFD lite Module RAM initialization state.
  This bit is set automatically when the RS-CANFD lite module enters Global Sleep Mode after CAN Reset.
  This bit is cleared automatically when the RS-CANFD lite module has finished the RAM initialization.

- **GSLPSTS**

  Indicates the Global RS-CANFD lite Module Sleep mode.
  This bit is set automatically when the RS-CANFD lite module enters Global Sleep Mode.
  This bit is cleared automatically when the RS-CANFD lite module exits the Sleep Mode.

- **GHLTSTS**

  Indicates the Global RS-CANFD lite Module Halt mode.
  This bit is set automatically when the RS-CANFD lite module enters Global Halt Mode.
  This bit is cleared automatically when the RS-CANFD lite module exits the Halt Mode.

- **GRSTSTS**

  Indicates the Global RS-CANFD lite Module Reset mode.
  This bit is set automatically when the RS-CANFD lite module enters Global Reset Mode. When the mode is changed from Global Reset Mode to Global Sleep Mode, **GSTS.GRSTSTS** remains set.
  This bit is cleared automatically when the RS-CANFD lite module exits the Global Reset Mode.

19.3.14 Global Error Flag Register (GERFLHs, GERFLLs) [s = 0, 1]

Address: GERFLLs: F0320H, GERFLHs: F0322H
GERFLLsL: F0320H, GERFLHsL: F0322H

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GERFLHs	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	EEF
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GERFLLs	–	–	–	–	–	–	–	–	–	–	–	–	CMPOF	THLES	MES	DEF
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
31 to 17	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
16	EEF	ECC Error Flag	0: ECC Error not detected during TX-SCAN 1: ECC Error detected during TX-SCAN	R/W
15 to 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3	CMPOF	CAN-FD message payload overflow Flag	0: CAN-FD message payload overflow not detected 1: CAN-FD message payload overflow detected	R/W
2	THLES	TX History List Entry Lost Error Status	0: TX History List Entry Lost Error not detected 1: TX History List Entry Lost Error detected	R
1	MES	Message Lost Error Status	0: Message lost Error not detected 1: Message lost Error detected	R
0	DEF	DLC Error Flag	0: DLC Error not detected 1: DLC Error detected	R/W

The Global Error Flag Register indicates the detection of global errors.

- **EEF**

  This flag indicates that CAN RAM ECC Error was detected at two bits of the data read from the CAN RAM during CAN RAM read access while error detection is enabled.
  To be enabled this flag is referring **29.3.4 CAN RAM-ECC Function (RL78/F25 Only)**.
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP or GL_RESET mode.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.
  Writing 1 has no influence on the bit values.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  The bit is cleared by writing 0 to it.
  This bit will be cleared automatically in GL_RESET mode.

- **CMPOF**

  If this bit is set then a payload overflow happened on at least one channel.
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP or GL_RESET mode.
  Writing 1 has no influence on the bit values.
  This bit is set automatically when a CAN-FD message payload overflow is detected on at least one channel.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  The bit is cleared by writing 0 to it.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.
This bit will be cleared automatically in GL_RESET mode.

- **THLES**

  This bit is set automatically when a TX History List Entry Lost Error is detected.
  This bit is cleared automatically when all TX History List Entry Lost flags are cleared.
  This bit will be cleared automatically in GL_RESET mode.

- **MES**

  This bit is set automatically when a FIFO Message Lost Error is detected.
  This bit is cleared automatically when all FIFO Message Lost flags are cleared.
  This bit will be cleared automatically in GL_RESET mode.

- **DEF**

  This bit is set automatically when a DLC Error is detected in a received frame.
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP or GL_RESET mode.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.
  Writing 1 has no influence on the bit values.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  The bit is cleared by writing 0 to it.
  This bit will be cleared automatically in GL_RESET mode.

19.3.15 Global TX Interrupt Status Register (GTINTSTSs) [s = 0, 1]

Address: GTINTSTSs: F03A4H
GTINTSTSsL: F03A4H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GTINTSTSs	–	–	–	–	–	–	–	–	–	–	–	THIF	CFTIF	–	TAIF	TSIF
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	There bits are read as 0. The write value should be always 0.	R/W
4	THIF	TX History List Interrupt Flag	0: TX History List Interrupt flag not set 1: TX History List Interrupt flag set	R
3	CFTIF	COM FIFO TX Mode Interrupt Flag	0: COM FIFO TX mode Interrupt flag not set 1: COM FIFO TX mode Interrupt flag set	R
2	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
1	TAIF	TX Abort Interrupt Flag	0: TX abort Interrupt flag not set 1: TX abort Interrupt flag set	R
0	TSIF	TX Successful Interrupt Flag	0: TX Successful completion Interrupt flag not set 1: TX Successful completion Interrupt flag set	R

- **THIF**

  This bit is set automatically when the related TX History List Interrupt Flag (**THLSTS.THLIF**) is set when the Interrupt is enabled.
  This bit is cleared automatically when related TX History List Interrupt Flag is cleared or the Interrupt enable is disabled.
  This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

- **CFTIF**

  This bit is set automatically when the related Common TX FIFO Interrupt Flag (**CFSTS.CFTXIF**) is set when the Interrupt is enabled.
  This bit is cleared automatically when related Common TX FIFO Interrupt Flag is cleared or the Interrupt enable is disabled.
  This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

- **TAIF**

  This bit is set automatically when abort Successful flag of the related channel is set when the Interrupt is enabled.
  This bit is cleared automatically when the related TX MB Result status bits are cleared or the Interrupt enable is disabled.
  This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

- **TSIF**

  This bit is set automatically when the Transmission Successful flag of the related channel is set when the Interrupt is enabled.
  This bit is cleared automatically when related TX MB Result status bits are cleared or the Interrupt enable is disabled.
  This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

19.3.16 Global Timestamp Counter Register (GTSCs) [s = 0, 1]

Address: GTSCs: F0324H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GTSCs	TS[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	TS[15:0]	Timestamp Value	Timestamp Value	R

The Global Timestamp Counter Register stores the timestamp based on the selected configuration.

- **TS[15:0]**

  The Timestamp value is stored in this register based on the configuration of **GCFG.TSSS** and **GCFG.TSP[3:0]** bits.
  The proper incrementing of the Time Stamp counter cannot be guaranteed when moving to Halt state.
  These bits cannot be written when the RS-CANFD lite module is in GL_RESET or GL_SLEEP mode.
  These bits will be cleared automatically in GL_RESET mode.

19.3.17 Global Acceptance Filter List Entry Control Register (GAFLECTRs) [s = 0, 1]

Address: GAFLECTRs: F0328H
GAFLECTRsH: F0329H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GAFLECTRs	–	–	–	–	–	–	–	AFLDAE	–	–	–	–	–	–	–	–
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 9	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
8	AFLDAE	Acceptance Filter List Data Access Enable	0: Acceptance Filter List Data access disabled 1: Acceptance Filter List Data access enabled	R/W
7 to 0	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W

This register selects the Acceptance Filter List write access.

- **AFLDAE**

  This bit prevents Acceptance Filter List write access if cleared after configuration of the Acceptance Filter List.
  The acceptance filter list can be read regardless of the setting of this bit.
  This bit cannot be written when RS-CANFD lite module is in GL_SLEEP mode.
  This bit should be set to enable write access to Acceptance Filter List.

19.3.18 Global Acceptance Filter List Configuration Register (GAFLCFGs) [s = 0, 1]

Address: GAFLCFGs: F032EH
GAFLCFGsL: F032EH

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GAFLCFGs	–	–	–	–	–	–	–	–	–	–	–	RNC[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 read as 0. The write value should be always 0.	R/W
4 to 0	RNC[4:0]	Rule Number	Number of rules dedicated	R/W

- **RNC[4:0]**

  These bits define the number of rules in the Acceptance Filter List.
  Write these bits only when RS-CANFD lite module is in GL_RESET mode.
  Set these bits to a value within a range of 00000B to 10000B.

19.3.19 Global Acceptance Filter List ID Register i (GAFLIDiHs, GAFLIDiLs) [i = 0 to 15] [s = 0, 1]

Address: GAFLIDiLs: F0420H + (10H×i), GAFLIDiHs: F0422H + (10H×i)
GAFLIDiLsL: F0420H + (10H×i), GAFLIDiLsH: F0421H + (10H×i),
GAFLIDiHsL: F0422H + (10H×i), GAFLIDiHsH: F0423H + (10H×i)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GAFLIDiHs	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GAFLIDiLs	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
31	GAFLIDE	Global Acceptance Filter List Entry IDE Field	0: Standard Identifier of Rule entry ID is valid for acceptance filtering. 1: Extended Identifier of Rule entry ID is valid for acceptance filtering.	R/W
30	GAFLRTR	Global Acceptance Filter List Entry RTR Field	0: Data Frame 1: Remote Frame	R/W
29	GAFLLB	Global Acceptance Filter List Entry Loopback Configuration	0: Global Acceptance Filter List entry ID for acceptance filtering has attribute ‘RX’. 1: Global Acceptance Filter List entry ID for acceptance filtering has attribute ‘TX’.	R/W
28 to 0	GAFLID[28:0]	Global Acceptance Filter List Entry ID Field	ID part of the Global Acceptance Filter List entry.	R/W

These registers are used to configure the ID field of the rule entries in the Global Acceptance Filter List.

- **GAFLIDE**

  This bit allows the configuration of the ID format (standard ID or extended ID) for each of the Global Acceptance Filter List entry. For each Rule entry of the related CAN channel the Acceptance filter process compares this bit against the IDE bit of the received CAN message.
  This bit cannot be written when **GAFLECTR.AFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GAFLRTR**

  This bit allows the configuration of the specified frame format (data frame or remote frame) for each Global Acceptance Filter List entry. For each Rule entry in a CAN channel the Acceptance filter process compares this bit against the RTR bit of the received CAN message.
  This bit cannot be written when **GAFLECTR.AFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GAFLLB**

  This bit selects if the Global Acceptance Filter List entry gets the attribute RX or TX. This attribute decides about the validity of the entry in the mirror mode case, loopback test mode case and during standard (non-loopback) reception.

See **Table 19-16** for detailed description of the validity of the Global Acceptance Filter List entry depending on transmitter/receiver case, type of loopback mode and RX/TX attribute.
This bit cannot be written when **GAFLECTR.AFLDAE** bit is 0.
Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GAFLID[28:0]**

  These bits represent the CAN Identifier (ID) field of each of the Global Acceptance Filter List entry. Acceptance filter process compares this field against the ID of a received CAN message.
  For alignment of these bits in standard and extended frame format, see **19.4 Identifier Bits Alignment**.
  These bits cannot be written when **GAFLECTR.AFLDAE** bit is 0.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.20 Global Acceptance Filter List Mask Register i (GAFLMiHs, GAFLMiLs) [i = 0 to 15] [s = 0, 1]

Address: GAFLMiLs: F0424H + (10H×i), GAFLMiHs: F0426H + (10H×i)
GAFLMiLsL: F0424H + (10H×i), GAFLMiLsH: F0425H + (10H×i),
GAFLMiHsL: F0426H + (10H×i), GAFLMiHsH: F0427H + (10H×i)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GAFLMiHs	GAFLIDEM	GAFLRTRM	GAFLIFL1	GAFLIDM[28:16]												
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GAFLMiLs	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
31	GAFLIDEM	Global Acceptance Filter List IDE Mask	0: IDE bit is not considered for ID matching. 1: IDE bit is considered for ID matching.	R/W
30	GAFLRTRM	Global Acceptance Filter List Entry RTR Mask	0: RTR bit is not considered for ID matching. 1: RTR bit is considered for ID matching.	R/W
29	GAFLIFL1	Global Acceptance Filter List Information Label 1	Global Acceptance Filter List Information Label bit1.	R/W
28 to 0	GAFLIDM[28:0]	Global Acceptance Filter List ID Mask Field	0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.	R/W

These registers are used to configure the mask field of each rule entries in the Global Acceptance Filter List.

- **GAFLIDEM**

  This bit allows the configuration of the IDE mask bit for each Global Acceptance Filter List entry.
  When IDE mask bit is 0, then the ID comparison depends upon the received IDE bit.
  - If received IDE bit is 0, then STD-ID comparison takes place.
  - If received IDE bit is 1, then EXT-ID comparison takes place.

  This bit cannot be written when **GAFLECTR.AFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GAFLRTRM**

  This bit allows the configuration of the RTR mask bit for each Global Acceptance Filter List entry.
  This bit cannot be written when **GAFLECTR.AFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GAFLIFL1**

  This bit allow the configuration of a 2-bit Information label that will be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a MSB bit of an information label.
  This bit cannot be written when **GAFLECTR.AFLDAE** bit is 0.
  Write this bit when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  This bit is stored in Information Label Field [1] (**RMSTSn.RMIFL[1]**, **RFFDSTSk.RFIFL[1]**, and **CFFDCSTS.CFIFL[1]**) of the storage location of an incoming message.

- **GAFLIDM[28:0]**

  These bits are the filter mask bits for the related bits in the CAN Identifier field of each Global Acceptance Filter List entry.

  These bits cannot be written when **GAFLECTR.AFLDAE** bit is 0.

  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.21 Global Acceptance Filter List Pointer 0 Register i (GAFLP0iHs, GAFLP0iLs) [i = 0 to 15] [s = 0, 1]

Address: GAFLP0iLs: F0428H + (10H×i), GAFLP0iHs: F042AH + (10H×i)
GAFLP0iLsL: F0428H + (10H×i), GAFLP0iLsH: F0429H + (10H×i),
GAFLP0iHsL: F042AH + (10H×i), GAFLP0iHsH: F042BH + (10H×i)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GAFLP0iHs	GAFLPTR[15:0]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GAFLP0iLs	GAFLRMV	–	–	GAFLRMDP[4:0]					GAFLIFL0	–	–	–	GAFLDLC[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
31 to 16	GAFLPTR[15:0]	Global Acceptance Filter List Pointer Field	Global Acceptance Filter List Pointer	R/W
15	GAFLRMV	Global Acceptance Filter List RX Message Buffer Valid	0: Global Acceptance Filter List Single Message Buffer Direction Pointer is invalid. 1: Global Acceptance Filter List Single Message Buffer Direction Pointer is valid.	R/W
14, 13	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
12 to 8	GAFLRMDP[4:0]	Global Acceptance Filter List RX Message Buffer Direction Pointer	RX Message Buffer number for storage of received messages.	R/W
7	GAFLIFL0	Global Acceptance Filter List Information Label 0	Global Acceptance Filter List Information Label bit0.	R/W
6 to 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3 to 0	GAFLDLC[3:0]	Global Acceptance Filter List DLC Field	Minimum no. of Data Bytes in a Data Frame required for its acceptance.	R/W

These registers are used to configure the DLC, SW pointer, single message buffer select and message buffer direction pointer for each rule entry in the Global Acceptance Filter List.

- **GAFLPTR[15:0]**

  These bits allow the configuration of a 16-bit pointer that will be attached to a received message accepted by the related Global Acceptance Filter List entry. The Pointer will be added during message storage in the Message Buffer area and can be used by the application as support function. The pointer information could be used for example to support PDU Identifier allocation for the received message in AUTOSAR systems.
  These bits cannot be written when **GAFLECTR.AFLDAE** bit is 0.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GAFLRMV**

  This bit allows the enabling/disabling of a single reception Message Buffer as the target for a received message that is passing the acceptance check of the related Global Acceptance Filter List entry.
  This bit cannot be written when **GAFLECTR.AFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GAFLRMDP[4:0]**

  These bits allow the configuration of a single reception Message Buffer as the destination target for a received message that is passing the acceptance check of the related Global Acceptance Filter List entry. The value entered is the single destination Message Buffer number.
  These bits cannot be written when **GAFLECTR.AFLDAE** bit is 0.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  RMNB.NRXMB[4:0] is the value entered in the RX Message Buffer Number Register to configure the number of RX Message Buffers. The value to be entered in **GAFLP0i.GAFLRMDP[4:0]** bits should only be between 00000B and "(**RMNB.NRXMB[4:0]** value) − 1".
  If **RMNB.NRXMB[4:0]** = 00000B, then the **GAFLP0i.GAFLRMV** bit should be configured as 0.

- **GAFLIFL0**

  This bit allow the configuration of a 2-bit Information label that will be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a LSB bit of an information label.
  This bit cannot be written when **GAFLECTRs.AFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  This bit is stored in Information Label Field [0] (**RMSTSn.RMIFL[0]**, **RFFDSTSk.RFIFL[0]**, and **CFFDCSTS.CFIFL[0]**) of the storage location of an incoming message.

- **GAFLDLC[3:0]**

  These bits allow the configuration of the minimum DLC (Data Length Code) value for a message to be accepted by the related Global Acceptance Filter List entry (automatic DLC filter function). DLC filter process is only passed if the DLC value of the message accepted by a Global Acceptance Filter List entry is equal or higher than the DLC value configured for this related Global Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding Rule Entry when this field is set to 0000B.

  Following binary values can be configured:

Format	GAFLDLC[3]	GAFLDLC[2]	GAFLDLC[1]	GAFLDLC[0]	Description
CAN and CAN-FD	0	0	0	0	DLC of received message = 0 or more (DLC Filter check is disabled)
CAN and CAN-FD	0	0	0	1	DLC of received message = 1 or more
CAN and CAN-FD	0	0	1	0	DLC of received message = 2 or more
CAN and CAN-FD	0	0	1	1	DLC of received message = 3 or more
CAN and CAN-FD	0	1	0	0	DLC of received message = 4 or more
CAN and CAN-FD	0	1	0	1	DLC of received message = 5 or more
CAN and CAN-FD	0	1	1	0	DLC of received message = 6 or more
CAN and CAN-FD	0	1	1	1	DLC of received message = 7 or more
CAN	1	x	x	x	DLC of received message = 8 or more
CAN-FD	1	0	0	0	DLC of received message = 8 or more
CAN-FD	1	0	0	1	DLC of received message = 12 or more
CAN-FD	1	0	1	0	DLC of received message = 16 or more
CAN-FD	1	0	1	1	DLC of received message = 20 or more
CAN-FD	1	1	0	0	DLC of received message = 24 or more
CAN-FD	1	1	0	1	DLC of received message = 32 or more
CAN-FD	1	1	1	0	DLC of received message = 48 or more
CAN-FD	1	1	1	1	DLC of received message = 64

  These bits cannot be written when **GAFLECTR.AFLDAE** bit is 0.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.22 Global Acceptance Filter List Pointer 1 Register i (GAFLP1iLs) [i = 0 to 15] [s = 0, 1]

Address: GAFLP1iLs: F042CH + (10H × i)
GAFLP1iLsL: F042CH + (10H × i), GAFLP1iLsH: F042DH + (10H × i),
Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GAFLP1iLs	–	–	–	–	–	–	–	GAFLFDP[8]	–	–	–	–	–	–	GAFLFDP[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 9	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
8	GAFLFDP[8]	Global Acceptance Filter List FIFO Direction Pointer (GAFLFDP[8])	FIFO direction pointer bits for received message storage	R/W
7 to 2	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
1, 0	GAFLFDP[1:0]	Global Acceptance Filter List FIFO Direction Pointer (GAFLFDP[1:0])	FIFO direction pointer bits for received message storage	R/W

These registers are used to configure the FIFO direction pointer fields in each rule entry of the Global Acceptance Filter List.

- **GAFLFDP[8], [1:0]**

  These bits allow the configuration of FIFO Buffers as the target for a received message passing the acceptance check of the related Global Acceptance Filter List entry. Each bit of the **GAFLP1i.GAFLFDP[8]**, **[1:0]** is configuring a dedicated FIFO:

Bit	Symbol	Value (Binary)	Function
8	GAFLFDP[8]	0	Disable Common FIFO as target for reception
		1	Enable Common FIFO as target for reception
1	GAFLFDP[1]	0	Disable RX FIFO 1 as target for reception
		1	Enable RX FIFO 1 as target for reception
0	GAFLFDP[0]	0	Disable RX FIFO 0 as target for reception
		1	Enable RX FIFO 0 as target for reception

  These bits cannot be written when **GAFLECTR.AFLDAE** bit is 0.
  For storage in Common FIFO, target for reception can only be those Common FIFO Buffers that are configured as RX FIFO.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  These bits should be up to 2 destination FIFO Buffers or 1 destination FIFO Buffers plus one RX Message Buffer.

19.3.23 RX Message Buffer Number Register (RMNBs) [s = 0, 1]

Address: RMNBs: F0330H
RMNBsL: F0330H, RMNBsH: F0331H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RMNBs	–	–	–	–	–	RMPLS[2:0]			–	–	–	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 11	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
10 to 8	RMPLS[2:0]	Reception Message Buffer Payload Data Size	000B: 8 Bytes 001B: 12 Bytes 010B: 16 Bytes 011B: 20 Bytes 100B: 24 Bytes 101B: 32 Bytes 110B: 48 Bytes 111B: 64 Bytes	R/W
7 to 5	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
4 to 0	NRXMB[4:0]	Number of RX Message Buffers	Used to define the number of RX Message Buffer	R/W

The RX Message Buffer Number Register is used to configure the total number of RX Message Buffers allocated to channel.

- **RMPLS[2:0]**
  These bits are used to configure the message buffer payload data size.
  Write these bits only when RS-CANFD lite module is in GL_RESET mode.

- **NRXMB[4:0]**
  These bits are used to configure the number of RX Message Buffers.
  Write these bits only when RS-CANFD lite module is in GL_RESET mode.
  Set these bits to a value within a range of 00000B to 10000B. Setting these bits to 00000B makes RX Message Buffers unavailable.

19.3.24 RX Message Buffer New Data Register (RMNDs) [s = 0, 1]

Address: RMNDs: F0334H
RMNDsL: F0334H, RMNDsH: F0335H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RMNDs	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]	RX Message Buffer New Data Status	0: New Data not stored in corresponding RX Message Buffer 1: New Data stored in corresponding RX Message Buffer	R/W

The RX Message Buffer New Data Register bits show the new data storage status of the RX Message Buffers.
The bit position of **RMND** register corresponds to the buffer number of RXMB.

- **RMNS[15:0]**

  These bits show the NewData Flag status for the corresponding RX Message Buffer. **RMNS** bit 0 corresponds to RX Message Buffer 0 and so on.
  These bits cannot be written when RS-CANFD lite module is in GL_RESET or GL_SLEEP mode.
  Writing 1 has no influence on the bit values.

  These bits cannot be cleared when message storage in the corresponding RX Message Buffer is in progress.
  Do not use bit clear instruction for clearing these bits. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.
  The field is set automatically when storage of a new message starts in the corresponding RX Message Buffer.
  The duration of message storage time is 6 peripheral clock (pclk) cycles for **RMNB.RMPLS[2:0]** = 000B (max 8 bytes payload).
  For **RMNB.RMPLS[2:0]** > 000B, it is 6 + 1 for each four bytes (max 20 peripheral clock (pclk) cycles for 64 bytes: 6 + (64-8)/4 = 20).
  These bits are cleared by writing 0 to it.
  These bits are cleared automatically when RS-CANFD lite module enters GL_RESET mode.

19.3.25 RX FIFO Configuration / Control Register k (RFCCks) [k = 0, 1] [s = 0, 1]

Address: RFCCks: F033CH + (4H×k)
RFCCksL: F033CH + (4H×k), RFCCksH: F033DH + (4H×k)

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RFCCks	RFIGCV[2:0]			RFIM	–	RFDC[2:0]			–	RFPLS[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]	RX FIFO Interrupt Generation Counter Value	000B: Interrupt generated when FIFO is 1/8th Full 001B: Interrupt generated when FIFO is 1/4th Full 010B: Interrupt generated when FIFO is 3/8th Full 011B: Interrupt generated when FIFO is 1/2 Full 100B: Interrupt generated when FIFO is 5/8th Full 101B: Interrupt generated when FIFO is 3/4th Full 110B: Interrupt generated when FIFO is 7/8th Full 111B: Interrupt generated when FIFO is Full	R/W
12	RFIM	RX FIFO Interrupt Mode	0: Interrupt generated when RX FIFO counter reaches RFIGCV value from values smaller than RFIGCV 1: Interrupt generated at the end of every received message storage	R/W
11	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
10 to 8	RFDC[2:0]	RX FIFO Depth Configuration	000B: FIFO Depth = 0 Messages 001B: FIFO Depth = 4 Messages 010B: FIFO Depth = 8 Messages 011B: FIFO Depth = 16 Messages 100B: Reserved 101B: Reserved 110B: Reserved 111B: Reserved	R/W
7	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
6 to 4	RFPLS[2:0]	RX FIFO Payload Data Size Configuration	000B: 8 Bytes 001B: 12 Bytes 010B: 16 Bytes 011B: 20 Bytes 100B: 24 Bytes 101B: 32 Bytes 110B: 48 Bytes 111B: 64 Bytes	R/W
3, 2	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
1	RFIE	RX FIFO Interrupt Enable	0: FIFO Interrupt generation disabled 1: FIFO Interrupt generation enabled	R/W
0	RFE	RX FIFO Enable	0: FIFO disabled 1: FIFO enabled	R/W

The RX FIFO Configuration / Control Registers are used to configure and control the 2 RX FIFOs.

- **RFIGCV[2:0]**
  These bits select the counter value of the FIFO for generation of FIFO Interrupt. These values represent fractions of the FIFO depth for which Interrupt is generated.

These bits cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.
The setting of these bits should be synchronized with the **RFCCk.RFDC[2:0]** bits. Refer to **19.8.2.1.5 FIFO Interrupt Configuration** for detailed information.
Write these bits only when RS-CANFD lite module is in GL_RESET mode.

- **RFIM**

  This bit selects the Interrupt generation condition for the FIFO.
  This bit cannot be writte when the RS-CANFD lite module is in GL_SLEEP mode.
  Write this bit only when RS-CANFD lite module is in GL_RESET mode.

- **RFDC[2:0]**

  These bits select the depth of the FIFO in terms of number of Messages. If the FIFO depth is configured to 0 Messages then the FIFO cannot be used.
  Write these bits only when RS-CANFD lite module is in GL_RESET mode.

- **RFPLS[2:0]**

  These bits define the message data payload allocation in the RAM.
  This is the max. number of Bytes which can be received by this FIFO.
  Refer to **19.8.2.1.4 FIFO Payload Size Configuration** for details.
  Write these bits only when RS-CANFD lite module is in GL_RESET mode.

- **RFIE**

  This bit enables generation of the FIFO Interrupt when it is set.
  This bit cannot be written when RS-CANFD lite module is in GL_SLEEP mode.

- **RFE**

  This bit enables the FIFO when it is set. If this bit is cleared, the RX FIFO will be cleared and is empty.
  Write this bit only when RS-CANFD lite module is in GL_HALT or GL_OPERATION modes.
  This bit can only be set if the configured FIFO depth is greater than 0 and less than 4 (000B < **RFCCk.RFDC[2:0]** ≤ 011B).
  The **RFCCk.RFE** bit should be set by a separate write access to the **RFCCk** register, after all the other bits of the **RFCCk** register have been set.
  This bit is cleared automatically when RS-CANFD lite module enters GL_RESET mode.

19.3.26 RX FIFO Status Register k (RFSTSks) [k = 0, 1] [s = 0, 1]

Address: RFSTSks: F0344H + (4H×k)
RFSTSksL: F0344H + (4H×k), RFSTSksH: F0345H + (4H×k)

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RFSTSks	–	–	RFMC[5:0]						–	–	–	–	RFIF	RF MLT	RF FLL	RF EMP
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 read as 0. The write value should be always 0.	R/W
13 to 8	RFMC[5:0]	RX FIFO Message Count	Number of Messages stored in FIFO	R
7 to 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3	RFIF	RX FIFO Interrupt Flag	0: FIFO Interrupt condition not satisfied 1: FIFO Interrupt condition satisfied	R/W
2	RFMLT	RX FIFO Message Lost	0: No Message Lost in FIFO 1: FIFO Message Lost	R/W
1	RFFLL	RX FIFO Full	0: FIFO Not Full 1: FIFO Full	R
0	RFEMP	RX FIFO Empty	0: FIFO Not Empty 1: FIFO Empty	R

The RX FIFO Status Registers show the status of the messages stored in corresponding FIFO Buffers.

- **RFMC[5:0]**

  These bits indicate the number of CAN messages stored in the RX FIFO that can be read by the CPU.
  These bits are cleared automatically when the FIFO is disabled.
  These bits are cleared automatically when RS-CANFD lite module enters GL_RESET mode.

- **RFIF**

  This bit is set automatically when the configured Interrupt condition is satisfied.
  Write this bit only when RS-CANFD lite module is in GL_HALT or GL_OPERATION modes.
  The bit is cleared by writing 0 to it.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.
  Writing 1 has no influence on the bit values.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  The bit is cleared when RS-CANFD lite module enters GL_RESET mode.
  This bit will not be cleared automatically if the RX FIFO buffer is disabled.

- **RFMLT**

  This bit is set automatically whenever a message is lost due to attempted storage when the FIFO is already full.
  Write this bit only when RS-CANFD lite module is in GL_HALT or GL_OPERATION modes.
  The bit is cleared by writing 0 to it.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.
  Writing 1 has no influence on the bit values.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  The bit is cleared when RS-CANFD lite module enters GL_RESET mode.

- **RFFLL**

  This bit is set automatically when number of CAN messages stored in the FIFO matches the configured FIFO depth.
  This bit is cleared automatically when the number of CAN messages stored in the FIFO is less than the configured FIFO depth.
  This bit is cleared automatically when RX FIFO is disabled by setting the **RFCCk.RFE** bit to 0.
  This bit is cleared automatically when RS-CANFD lite module enters GL_RESET mode.

- **RFEMP**

  This bit is set automatically when **RFSTSk.RFMC[5:0]** is 000000B.
  This bit is set automatically when RX FIFO is disabled by setting the **RFCCk.RFE** bit to 0.
  This bit is set automatically when RS-CANFD lite module enters GL_RESET mode.
  This bit is cleared automatically when the first message is stored in the RX FIFO buffer.

19.3.27 RX FIFO Pointer Control Register ks (RFPCTRks) [k = 0, 1] [s = 0, 1]

Address: RFPCTRks: F034CH + (4H×k)
RFPCTRksL: F034CH + (4H×k)

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RFPCTRks	–	–	–	–	–	–	–	–	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	These bits are read as 0. The write value should be always 0.	R/W
7 to 0	RFPC[7:0]	RX FIFO Pointer Control	Increments read pointer of the corresponding RX FIFO Buffers	W

These registers can be used to increment the read pointer of the corresponding RX FIFO buffers.

- **RFPC[7:0]**
  When the value FFH is written to these bits, then the pointer of the corresponding RX FIFO buffer is moved to the next FIFO entry.
  Read value from these bits is always 00H.
  Write these bits only when RS-CANFD lite module is in GL_HALT or GL_OPERATION mode.
  These bits can only write the value of FFH when the corresponding RX FIFO is enabled and not empty.

19.3.28 Common FIFO Configuration / Control Register (CFCCLs, CFCCHs) [s = 0, 1]

Address: CFCCL: F0354H, CFCCH: F0356H
CFCCLL: F0354H, CFCCLH: F0355H, CFCCHL: F0356H, CFCCHH: F0357H

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CFCCHs	CFITT[7:0]								CFDC[2:0]			–	–	–	CFTML[1:0]	
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CFCCLs	CFIGCV[2:0]			CFIM	CFITR	CFITSS	–	CFM	–	CFPLS[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
31 to 24	CFITT[7:0]	Common FIFO Interval Transmission Time	Delay the start of transmission from the FIFO if configured in TX mode, delay is a multiple of basic Interval Timer Clock Source unit	R/W
23 to 21	CFDC[2:0]	Common FIFO Depth Configuration	000B: FIFO Depth = 0 Messages 001B: FIFO Depth = 4 Messages 010B: FIFO Depth = 8 Messages 011B: FIFO Depth = 16 Messages 100B: FIFO Depth = Reserved 101B: FIFO Depth = Reserved 110B: FIFO Depth = Reserved 111B: FIFO Depth = Reserved	R/W
20 to 18	–	Reserved	There bits are read as 0. The write value should be always 0.	R/W
17, 16	CFTML[1:0]	Common FIFO TX Message Buffer Link	Transmission scan link position of the corresponding channel	R/W
15 to 13	CFIGCV[2:0]	Common FIFO Interrupt Generation Counter Value	000B: Interrupt generated when FIFO is 1/8th Full 001B: Interrupt generated when FIFO is 1/4th Full 010B: Interrupt generated when FIFO is 3/8th Full 011B: Interrupt generated when FIFO is 1/2 Full 100B: Interrupt generated when FIFO is 5/8th Full 101B: Interrupt generated when FIFO is 3/4th Full 110B: Interrupt generated when FIFO is 7/8th Full 111B: Interrupt generated when FIFO is Full	R/W
12	CFIM	Common FIFO Interrupt Mode	0: • RX FIFO Mode: RX Interrupt generated when Common FIFO counter reaches CFIGCV value from a lower value; • TX FIFO Mode: TX Interrupt generated when Common FIFO transmits the last message successfully; 1: • RX FIFO Mode: RX Interrupt generated at the end of every received message storage; • TX FIFO Mode: Interrupt generated for every successfully transmitted message;	R/W
11	CFITR	Common FIFO Interval Timer Resolution	0: Reference Clock Period x1 1: Reference Clock Period x10	R/W
10	CFITSS	Common FIFO Interval Timer Source Select	0: Reference Clock (x1 / x10 period) 1: Bit Time Clock of related channel (FIFO is linked to fixed channel)	R/W
9	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
8	CFM	Common FIFO Mode	0: RX FIFO Mode 1: TX FIFO Mode	R/W
7	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
6 to 4	CFPLS[2:0]	Common FIFO Payload Data Size Configuration	000B: 8 Bytes 001B: 12 Bytes 010B: 16 Bytes 011B: 20 Bytes 100B: 24 Bytes 101B: 32 Bytes 110B: 48 Bytes 111B: 64 Bytes	R/W
3	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
2	CFTXIE	Common FIFO TX Interrupt Enable	0: FIFO Interrupt generation disabled for Frame TX 1: FIFO Interrupt generation enabled for Frame TX	R/W
1	CFRXIE	Common FIFO RX Interrupt Enable	0: FIFO Interrupt generation disabled for Frame RX 1: FIFO Interrupt generation enabled for Frame RX	R/W
0	CFE	Common FIFO Enable	0: FIFO disabled 1: FIFO enabled	R/W

- **CFITT[7:0]**

  These bits select the delay in the start of transmission for all messages transmitted from this FIFO when configured in TX mode. The delay is a multiple of the basic Interval Timer Clock Source period (Reference Clock x1, Reference Clock x10 or Bit Time Clock of the related CAN channel).
  These bits cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.
  Do not write to these bits when the **CFCC.CFE** bit is set to 1.
  For **GCFG.ITRCP[15:0]** = 0000H these bits should only be set to 00H.

- **CFDC[2:0]**

  These bits select the depth of the Common FIFO in terms of number of Messages. If the FIFO depth is configured to 0 Messages then the FIFO cannot be used.
  Write these bits only when RS-CANFD lite module is in GL_RESET mode.

- **CFTML[1:0]**

  These bits select the normal transmit Message Buffer position where the TX FIFO is linked to, for transmission scanning.
  These bits cannot be written in GL_OPERATION or GL_SLEEP modes.
  Write these bits only when RS-CANFD lite module is in GL_RESET mode.

- **CFIGCV[2:0]**

  These bits select the message counter value for the generation of the FIFO Interrupt. These values represent fractions of the FIFO depth at which Interrupt is to be generated.
  These bits cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.
  The setting of these bits should be synchronized with the **CFCC.CFDC[2:0]** bits. Refer to **19.8.2.1.5 FIFO Interrupt Configuration** for detailed information.
  Write these bits only when RS-CANFD lite module is in GL_RESET mode.

- **CFIM**

  This bit selects the Interrupt generation condition for the FIFO.
  This bit cannot be written in GL_SLEEP mode.
  Write this bit only when RS-CANFD lite module is in GL_RESET mode.

- **CFITR**

  This bit selects the resolution of the Reference Clock for the Interval Transmission Timer (Peripheral Clock is the source for the Reference Clock).
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.
  Do not write to this bit when the **CFCC.CFE** bit is set to 1.

- **CFITSS**

  This bit selects the basic clock source for the Interval Transmission Timer.
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.
  Do not write to this bit when the **CFCC.CFE** bit is set to 1.
  Do not write 1 to this bit when CAN-FD communication will be used.
  Note bit time clock could be variable depending on the nominal and data rate bit configuration.

- **CFM**

  This bit selects the Mode of the FIFO. When CAN Reset is active, all the Common FIFO Buffers will be configured in RX FIFO mode.
  This bit cannot be written in GL_OPERATION or GL_SLEEP modes.
  Write this bit only when RS-CANFD lite module is in GL_RESET mode.

- **CFPLS[2:0]**

  These bits define the message data payload allocation in the RAM.
  This is the max. number of Bytes which can be received or transmitted by this FIFO.
  Refer to **19.8 FIFO Buffers and Normal MB Configuration** for details.
  Write these bits only when RS-CANFD lite module is in GL_RESET mode.

- **CFTXIE**

  This bit enables the generation of the Common FIFO Interrupt when the Interrupt flag is set after transmission of a frame from the corresponding FIFO.
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.

- **CFRXIE**

  This bit enables generation of the FIFO Interrupt when the Interrupt flag is set after reception of a frame in the corresponding FIFO.
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.

- **CFE**

  This bit enables the FIFO when it is set.
  FIFO is disabled when this bit is cleared.
  This bit can also be used, by clearing it, to abort transmission from Common FIFO when configured in TX Mode.
  This bit can also be used, by clearing it, to stop reception into the Common FIFO in RX mode.
  Write this bit when RS-CANFD lite module is in GL_HALT or GL_OPERATION modes and the related RS-CANFD lite channel is not in CH_RESET mode for FIFOs configured as TX FIFO.
  This bit can only be set if the configured FIFO depth is greater than 0 and less than 4 (000B < **CFCCHs.CFDC[2:0]** ≤ 011B).
  The **CFCC.CFE** bit should be set by a separate write access to the **CFCC** register, after all the other bits of the **CFCC** register have been set.
  This bit is cleared automatically when RS-CANFD lite module enters GL_RESET mode.
  This bit is cleared automatically when the related channel enters CH_RESET mode if the FIFO is configured in TX mode.

19.3.29 Common FIFO Status Register (CFSTSs) [s = 0, 1]

Address: CFSTSs: F0358H
CFSTSsL: F0358H, CFSTSsH: F0359H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CFSTSs	–	–	CFMC[5:0]						–	–	–	CFTX IF	CFRX IF	CF MLT	CF FLL	CF EMP
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 read as 0. The write value should be always 0.	R/W
13 to 8	CFMC[5:0]	Common FIFO Message Count	No. of Messages stored in FIFO	R
7 to 5	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
4	CFTXIF	Common TX FIFO Interrupt Flag	0: FIFO Interrupt condition not satisfied after Frame Transmission 1: FIFO Interrupt condition satisfied after Frame Transmission	R/W
3	CFRXIF	Common RX FIFO Interrupt Flag	0: FIFO Interrupt condition not satisfied after Frame Reception 1: FIFO Interrupt condition satisfied after Frame Reception	R/W
2	CFMLT	Common FIFO Message Lost	0: No Message Lost in FIFO 1: FIFO Message Lost	R/W
1	CFFLL	Common FIFO Full	0: FIFO Not Full 1: FIFO Full	R
0	CFEMP	Common FIFO Empty	0: FIFO Not Empty 1: FIFO Empty	R

- **CFMC[5:0]**

  These bits indicate the following:
  Number of CAN messages stored by the CPU in the FIFO configured in TX Mode pending for transmission.
  Number of CAN messages stored in the FIFO Buffer configured in RX Mode by RS-CANFD lite to be read by the CPU.
  These bits are cleared automatically under the following conditions:
  - When the FIFO is disabled.
  - When RS-CANFD lite module enters GL_RESET mode.
  - When the related channel enters CH_RESET mode when the FIFO is configured in TX mode.

- **CFTXIF**

  This bit will not be cleared automatically if the Common FIFO Buffer is disabled.
  Write this bit only when RS-CANFD lite module is in GL_HALT or GL_OPERATION mode and the related RS-CANFD lite channel is not in CH_RESET mode for FIFOs configured as TX FIFO.
  Writing 1 has no influence on the bit values.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.
  This bit is set automatically when the configured Interrupt condition is satisfied for Common FIFO Buffers configured in TX mode.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  This bit is cleared under the following conditions.
  - Writing 0 to this bit.
  - When RS-CANFD lite module enters GL_RESET mode.
  - When the related channel enters CH_RESET mode when the FIFO is configured in TX mode.

- **CFRXIF**

  This bit will not be cleared automatically if the Common FIFO Buffer is disabled.
  Write this bit only when RS-CANFD lite module is in GL_HALT or GL_OPERATION mode and the related RS-CANFD lite channel is not in CH_RESET mode for FIFOs configured as TX FIFO.
  Writing 1 has no influence on the bit values.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.
  This bit is set automatically when the configured Interrupt condition is satisfied for Common FIFO Buffers when configured in RX mode.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  The bit is cleared by writing 0 to it.
  The bit is cleared when RS-CANFD lite module enters GL_RESET mode.

- **CFMLT**

  Write this bit only when RS-CANFD lite module is in GL_HALT or GL_OPERATION mode and the related RS-CANFD lite channel is not in CH_RESET mode for FIFOs configured as TX FIFO.
  Writing 1 has no influence on the bit values.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.
  This bit is set automatically whenever a message is lost due to attempted storage of a new message when FIFO is already full in RX mode.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  This bit is cleared under the following conditions.
  - Writing 0 to this bit.
  - When RS-CANFD lite module enters GL_RESET mode.
  - When the related channel enters CH_RESET mode when the FIFO is configured in TX mode.

- **CFFLL**

  This bit is set automatically when the number of CAN messages stored in the FIFO matches the configured FIFO depth.
  This bit is cleared automatically under the following conditions.
  - When the number of CAN messages stored in the FIFO is less than the configured FIFO depth.
  - When the FIFO is disabled by setting the CFE bit to 0.
  - When module enters GL_RESET.
  - When FIFO is configured in TX Mode and module enters CH_RESET.

- **CFEMP**

  This bit is set automatically under the following conditions.
  - When the CPU has read all messages from the FIFO configured in RX mode.
  - When all messages have been transmitted from the FIFO configured in TX Mode.
  - When FIFO is disabled by setting the CFE bit to 0.
  - When module enters GL_RESET mode.
  - When FIFO is configured in TX Mode and module enters CH_RESET.

  This bit is cleared automatically under the following conditions.
  - When the first reception message is stored in the FIFO when configured in RX Mode.
  - When the first message to be transmitted is stored in the FIFO when configured in TX Mode.

19.3.30 Common FIFO Pointer Control Register (CFPCTRs) [s = 0, 1]

Address: CFPCTRs: F035CH
CFPCTRsL: F035CH

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CFPCTRs	–	–	–	–	–	–	–	–	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	These bits are read as 0. The write value should be always 0.	R/W
7 to 0	CFPC[7:0]	Common FIFO Pointer Control	Increments read or write pointer of the corresponding Common FIFO Buffers depending upon the mode configuration	W

This register can be used to increment the read or write pointer of the corresponding Common FIFO.

- **CFPC[7:0]**
  When the value FFH is written into these bits, then the Read Pointer of the corresponding Common FIFO buffer, when configured in RX mode, or the write pointer of the corresponding Common FIFO buffer, when configured in TX mode, moves to the next FIFO entry.
  Read value from these bits is always 00H.
  Write these bits only when RS-CANFD lite module is in GL_HALT or GL_OPERATION modes.
  Write FFH to these bits only when the Common FIFO is enabled and is not empty if configured in RX mode.
  Write FFH to these bits only when the Common FIFO is enabled and is not full if configured in TX mode.

19.3.31 FIFO Empty Status Register (FESTSs) [s = 0, 1]

Address: FESTSs: F0360H
FESTSsL: F0360H, FESTSsH: F0361H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FESTSs	–	–	–	–	–	–	–	CF EMP	–	–	–	–	–	–	RF1 EMP	RF0 EMP
After Reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
15 to 9	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
8	CFEMP	Common FIFO Empty Status	0: Corresponding FIFO is not Empty 1: Corresponding FIFO is Empty	R
7 to 2	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
1	RF1EMP	RX FIFO 1 Empty Status	0: Corresponding FIFO is not Empty 1: Corresponding FIFO is Empty	R
0	RF0EMP	RX FIFO 0 Empty Status	0: Corresponding FIFO is not Empty 1: Corresponding FIFO is Empty	R

The FIFO Empty Status Register bits show the status of the empty bits of the FIFO Buffers.

- **CFEMP**

  This bit is set when the RS-CANFD lite module enters GL_RESET mode.
  Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.
  Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

- **RF1EMP, RF0EMP**

  This bit is set when the RS-CANFD lite module enters GL_RESET mode.
  Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.
  Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

19.3.32 FIFO Full Status Register (FFSTSs) [s = 0, 1]

Address: FFSTSs: F0364H
FFSTSsL: F0364H, FFSTSsH: F0365H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FFSTSs	–	–	–	–	–	–	–	CF FLL	–	–	–	–	–	–	RF1 FLL	RF0 FLL
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 9	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
8	CFFLL	Common FIFO Full Status	0: Corresponding FIFO is not Full 1: Corresponding FIFO is Full	R
7 to 2	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
1	RF1FLL	RX FIFO 1 Full Status	0: Corresponding FIFO is not Full 1: Corresponding FIFO is Full	R
0	RF0FLL	RX FIFO 0 Full Status	0: Corresponding FIFO is not Full 1: Corresponding FIFO is Full	R

The FIFO Full Status Register bits show the status of the full bits of the FIFO Buffers.

- **CFFLL**

  Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.
  Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.
  This bit is cleared when RS-CANFD lite module enters GL_RESET mode.

- **RF1FLL, RF0FLL**

  Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.
  Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.
  This bit is cleared when RS-CANFD lite module enters GL_RESET mode.

19.3.33 FIFO Message Lost Status Register (FMSTSs) [s = 0, 1]

Address: FMSTSs: F0368H
FMSTSsL: F0368H, FMSTSsH: F0369H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FMSTSs	–	–	–	–	–	–	–	CF MLT	–	–	–	–	–	–	RF1 MLT	RF0 MLT
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 9	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
8	CFMLT	Common FIFO Msg Lost Status	0: Corresponding FIFO Msg Lost flag is not set 1: Corresponding FIFO Msg Lost flag is set	R
7 to 2	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
1	RF1MLT	RX FIFO 1 Msg Lost Status	0: Corresponding FIFO Msg Lost flag is not set 1: Corresponding FIFO Msg Lost flag is set	R
0	RF0MLT	RX FIFO 0 Msg Lost Status	0: Corresponding FIFO Msg Lost flag is not set 1: Corresponding FIFO Msg Lost flag is set	R

The FIFO Message Lost status Register bits show the status of the message lost bits of the FIFO buffers.

- **CFMLT**

  This bit is cleared when RS-CANFD lite module enters GL_RESET mode.
  Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.
  Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

- **RF1MLT, RF0MLT**

  This bit is cleared when RS-CANFD lite module enters GL_RESET mode.
  Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.
  Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

19.3.34 RX FIFO Interrupt Flag Status Register (RFISTSs) [s = 0, 1]

Address: RFISTSs: F036CH
RFISTSsL: F036CH

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RFISTSs	–	–	–	–	–	–	–	–	–	–	–	–	–	–	RF1IF	RF0IF
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 2	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
1	RF1IF	RX FIFO 1 Interrupt Flag Status	0: Corresponding RX FIFO interrupt flag not set 1: Corresponding RX FIFO interrupt flag set	R
0	RF0IF	RX FIFO 0 Interrupt Flag Status	0: Corresponding RX FIFO interrupt flag not set 1: Corresponding RX FIFO interrupt flag set	R

The FIFO Interrupt Flag Status Register bits show the status of the interrupt flag bits of the RX FIFO buffers.

- **RF1IF, RF0IF**

  Each bit is set automatically when the corresponding interrupt flag bit is set in the RX FIFO Status Registers.
  This bit is cleared when RS-CANFD lite module enters GL_RESET mode.
  Each bit is cleared automatically when the corresponding interrupt flag bit is cleared in the RX FIFO Status Registers.

19.3.35 CANs TX Message Buffer Control Register m (TMCms) [m = 0 to 3] [s = 0, 1]

Address: TMCms: F0370H + m

Symbol	b7	b6	b5	b4	b3	b2	b1	b0
TMCms	–	–	–	–	–	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 read as 0. The write value should be always 0.	R/W
2	TMOM	TX Message Buffer One-shot Mode	0: TX Message Buffer not configured in one-shot mode 1: TX Message Buffer configured in one-shot mode	R/W
1	TMTAR	TX Message Buffer Transmission Abort Request	0: TX Message Buffer transmission request abort not requested 1: TX Message Buffer transmission request abort requested	R/W
0	TMTR	TX Message Buffer Transmission Request	0: TX Message Buffer Transmission not requested 1: TX Message Buffer Transmission requested	R/W

The CANs TX Message Buffer Control Register configures the TX Message Buffer functions.

- **TMOM**

  If this bit is set, then the RS-CANFD lite module logic will attempt transmission of the message only once.
  If the transmission is successful, the **TMSTSm.TMTRF[1:0]** bits are set to 10B or 11B. If it is not successful due to bus error or bus arbitration lost, the transmission is automatically aborted and **TMSTSm.TMTRF[1:0]** bits are set to 01B.
  The **TMCm.TMOM** bit will remain set if the transmission is completed successfully or aborted due to error or loss of arbitration.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.
  Set this bit at the same time as **TMCm.TMTR** bit.
  This bit is cleared by a write access.
  If a message has been already requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.
  This bit will be automatically cleared by the RS-CANFD lite module logic when the RS-CANFD lite module enters GL_RESET mode or the related channel enters CH_RESET mode.

- **TMTAR**

  If this bit is set, then the RS-CANFD lite module logic will try to abort the transmission of the frame stored in the corresponding Message Buffer.
  In most cases, transmission cannot be aborted if the internal scan for transmission is completed and the Message Buffer has already been selected for Transmission. In this case, frame may be transmitted successfully from the Message Buffer. The Message Buffer selection will be released by entering CH_HALT mode.
  However, MB selected for transmission can be aborted by Abort request when the CAN node detects a new message on the bus (RX pin) before it can start transmission from the selected MB.
  Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.
  This bit can only be set when the related transmit request (**TMCm.TMTR**) bit is set.
  This bit cannot be cleared by a CPU write access.
  Clearing of this bit by RS-CANFD lite has priority over setting by CPU write access.

This bit is cleared automatically under the following conditions.
- When the transmission is successful.
- When the transmission is abort.
- When CAN bus error or arbitration loss is detected.
- When the RS-CANFD lite module enters GL_RESET mode or the related channel enters CH_RESET mode.

- **TMTR**

If this bit is set, then the RS-CANFD lite module logic will try to transmit the message stored in the corresponding Message Buffer.
Write this bit only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.
Do not set this bit if the corresponding TX Message Buffer is linked to a COM FIFO in TX mode.
This bit cannot be directly cleared by a CPU write access.
This bit can only be set when Transmission Result Flag (**TMSTSm.TMTRF[1:0]**) corresponding to the MB are cleared to 00B.
This bit is cleared automatically under the following conditions.
- When the transmission is successful.
- When the transmission is abort when the corresponding TMCm.TMTAR bit is set to 1.
- When CAN bus error or arbitration loss is detected if TMCm.TMOM bit is set for Message Buffer.
- When the RS-CANFD lite module enters GL_RESET mode or the related channel enters CH_RESET mode.

19.3.36 CANs TX Message Buffer Status Register m (TMSTSms) [m = 0 to 3] [s = 0, 1]

Address: TMSTSms: F0374H + m

Symbol	b7	b6	b5	b4	b3	b2	b1	b0
TMSTSms	–	–	–	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 read as 0. The write value should be always 0.	R/W
4	TMTARM	TX Message Buffer Transmission Abort Request Mirrored	0: TX Message Buffer transmission request abort not requested 1: TX Message Buffer transmission request abort requested	R
3	TMTRM	TX Message Buffer Transmission Request Mirrored	0: TX Message Buffer Transmission not requested 1: TX Message Buffer Transmission requested	R
2, 1	TMTRF[1:0]	TX Message Buffer Transmission Result Flag	00B: No Result 01B: Transmission aborted from the TX MB 10B: Transmission successful from the TX MB & Transmission abort was not requested 11B: Transmission successful from the TX MB & Transmission abort was requested	R/W
0	TMTSTS	TX Message Buffer Transmission Status	0: No transmission ongoing 1: Transmission ongoing	R

The TX Message Buffer Status Register bits show the transmission and transmission abort status for the corresponding message buffers.

- **TMTARM**

  This bit is set when the corresponding **TMCm.TMTAR** bit is set.
  This bit is cleared when the corresponding **TMCm.TMTAR** bit is cleared.

- **TMTRM**

  This bit is set when the corresponding **TMCm.TMTR** bit is set.
  This bit is cleared when the corresponding **TMCm.TMTR** bit is cleared.

- **TMTRF[1:0]**

  These bits show the result for the corresponding TX MB. The status is as follows:

  00B: Transmission in progress or has not been requested.
  01B: Transmission has been aborted from the corresponding TX MB.
  10B: Transmission was successful from the corresponding TX MB and **TMCm.TMTAR** bit was not set for this TX Message Buffer.
  11B: Transmission was successful from the corresponding TX MB. But **TMCm.TMTAR** bit was set for this TX MB.

  Write these bits only when the related RS-CANFD lite channel is in CH_HALT or CH_OPERATION mode.
  These bits will be cleared automatically when the RS-CANFD lite module enters GL_RESET or the related channel enters CH_RESET mode.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the **TMSTSm.TMTRF[1:0]** are set.

- **TMTSTS**

  This bit is set automatically at the start of the transmission from the corresponding TX Message Buffer.

  This bit is cleared automatically under the following conditions.

  - When the transmission stops.
  - When the RS-CANFD lite module enters GL_RESET mode.
  - When the related channel enters CH_RESET mode.

19.3.37 CANs TX Message Buffer Transmission Request Status Register (TMTRSTSs) [s = 0, 1]

Address: TMTRSTSs: F0378H
TMTRSTSsL: F0378H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TMTRSTSs	–	–	–	–	–	–	–	–	–	–	–	–	TMTRSTS[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 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3	TMTRSTS3	TX Message Buffer 3 Transmission Request Status	0: Transmission not requested for TX Message Buffer 3 1: Transmission requested for TX Message Buffer 3	R
2	TMTRSTS2	TX Message Buffer 2 Transmission Request Status	0: Transmission not requested for TX Message Buffer 2 1: Transmission requested for TX Message Buffer 2	R
1	TMTRSTS1	TX Message Buffer 1 Transmission Request Status	0: Transmission not requested for TX Message Buffer 1 1: Transmission requested for TX Message Buffer 1	R
0	TMTRSTS0	TX Message Buffer 0 Transmission Request Status	0: Transmission not requested for TX Message Buffer 0 1: Transmission requested for TX Message Buffer 0	R

These bits show the TX message buffer transmission request status for the corresponding TX Message Buffer.
The bit position of **TMTRSTS** register corresponds to the buffer number of TX Message Buffer.

e.g. The bit 0 of the **TMTRSTS** register corresponds to the TX Message Buffer 0.

- **TMTRSTS[3:0]**
  These bits show the status of the **TMCm.TMTR** bit of the CANs TX Message Buffer Control Register.
  Each bit is cleared automatically when the corresponding bit is cleared in the CANs TX Message Buffer Control Register.
  Each bit is cleared automatically when the RS-CANFD lite module enters GL_RESET or CH_RESET mode.

19.3.38 CANs TX Message Buffer Transmission Abort Request Status Register (TMTARSTSs) [s = 0, 1]

Address: TMTARSTSs: F037CH
TMTARSTSsL: F037CH

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TMTARSTSs	–	–	–	–	–	–	–	–	–	–	–	–	TMTARSTS[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 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3	TMTARSTS3	TX Message Buffer 3 Transmission Abort Request Status	0: Transmission abort not requested for TX Message Buffer 3 1: Transmission abort requested for TX Message Buffer 3	R
2	TMTARSTS2	TX Message Buffer 2 Transmission Abort Request Status	0: Transmission abort not requested for TX Message Buffer 2 1: Transmission abort requested for TX Message Buffer 2	R
1	TMTARSTS1	TX Message Buffer 1 Transmission Abort Request Status	0: Transmission abort not requested for TX Message Buffer 1 1: Transmission abort requested for TX Message Buffer 1	R
0	TMTARSTS0	TX Message Buffer 0 Transmission Abort Request Status	0: Transmission abort not requested for TX Message Buffer 0 1: Transmission abort requested for TX Message Buffer 0	R

These bits show the TX message buffer transmission abort request status for the corresponding TX Message Buffer.
The bit position of **TMTARSTS** register corresponds to the buffer number of TX Message Buffer.

e.g. The bit 0 of the **TMTARSTS** register corresponds to the TX Message Buffer 0.

- **TMTARSTS[3:0]**

  These bits show the status of the **TMCm.TMTAR** bit of the CANs TX Message Buffer Control Register.
  Each bit is set automatically when the corresponding bit is set in the CANs TX Message Buffer Control Register.
  Each bit is cleared automatically when the corresponding bit is cleared in the CANs TX Message Buffer Control Register.
  Each bit is cleared automatically when the RS-CANFD lite module enters GL_RESET or CH_RESET mode.
  If a CAN channel enters CH_RESET mode, then the bits related to that channel will be cleared.

19.3.39 CANs TX Message Buffer Transmission Completion Status Register (TMTCSTSs) [s = 0, 1]

Address: TMTCSTSs: F0380H
TMTCSTSsL: F0380H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TMTCSTSs	–	–	–	–	–	–	–	–	–	–	–	–	TMTCSTS[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 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3	TMTCSTS3	TX Message Buffer 3 Transmission Completion Status	0: Transmission not complete for TX Message Buffer 3 1: Transmission completed for TX Message Buffer 3	R
2	TMTCSTS2	TX Message Buffer 2 Transmission Completion Status	0: Transmission not complete for TX Message Buffer 2 1: Transmission completed for TX Message Buffer 2	R
1	TMTCSTS1	TX Message Buffer 1 Transmission Completion Status	0: Transmission not complete for TX Message Buffer 1 1: Transmission completed for TX Message Buffer 1	R
0	TMTCSTS0	TX Message Buffer 0 Transmission Completion Status	0: Transmission not complete for TX Message Buffer 0 1: Transmission completed for TX Message Buffer 0	R

These bits show the TX message buffer transmission completion status for the corresponding TX Message Buffer.
The bit position of **TMTCSTS** register corresponds to the buffer number of TX Message Buffer.

e.g. The bit 0 of the **TMTCSTS** register corresponds to the TX Message Buffer 0.

- **TMTCSTS[3:0]**
  These bits show the status of successful completion of the CANs TX Message Buffer Status Register.
  Each bit is set automatically when the corresponding bit is set in the CANs TX Message Buffer Status Register.
  Each bit is cleared automatically when the corresponding bit is cleared in the CANs TX Message Buffer Status Register.
  Each bit is cleared automatically when the RS-CANFD lite module enters GL_RESET or CH_RESET mode.
  If a CAN channel enters CH_RESET mode, then the bits related to that channel will be cleared.

19.3.40 CANs TX Message Buffer Transmission Abort Status Register (TMTASTSs) [s = 0, 1]

Address: TMTASTSs: F0384H
TMTASTSsL: F0384H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TMTASTSs	–	–	–	–	–	–	–	–	–	–	–	–	TMTASTS[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 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3	TMTASTS3	TX Message Buffer 3 Transmission Abort Status	0: Transmission not aborted for TX Message Buffer 3 1: Transmission aborted for TX Message Buffer 3	R
2	TMTASTS2	TX Message Buffer 2 Transmission Abort Status	0: Transmission not aborted for TX Message Buffer 2 1: Transmission aborted for TX Message Buffer 2	R
1	TMTASTS1	TX Message Buffer 1 Transmission Abort Status	0: Transmission not aborted for TX Message Buffer 1 1: Transmission aborted for TX Message Buffer 1	R
0	TMTASTS0	TX Message Buffer 0 Transmission Abort Status	0: Transmission not aborted for TX Message Buffer 0 1: Transmission aborted for TX Message Buffer 0	R

These bits show the TX message buffer transmission abort status for the corresponding TX Message Buffer.
The bit position of **TMTASTS** register corresponds to the buffer number of TX Message Buffer.

e.g. The bit 0 of the **TMTASTS** register corresponds to the TX Message Buffer 0.

- **TMTASTS[3:0]**
  These bits show the status of the successful transmission abort of the corresponding TX Message Buffer.
  Each bit is set automatically when the **TMSTSm.TMTRF[1:0]** bits are set to 01B in the corresponding CANs TX Message Buffer Status Register.
  Each bit is cleared automatically when the **TMSTSm.TMTRF[1:0]** bits are cleared in the corresponding CANs TX Message Buffer Status Register.
  Each bit is cleared automatically when the RS-CANFD lite module enters GL_RESET or CH_RESET mode.
  If a CAN channel enters CH_RESET mode, then the bits related to that channel will be cleared.

19.3.41 CANs TX Message Buffer Interrupt Enable Configuration Register (TMIECs) [s= 0, 1]

Address: TMIECs: F0388H
TMIECsL: F0388H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TMIECs	–	–	–	–	–	–	–	–	–	–	–	–	TMIE[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 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3	TMIE3	TX Message Buffer 3 Interrupt Enable	0: TX Message Buffer 3 interrupt disabled 1: TX Message Buffer 3 interrupt enabled	R/W
2	TMIE2	TX Message Buffer 2 Interrupt Enable	0: TX Message Buffer 2 interrupt disabled 1: TX Message Buffer 2 interrupt enabled	R/W
1	TMIE1	TX Message Buffer 1 Interrupt Enable	0: TX Message Buffer 1 interrupt disabled 1: TX Message Buffer 1 interrupt enabled	R/W
0	TMIE0	TX Message Buffer 0 Interrupt Enable	0: TX Message Buffer 0 interrupt disabled 1: TX Message Buffer 0 interrupt enabled	R/W

These bits show the TX message buffer interrupt enable for the corresponding TX Message Buffer.
The bit position of **TMIEC** register corresponds to the buffer number of TX Message Buffer.

e.g. The bit 0 of a **TMIEC** register corresponds to the TX Message Buffer 0.

- **TMIE[3:0]**

  If this bit is set, then an interrupt will be generated at the end of a successful transmission from the corresponding Message Buffer.
  Refer to **19.12 Interrupt** for TX Message Buffer interrupt specification.
  This bit cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.
  Do not write to this bit when the related CAN_channel is in CH_SLEEP mode.
  Do not write to this bit if the corresponding TX Message Buffer is linked to a Common FIFO (via **CFCC.CFTML[1:0]** bits)

19.3.42 CANs TX History List Configuration / Control Register (THLCCs) [s = 0, 1]

Address: THLCCs: F0398H
THLCCsL: F0398H, THLCCsH: F0399H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
THLCCs	–	–	–	–	–	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 read as 0. The write value should be always 0.	R/W
10	THLDTE	TX History List Dedicated TX Enable	0: TX FIFO 1: Flat TX MB + TX FIFO	R/W
9	THLIM	TX History List Interrupt Mode	0: Interrupt generated if TX History List level reaches 6/8 of the TX History List depth. 1: Interrupt generated for every successfully stored entry	R/W
8	THLIE	TX History List Interrupt Enable	0: TX History List Interrupt disabled 1: TX History List Interrupt enabled	R/W
7 to 1	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
0	THLE	TX History List Enable	0: TX History List disabled 1: TX History List enabled	R/W

The CANs TX History List Configuration / Control Register configures the TX History List functions.

- **THLDTE**

  This bit selects the conditions for storing an entry in the TX History list after successful transmission.
  This bit cannot be written when RS-CANFD lite module is in GL_SLEEP mode.
  Do not write to this bit when RS-CANFD lite module is in GL_HALT or GL_OPERATION mode.

- **THLIM**

  This bit selects the Interrupt generation condition for the FIFO.
  This bit cannot be written when RS-CANFD lite module is in GL_SLEEP mode.
  Do not write to this bit when RS-CANFD lite module is in GL_HALT or GL_OPERATION mode.

- **THLIE**

  This bit enables the generation of the TX History List Interrupt when it is set.
  This bit cannot be written when RS-CANFD lite module is in GL_SLEEP mode.

- **THLE**

  This bit enables the TX History List Buffer when it is set.
  This bit cannot be written when the related RS-CANFD lite channel is in CH_RESET or CH_SLEEP mode.
  This bit is cleared automatically when the related RS-CANFD lite channel is in CH_RESET mode.

19.3.43 CANs TX History List Status Register (THLSTSs) [s = 0, 1]

Address: THLSTSs: F039CH
THLSTSsL: F039CH, THLSTSsH: F039DH

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
THLSTSs	–	–	–	–	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 read as 0. The write value should be always 0.	R/W
11 to 8	THLMC[3:0]	TX History List Message Count	Number of messages stored in TX History List	R
7 to 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3	THLIF	TX History List Interrupt Flag	0: TX History List Interrupt condition not satisfied 1: TX History List Interrupt condition satisfied	R/W
2	THLELT	TX History List Entry Lost	0: No Entry Lost in TX History List 1: TX History List Entry Lost	R/W
1	THLFLL	TX History List Full	0: TX History List Not Full 1: TX History List Full	R
0	THLEMP	TX History List Empty	0: TX History List Not Empty 1: TX History List Empty	R

The CANs TX History List Status Register shows the status of the data stored in the TX History List Buffer.

- **THLMC[3:0]**
  These bits show the number of transmitted messages stored in the TX History List.
  These bits are cleared automatically when the related channel is in CH_RESET mode.

- **THLIF**
  Write this bit only when RS-CANFD lite module is in CH_HALT or CH_OPERATION modes.
  Writing 1 has no influence on the bit values.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.
  This bit is set when the configured Interrupt condition is satisfied.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  The bit is cleared by writing 0 to it.
  This bit is automatically cleared in CH_RESET mode.

- **THLELT**
  Write this bit only when RS-CANFD lite module is in CH_HALT or CH_OPERATION modes.
  Writing 1 has no influence on the bit values.
  Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.
  This bit is set when a new Entry cannot be stored as the related TX History List Buffer is already full.
  If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
  This bit is cleared by writing 0 to it.
  This bit is automatically cleared in CH_RESET mode.

- **THLFLL**

  Each TX History List can store up to 8 entries.

  This bit is set automatically when the number of entries in the TX History List Buffer matches the TX History List depth.

  This bit is cleared automatically under the following conditions.

  - When the number of entries in the TX History List Buffer is less than the TX History List depth.
  - When the TX History List is disabled.
  - When the corresponding CAN channel enters CH_RESET mode.

- **THLEMP**

  This bit is set automatically under the following conditions.

  - When CPU has read all entries from the TX History List Buffer.
  - When the TX History List is disabled.
  - When the corresponding CAN channel enters CH_RESET mode.

  This bit is cleared automatically when the first entry is stored to the TX History List.

19.3.44 CANs TX History List Access Register 0 (THLACC0Hs, THLACC0Ls) [s = 0, 1]

Address: THLACC0Ls: F0640H, THLACC0Hs: F0642H
THLACC0LsL: F0640H, THLACC0HsL: F0642H, THLACC0HsH: F0643H

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
THLACC0Hs	TMTS[15:0]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
THLACC0Ls	–	–	–	–	–	–	–	–	–	–	–	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
31 to 16	TMTS[15:0]	Transmit Timestamp	Transmit Timestamp value for SW drivers	R
15 to 5	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
4, 3	BN[1:0]	Buffer No.	Number of the Message Buffer	R
2	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
1, 0	BT[1:0]	Buffer Type	01B: TX Message Buffer 10B: TX FIFO Buffer Others: Setting is prohibited.	R

The CANs TX History List Access Register provides access to the entry in the TX History List based on the read timestamp value.

- **TMTS[15:0]**

  These bits indicate the Timestamp for use by the SW drivers.

- **BN[1:0]**

  These bits show the Message Buffer from which transmission was successfully completed. If a message from a Common FIFO is transmitted, then these bits show the Message Buffer that is linked to the Common FIFO for transmission.

- **BT[1:0]**

  These bits indicate the transmit source buffer type of transmit history data stored in the transmit history buffer.

19.3.45 CANs TX History List Access Register 1 (THLACC1Hs, THLACC1Ls) [s = 0, 1]

Address: THLACC1Ls: F0644H THLACC1Hs: F0646H
THLACC1LsL: F0644H, THLACC1LsH: F0645H, THLACC1HsL: F0646H

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
THLACC1Hs	–	–	–	–	–	–	–	–	–	–	–	–	–	–	TIFL[1:0]	
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
THLACC1Ls	TID[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
31 to 18	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
17, 16	TIFL[1:0]	Transmit Information Label	MB information label or TX FIFO information label or AFL information label is stored for SW drivers	R
15 to 0	TID[15:0]	Transmit ID	MB reference ID or TX FIFO references ID or AFL pointer field is stored for SW drivers.	R

The CANs TX History List Access Register 1 provides access to the entry in the TX History List based on the read pointer value.

- **TIFL[1:0]**
  These bits indicate the MB information label (**TMFDCTRm.TMIFL[1:0]**) or the TX FIFO information label (**CFFDCSTS.CFIFL[1:0]**) for use by the SW drivers.

- **TID[15:0]**
  These bits indicate the MB reference ID (**TMFDCTRm.TMPTR[15:0]**) or the TX FIFO reference ID (**CFFDCSTS.CFPTR[15:0]**) for use by the SW drivers.

19.3.46 CANs TX History List Pointer Control Register (THLPCTRs) [s = 0, 1]

Address: THLPCTRs: F03A0H
THLPCTRsL: F03A0H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
THLPCTRs	–	–	–	–	–	–	–	–	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	These bits are read as 0. The write value should be always 0.	R/W
7 to 0	THLPC[7:0]	TX History List Pointer Control	Increments the read pointer to the TX History List in the corresponding channel	W

This register can be used to increment the read pointer of the TX History List.

- **THLPC[7:0]**
  When FFH is written to these bits, the read pointer of the TX History List is moved to the next TX History List entry address.
  Read value from these bits are always 00H.
  Write these bits only when RS-CANFD lite module is in CH_HALT or CH_OPERATION modes.
  These bits can only write FFH when the corresponding TX History List is enabled and is not empty.

19.3.47 Global Reset Control Register (GRSTCs) [s = 0, 1]

Address: GRSTCs: F03D8H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GRSTCs	KEY[7:0]								–	–	–	–	–	–	–	SRST
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	KEY[7:0]	Key Code	These bits control the right or wrong of rewriting of a SRST bit.	W
7 to 1	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
0	SRST	SW Reset	0: normal state 1: SW reset state	R/W

- **KEY[7:0]**

  Writing C4H to these bits allows the **GRSTC.SRST** bit to be written.
  Read value from these bits is always 00H.
  Be sure to write the **GRSTC.SRST** bit and the **GRSTC.KEY[7:0]** bits simultaneously.

- **SRST**

  When this bit is set, RS-CANFD lite module will be in the same state as CAN reset.
  When reset of RS-CANFD lite module is required, write 1 to this bit. Then write 0 to this bit.
  When this bit is cleared, a RS-CANFD lite module is in GL_SLEEP mode.
  When this bit is cleared, the RAM initialization sequence does not operate. The configuration of RAM is performed by software.
  RAM is not initialized when software reset is performed during the initialization of RAM.
  Software needs to perform the initialization of RAM.
  Be sure to write the **GRSTC.SRST** bit and the **GRSTC.KEY[7:0]** bits simultaneously.

19.3.48 Global Test Configuration Register (GTSTCFGs) [s = 0, 1]

Address: GTSTCFGs: F03AAH
GTSTCFGsL: F03AAH

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GTSTCFGs	–	–	–	–	–	–	–	–	–	–	–	–	RTMPS[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 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3 to 0	RTMPS[3:0]	RAM Test Mode Page Select	Select a RAM Test Mode page	R/W

The Global Test Configuration Register is used to specify the RAM test page number.

- **RTMPS[3:0]**
  These bits specify the page numbet for the RAM test in RAM test mode.
  Refer to **19.11.2.1 RAM Test Mode** for details.
  These bits cannot be written when the RS-CANFD lite module is in GL_RESET or GL_SLEEP mode.
  Enter only values between 0 and 8 (1000B) for these bits.
  Write these bits only when RS-CANFD lite module is in GL_HALT mode.
  These bits are cleared automatically when the related RS-CANFD lite channel is in GL_RESET mode.

19.3.49 Global Test Control Register (GTSTCTRs) [s = 0, 1]

Address: GTSTCTRs: F03ACH
GTSTCTRsL: F03ACH

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GTSTCTRs	–	–	–	–	–	–	–	–	–	–	–	–	–	RTME	–	–
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 3	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
2	RTME	RAM Test Mode Enable	0: RAM Test Mode disabled 1: RAM Test Mode enabled	R/W
1, 0	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W

The Global Test Control Register is used to control the global test modes of the RS-CANFD lite module.

- **RTME**
  If this bit is set, RS-CANFD lite module is configured in RAM test mode.
  Refer to **19.11.2.1 RAM Test Mode** for details.
  Write this bit only when RS-CANFD lite module is in GL_HALT mode.
  This bit is cleared automatically when RS-CANFD lite module enters GL_RESET mode.

19.3.50 Global FD Configuration Register (GFDCFGs) [s = 0, 1]

Address: GFDCFGs: F03B0H
GFDCFGsL: F03B0H, GFDCFGsH: F03B1H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GFDCFGs	–	–	–	–	–	–	TSCCFG[1:0]		–	–	–	–	–	–	–	RPED
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 read as 0. The write value should be always 0.	R/W
9, 8	TSCCFG[1:0]	Timestamp Capture Configuration	00B: Timestamp capture at the sample point of SOF (start of frame) 01B: Timestamp capture at frame valid indication 10B: Timestamp capture at the sample point of res bit 11B: reserved	R/W
7 to 1	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
0	RPED	Res Bit Protocol Exception Disable	0: Protocol exception event detection enabled 1: Protocol exception event detection disabled	R/W

- **TSCCFG[1:0]**

  These bits configure the different capture points of the timestamp; for transmission and reception. When **GFDCFG.TSCCFG[1:0]** = 10B then the timestamp capture is done for CAN-FD frames at res bit and for Classical-CAN frames at the start of frame.
  Write these bits only when RS-CANFD lite module is in GL_RESET mode.

- **RPED**

  This bit configures the protocol exception event handling according to ISO 11898-1. When this bit is enabled then the protocol exception event detection is disabled, and the protocol controller will transmit an error frame when the protocol exception event is detected (res bit is sampled recessive).
  Write this bit only when RS-CANFD lite module is in GL_RESET mode.

19.3.51 Global Lock Key Register (GLOCKKs) [s = 0, 1]

Address: GLOCKKs: F03B8H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GLOCKKs	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]	Lock Key	Key bits for unlocking the protection of test modes	W

The Global Lock Key Register is a write only register that is used to unlock the protection for special test bits.
Refer to **19.11.2 Global Test Modes** for Lock Key specification.

- **LOCK[15:0]**
  The Lock Key sequence must be written in these bits to configure RS-CANFD lite module in RAM test.
  Read value from these bits is always 0000H.
  These bits cannot be written when RS-CANFD lite module is in GL_SLEEP or GL_RESET mode.
  Do not write to these bits when RS-CANFD lite module is in GL_OPERATION mode.

19.3.52 RAM Test Page Access Register r (RPGACCrHs, RPGACCrLs) [r = 0 to 63] [s = 0, 1]

Address: RPGACCrLs: F0580H + (4H×r), RPGACCrHs: F0582H + (4H×r)
RPGACCrLsL: F0580H + (4H×r), RPGACCrLsH: F0581H + (4H×r),
RPGACCrHsL: F0582H + (4H×r), RPGACCrHsH: F0583H + (4H×r)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RPGACCrHs	RDTA[31:16]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RPGACCrLs	RDTA[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
31 to 0	RDTA[31:0]	RAM Data Test Access	RAM Data Bytes	R/W

- **RDTA[31:0]**

  Data can be read from or written into these register bits when RS-CANFD lite module is configured in RAM test mode.
  Write these bits only when RS-CANFD lite module is in GL_HALT mode, and RAM test mode is enabled.
  If data is read when RAM test mode is not enabled, then it is always read as 00000000H.
  SW data should be read/written in the RAM Test Page Access Registers during RAM test mode.

19.3.53 Global Pretended Network Filter List Entry Control Register (GPFLECTRs) [s = 0, 1]

Address: GPFLECTRs: F03D0H
GPFLECTRsH: F03D1H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GPFLECTRs	–	–	–	–	–	–	–	PFL DAE	–	–	–	–	–	–	–	–
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 9	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
8	PFLDAE	Pretended Network Filter List Data Access Enable	0: Pretended Network Filter List Data access disabled 1: Pretended Network Filter List Data access enabled	R/W
7 to 0	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W

- **PFLDAE**

  This bit prevents pretended network filter list write access if cleared after configuration of the Pretended Network Filter List.
  Pretended Network Filter List can read data regardless of the status of this bit.
  This bit cannot be written when RS-CANFD lite module is in GL_SLEEP mode.
  This bit should be set to enable write access to Pretended Network Filter List.

19.3.54 Global Pretended Network Filter List Configuration Register (GPFLCFGs) [s = 0, 1]

Address: GPFLCFGs: F03D6H
GPFLCFGsH: F03D7H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GPFLCFGs	–	–	–	–	–	–	RNC[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 10	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
9, 8	RNC[1,0]	Rule Number	Number of rules dedicated	R/W
7 to 0	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W

This register is used to define the number of Pretended Network Filter List entries (called “rules”) applicable in the Pretended Network Filter List.

The maximum number of Pretended Network Filter is 2.

- **RNC[1:0]**

  These bits define the number of rules in the Pretended Network Filter List.

  Write these bits only when RS-CANFD lite module is in GL_RESET mode.

19.3.55 Global Pretended Network Filter List ID Register j (GPFLIDjHs, GPFLIDjLs) [j = 0, 1] [s = 0, 1]

Address: GPFLIDjLs: F0520H + (24H×j), GPFLIDjHs: F0522H + (24H×j)
GPFLIDjLsL: F0520H + (24H×j), GPFLIDjLsH: F0521H + (24H×j),
GPFLIDjHsL: F0522H + (24H×j), GPFLIDjHsH: F0523H + (24H×j)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GPFLIDjHs	GPFLIDE	GPFLRTR	GPFLLB	GPFLID[28:16]												
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GPFLIDjLs	GPFLID[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
31	GPFLIDE	Global Pretended Network Filter List IDE Field	0: Standard Identifier of Rule entry ID is valid for acceptance filtering 1: Extended Identifier of Rule entry ID is valid for acceptance filtering	R/W
30	GPFLRTR	Global Pretended Network Filter List Entry RTR Field	0: Data Frame 1: Remote Frame	R/W
29	GPFLLB	Global Pretended Network Filter List Entry Loopback Configuration	0: Global Pretended Network Filter List entry ID for acceptance filtering has attribute ‘RX’ 1: Global Pretended Network Filter List entry ID for acceptance filtering has attribute ‘TX’	R/W
28 to 0	GPFLID[28:0]	Global Pretended Network Filter List ID Field	ID part of the Global Pretended Network Filter List entry	R/W

These registers are used to configure the ID field of the rule entries in the Global Pretended Network Filter List.

- **GPFLIDE**

  This bit allows the configuration of the ID format (standard ID or extended ID) for each of the Global Pretended Network Filter List entry. For each rule entry of the related CAN channel the Pretended Network filter process compares this bit against the IDE bit of the received CAN message.
  This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GPFLRTR**

  This bit allows the configuration of the specified frame format (data frame or remote frame) for each Global Pretended Network Filter List entry.
  For each rule entry in the CAN channel, the Pretended Network Filter process compares this bit with the RTR bit in the received CAN message.
  This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GPFLLB**

  This bit selects if the Global Pretended Network Filter List entry gets the attribute ‘RX’ or ‘TX’. This attribute decides about the validity of the entry in the mirror mode case, loopback test mode case and during standard (non-loopback) reception. See **Table 19-32** for detailed description of the validity of the Global Pretended Network Filter List entry depending on transmitter/receiver case, type of loopback mode and RX/TX attribute.
  This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  When using Global Pretended Network Filter List Entry loopback, set this bit to 1.

- **GPFLID[28:0]**

  These bits represent the CAN identifier (ID) field of each of the Global Pretended Network Filter List entry. Pretended Network filter process compares this field against the ID of a received CAN message.
  For alignment of these bits in standard and extended frame format, see **19.4 Identifier Bits Alignment**.
  These bits cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.56 Global Pretended Network Filter List Mask Register j (GPFLMjHs, GPFLMjLs) [j = 0, 1] [s = 0, 1]

Address: GPFLMjLs: F0524H + (24H×j) GPFLMjHs: F0526H + (24H×j)
GPFLMjLsL: F0524H + (24H×j), GPFLMjLsH: F0525H + (24H×j),
GPFLMjHsL: F0526H + (24H×j), GPFLMjHsH: F0527H + (24H×j)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GPFLMjHs	GPFLIDEM	GPFLRTRM	GPFLIFL1	GPFLIDM[28:16]												
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GPFLMjLs	GPFLIDM[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
31	GPFLIDEM	Global Pretended Network Filter List IDE Mask	0: IDE bit is not considered for ID matching 1: IDE bit is considered for ID matching	R/W
30	GPFLRTRM	Global Pretended Network Filter List Entry RTR Mask	0: RTR bit is not considered for ID matching 1: RTR bit is considered for ID matching	R/W
29	GPFLIFL1	Global Pretended Network Filter List Information Label 1	Global Pretended Network Filter List Information Label bit1	R/W
28 to 0	GPFLIDM[28:0]	Global Pretended Network Filter List ID Mask Field	0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.	R/W

These registers are used to configure the mask field of each rule entries in the Global Pretended Network Filter List.

- **GPFLIDEM**

  This bit allows the configuration of the IDE mask bit for each Global Pretended Network Filter List entry.
  When IDE mask bit is 0, then the ID comparison depends upon the received IDE bit.
  If received IDE bit is 0, then STD-ID comparison takes place.
  If received IDE bit is 1, then EXT-ID comparison takes place.
  This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GPFLRTRM**

  This bit allows the configuration of the RTR mask bit for each Global Pretended Network Filter List entry.
  This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GPFLIFL1**

  This bit allows the configuration of a 2-bit information label that will be attached to a received message accepted by the related Global Pretended Network Filter List entry.
  This is MSB bit of an information label.

This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
This bit is stored in Information Label Field [1] (**RMSTSn.RMIFL[1]**, **RFFDSTSk.RFIFL[1]**, or **CFFDCSTS.CFIFL[1]**) of the storage location of an incoming message.

- **GPFLIDM[28:0]**
  These bits are the filter mask bits for the related bits in the CAN identifier field of each Global Pretended Network Filter List entry.
  These bits cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.57 Global Pretended Network Filter List Pointer 0 Register j (GPFLP0jHs, GPFLP0jLs) [j = 0, 1] [s = 0, 1]

Address: GPFLP0jLs: F0528H + (24H×j), GPFLP0jHs: F052AH + (24H×j)
GPFLP0jLsL: F0528H + (24H×j), GPFLP0jLsH: F0529H + (24H×j),
GPFLP0jHsL: F052AH +(24H×j), GPFLP0jHsH: F052BH + (24H×j)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GPFLP0jHs	GPFLPTR[15:0]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GPFLP0jLs	GPFLRMV	–	–	GPFLRMDP[4:0]					GPFLIFL0	–	–	–	GPFLDLC[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
31 to 16	GPFLPTR[15:0]	Global Pretended Network Filter List Pointer Field	Global Pretended Network Filter List Pointer	R/W
15	GPFLRMV	Global Pretended Network Filter List RX Message Buffer Valid	0: Global Pretended Network Filter List Single Message Buffer Direction Pointer is invalid 1: Global Pretended Network Filter List Single Message Buffer Direction Pointer is valid	R/W
14, 13	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
12 to 8	GPFLRMDP[4:0]	Global Pretended Network Filter List RX Message Buffer Direction Pointer	RX Message Buffer number for storage of received messages	R/W
7	GPFLIFL0	Global Pretended Network Filter List Information Label 0	Global Pretended Network Filter List Information Label bit0	R/W
6 to 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3 to 0	GPFLDLC[3:0]	Global Pretended Network Filter List DLC Field	Minimum no. of Data Bytes in a Data Frame required for its acceptance	R/W

These registers are used to configure the DLC, SW pointer, single message buffer select and message buffer direction pointer for each rule entry in the Global Pretended Network Filter List.

- **GPFLPTR[15:0]**

  These bits allow the configuration of a 16-bit pointer that will be attached to a received message accepted by the related Global Pretended Network Filter List entry.

  The Pointer will be added during message storage in the Message Buffer area and can be used by the application as support function.

  The pointer information could be used for example to support PDU Identifier allocation for the received message in AUTOSAR systems.

  These bits cannot be written when **GPFLECTR.PFLDAE** bit is 0.

  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GPFLRMV**

  This bit allows the enabling/disabling of a single reception Message Buffer as the target for a received message that is passing the acceptance check of the related Global Pretended Network Filter List entry.
  This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GPFLRMDP[4:0]**

  These bits allow the configuration of a single reception Message Buffer as the destination target for a received message that is passing the acceptance check of the related Global Pretended Network Filter List entry. The value entered is the single destination Message Buffer number.
  These bits cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  RMNB.NRXMB[4:0] bits are the value entered in the RX Message Buffer Number Register to configure the number of RX Message Buffers.
  The value to be entered in **GPFLP0j.GPFLRMDP[4:0]** bits should only be between 00000B and "(**RMNB.NRXMB[4:0]** value) − 1". If **RMNB.NRXMB[4:0]** = 00000B, then the **GPFLP0j.GPFLRMV** bit should be configured as 0.

- **GPFLIFL0**

  This bit allows the configuration of a 2-bit Information label that will be attached to a received message accepted by the related Global Pretended Network Filter List entry.
  This bit is a LSB bit of an information label.
  This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  This bit is stored in Information Label Field [0] (**RMSTSn.RMIFL[0]**, **RFFDSTSk.RFIFL[0]**, or **CFFDCSTS.CFIFL[0]**) of the storage location of an incoming message.

- **GPFLDLC[3:0]**

  These bits allow the configuration of the minimum DLC (Data Length Code) value for a message to be accepted by the related Global Pretended Network Filter List entry (automatic DLC filter function). DLC filter process is only passed if the DLC value of the message accepted by a Global Pretended Network Filter List entry is equal or higher than the DLC value configured for this related Global Pretended Network Filter List entry.
  Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0H.

Following binary values can be configured:

Format	GPFLDLC[3]	GPFLDLC[2]	GPFLDLC[1]	GPFLDLC[0]	Description
CAN and CAN-FD	0	0	0	0	DLC of received message = 0 or more (DLC Filter check is disabled)
CAN and CAN-FD	0	0	0	1	DLC of received message = 1 or more
CAN and CAN-FD	0	0	1	0	DLC of received message = 2 or more
CAN and CAN-FD	0	0	1	1	DLC of received message = 3 or more
CAN and CAN-FD	0	1	0	0	DLC of received message = 4 or more
CAN and CAN-FD	0	1	0	1	DLC of received message = 5 or more
CAN and CAN-FD	0	1	1	0	DLC of received message = 6 or more
CAN and CAN-FD	0	1	1	1	DLC of received message = 7 or more
CAN	1	x	x	x	DLC of received message = 8 or more
CAN-FD	1	0	0	0	DLC of received message = 8 or more
CAN-FD	1	0	0	1	DLC of received message = 12 or more
CAN-FD	1	0	1	0	DLC of received message = 16 or more
CAN-FD	1	0	1	1	DLC of received message = 20 or more
CAN-FD	1	1	0	0	DLC of received message = 24 or more
CAN-FD	1	1	0	1	DLC of received message = 32 or more
CAN-FD	1	1	1	0	DLC of received message = 48 or more
CAN-FD	1	1	1	1	DLC of received message = 64

These bits cannot be written when **GPFLECTR.PFLDAE** bit is 0.

Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.58 Global Pretended Network Filter List Pointer 1 Register j (GPFLP1jLs) [j = 0, 1] [s = 0, 1]

Address: GPFLP1jLs: F052CH + (24H×j)
GPFLP1jLsL: F052CH + (24H×j), GPFLP1jLsH: F052DH + (24H×j)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GPFLP1jLs	–	–	–	–	–	–	–	GPFLFDP[8]	–	–	–	–	–	–	GPFLFDP[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 9	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
8	GPFLFDP[8]	Global Pretended Network Filter List FIFO Direction Pointer (GPFLFDP[8])	FIFO direction pointer bits for received message storage	R/W
7 to 2	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
1, 0	GPFLFDP[1:0]	Global Pretended Network Filter List FIFO Direction Pointer (GPFLFDP[1:0])	FIFO direction pointer bits for received message storage	R/W

These registers are used to configure the FIFO direction pointer fields in each rule entry of the Global Pretended Network Filter List.

- **GPFLFDP[8], [1:0]**

  Global Pretended Network Filter List FIFO Direction Pointer (**GPFLFDP[8], [1:0]**)
  These bits allow the configuration of FIFO Buffers as the target for a received message passing the acceptance check of the related Global Pretended Network Filter List entry.
  Each bit of the **GPFLP1j.GPFLFDP[8]**, **[1:0]** is configuring a dedicated FIFO:

Bit	Symbol	Value (Binary)	Function
0	GPFLFDP[0]	0	Disable RX FIFO 0 as target for reception
		1	Enable RX FIFO 0 as target for reception
1	GPFLFDP[1]	0	Disable RX FIFO 1 as target for reception
		1	Enable RX FIFO 1 as target for reception
8	GPFLFDP[8]	0	Disable Common FIFO as target for reception
		1	Enable Common FIFO as target for reception

  These bits cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  For storage in Common FIFO, target for reception can only be those Common FIFO buffers that are configured as RX FIFO.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.
  These bits should be up to 2 destination FIFO buffers or 1 destination FIFO buffers plus one RX Message Buffer.

19.3.59 Global Pretended Network Filter List Payload Type Register j (GPFLPTjHs, GPFLPTjLs) [j = 0, 1] [s = 0, 1]

Address: GPFLPTjLs: F0530H + (24H×j), GPFLPTjHs: F0532H + (24H×j)
GPFLPTjLsL: F0530H + (24H×j), GPFLPTjLsH: F0531H + (24H×j),
GPFLPTjHsL: F0532H + (24H×j), GPFLPTjHsH: F0533H + (24H×j)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GPFLPTjHs	GPFLANDOR	GPFLRANG0	GPFLOUT0	–	–	–	–	–	–	–	–	–	GPFLOFFSET0[3:0]			
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GPFLPTjLs	–	GPFLRANG1	GPFLOUT1	–	–	–	–	–	–	–	–	–	GPFLOFFSET1[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
31	GPFLANDOR	Global Pretended Network filter conditions of the filters 0 and 1	0: Both of filters 0 and 1 are successful. 1: One of the filter 0 or 1 is successful.	R/W
30	GPFLRANG0	Global Pretended Network filter comparison conditions of the filter0	0: payload data match filter 1: upper / lower filter	R/W
29	GPFLOUT0	Global Pretended Network filter conditions of upper / lower filter of the filter0	0: Within the range of upper limit and lower limit 1: Outside of the range of upper limit and lower limit	R/W
28 to 20	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
19 to 16	GPFLOFFSET0[3:0]	Global Pretended Network filter offset value of the filter0	offset value of the filter0 is specified.	R/W
15	–	Reserved	This bit is read as 0. The write value should be always 0.	R/W
14	GPFLRANG1	Global Pretended Network filter comparison conditions of the filter1	0: payload data match filter 1: upper / lower filter	R/W
13	GPFLOUT1	Global Pretended Network filter conditions of upper / lower filter of the filter1	0: Within the range of upper limit and lower limit 1: Outside of the range of upper limit and lower limit	R/W
12 to 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3 to 0	GPFLOFFSET1[3:0]	Global Pretended Network filter offset value of the filter1	offset value of the filter1 is specified.	R/W

- **GPFLANDOR**

  This bit can set up the comparison conditions of two filters.
  This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GPFLRANG0**

  The comparison conditions of the filter0.
  This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GPFLOUT0**

  The conditions of upper / lower filter of the filter0.
  This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GPFLOFFSET0[3:0]**

  The offset value of the filter0 of Global Pretended Network filter is specified.
  Offset can be specified per 4 bytes.
  These bits cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GPFLRANG1**

  The comparison conditions of the filter1.
  This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GPFLOUT1**

  The conditions of upper / lower filter of the filter1.
  This bit cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

- **GPFLOFFSET1[3:0]**

  The offset value of the filter1 of Global Pretended Network filter is specified.
  Offset can be specified per 4 bytes.
  These bits cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write this bit only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.60 Global Pretended Network Filter List Payload Data 0 Register j (GPFLPD0jHs, GPFLPD0jLs) [j = 0, 1] [s = 0, 1]

Address: GPFLPD0jLs: F0534H + (24H×j), GPFLPD0jHs: F0536H + (24H×j)
GPFLPD0jLsL: F0534H + (24H×j), GPFLPD0jLsH: F0535H + (24H×j),
GPFLPD0jHsL: F0536H + (24H×j), GPFLPD0jHsH: F0537H + (24H×j)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GPFLPD0jHs	FDATA0[31:16]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GPFLPD0jLs	FDATA0[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
31 to 0	FDATA0[31:0]	Pretended Network Filter List Filter data	Comparison data for payload filters	R/W

- **FDATA0[31:0]**

  These bits set the payload filter data of the Pretended Network Filter List.
  When **GPFLPTj.GPFLRANG0** is 0, these bits set up the payload match data of the filter 0.
  When **GPFLPTj.GPFLRANG0** is 1, these bits set up the upper-limit filter value of the filter 0.
  These bits cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.61 Global Pretended Network Filter List Payload Data 1 Register j (GPFLPD1jHs, GPFLPD1jLs) [j = 0, 1] [s = 0, 1]

Address: GPFLPD1jLs: F053CH + (24H×j), GPFLPD1jHs: F053EH + (24H×j)
GPFLPD1jLsL: F053CH + (24H×j), GPFLPD1jLsH: F053DH + (24H×j),
GPFLPD1jHsL: F053EH + (24H×j), GPFLPD1jHsH: F053FH + (24H×j)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GPFLPD1jHs	FDATA1[31:16]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GPFLPD1jLs	FDATA1[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
31 to 0	FDATA1[31:0]	Pretended Network Filter List Filter Data	Comparison data for payload filters	R/W

- **FDATA1[31:0]**

  These bits set the payload filter data of the Pretended Network Filter List.
  When **GPFLPTj.GPFLRANG1** is 0, these bits set up the payload match data of the filter 1.
  When **GPFLPTj.GPFLRANG1** is 1, these bits set up the upper-limit filter value of the filter 1.
  These bits cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.62 Global Pretended Network Filter List Payload Mask 0 Register j (GPFLPM0jHs, GPFLPM0jLs) [j= 0, 1] [s = 0, 1]

Address: GPFLPM0jLs: F0538H + (24H×j), GPFLPM0jHs: F053AH + (24H×j)
GPFLPM0jLsL: F0538H + (24H×j), GPFLPM0jLsH: F0539H + (24H×j),
GPFLPM0jHsL: F053AH + (24H×j), GPFLPM0jHsH: F053BH + (24H×j)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GPFLPM0jHs	FMASK0[31:24]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GPFLPM0jLs	FMASK0[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
31 to 0	FMASK0[31:0]	Pretended Network Filter List Filter Data Mask Field	Global Pretended Network Filter List Mask field bits for payload data field bits	R/W

- **FMASK0[31:0]**

  These bits set the payload filter data mask of the Pretended Network Filter List.
  When **GPFLPTj.GPFLRANG0** is 0, these bits set up the payload data mask of the filter0.
  When **GPFLPTj.GPFLRANG0** is 1, these bits set up the lower-limit filter value of the filter0.
  These bits cannot be written when **GPFLECTRs.PFLDAE** bit is 0.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.63 Global Pretended Network Filter List Payload Mask 1 Register j (GPFLPM1jHs, GPFLPM1jLs) [j = 0, 1] [s = 0, 1]

Address: GPFLPM1jLs: F0540H + (24H×j), GPFLPM1jHs: F0542H + (24H×j)
GPFLPM1jLsL: F0540H + (24H×j), GPFLPM1jLsH: F0541H + (24H×j),
GPFLPM1jHsL: F0542H + (24H×j), GPFLPM1jHsH: F0543H + (24H×j)

Condition: **CFDGRWCs.RPAGE[1:0]** = 00B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
GPFLPM1jHs	FMASK1[31:16]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GPFLPM1jLs	FMASK1[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
31 to 0	FMASK1[31:0]	Pretended Network Filter List Filter Data Mask Field	Global Pretended Network Filter List Mask field bits for payload data field bits	R/W

- **FMASK1[31:0]**

  These bits set the payload filter data mask of the Pretended Network Filter List.
  When **GPFLPTj.GPFLRANG1** is 0, these bits set up the payload data mask of the filter1.
  When **GPFLPTj.GPFLRANG1** is 1, these bits set up the lower-limit filter value of the filter1.
  These bits cannot be written when **GPFLECTR.PFLDAE** bit is 0.
  Write these bits only when the related RS-CANFD lite channel is in CH_RESET or CH_HALT mode.

19.3.64 Global AFL Ignore Entry Register (GAFLIGENTs) [s = 0, 1]

Address: GAFLIGENTs: F03C0H
GAFLIGENTsL: F03C0H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GAFLIGENTs	–	–	–	–	–	–	–	–	–	–	–	–	IRN[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 4	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
3 to 0	IRN[3:0]	Ignore Rule Number	Define rule number which ignores an AFL entry	R/W

- **IRN[3:0]**
  These bits define the rule number which updates an AFL entry.
  Up to 16 rules can be set for AFL entry.
  Write these bits only when **GAFLIGCTR.IREN** bit is 0.
  These bits cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.

19.3.65 Global AFL Ignore Control Register (GAFLIGCTRs) [s = 0, 1]

Address: GAFLIGCTRs: F03C4H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GAFLIGCTRs	KEY[7:0]								–	–	–	–	–	–	–	IREN
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	KEY[7:0]	Key Code	These bits control the right or wrong of rewriting of the IREN bit.	W
7 to 1	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
0	IREN	Ignore Rule Enable	0: AFL entry number does not ignore 1: AFL entry number ignores	R/W

- **KEY[7:0]**

  Writing C4H to these bits allows the **GAFLIGCTR.IREN** bit to be written.
  Read value from these bits are always 00H.
  Be sure to write the **GAFLIGCTR.IREN** bit and the **GAFLIGCTR.KEY[7:0]** bits simultaneously.

- **IREN**

  When this bit is set, the entry number (selected by **GAFLIGENT.IRN[3:0]** bits) is ignored.
  This bit is cleared automatically when RS-CANFD lite module enters GL_RESET mode.
  Be sure to write the **GAFLIGCTR.IREN** bit and the **GAFLIGCTR.KEY[7:0]** bits simultaneously.

19.3.66 RX Message Buffer Interrupt Enable Configuration Register (RMIECs) [s = 0, 1]

Address: RMIECs: F0338H
RMIECsL: F0338H, RMIECsH: F0339H

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RMIECs	RMIE[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	RMIE[15:0]	RX Message Buffer Interrupt Enable	0: RX Message Buffer Interrupt disabled for corresponding RX message buffer 1: RX Message Buffer Interrupt enabled for corresponding RX Message Buffer	R/W

These bits show the RX Message Buffer interrupt enable for the corresponding RX Message Buffer. The bit position of **RMIEC** register corresponds to the buffer number of RX Message Buffer.

e.g. The bit 0 of the **RMIEC** register corresponds to the RX Message Buffer 0.

- **RMIE[15:0]**
  If this bit is set, then an interrupt will be generated at the end of a successful reception from the corresponding message buffer.
  Please refer to **19.12 Interrupt** for RX Message Buffer interrupt specification.
  These bits cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.

19.3.67 RS-CANFD lite Global RAM Windows Control Register (CFDGRWCs) [s = 0, 1]

Address: CFDGRWCs: F03DCH

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CFDGRWCs	KEY[7:0]								–	–	–	–	–	–	RPAGE[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	KEY[7:0]	Key Code	These bits control the right or wrong of rewriting of RPAGE bits.	W
7 to 2	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
1, 0	RPAGE[1:0]	RAM Page Select	00B : Page 0 selected 01B : Page 1 selected 10B : Page 2 selected 11B : Page 3 selected	R/W

The Global RAM Page Control Register controls the selection of the address space.

- **KEY[7:0]**

  Writing C4H to these bits allows the **CFDGRWC.RPAGE[1:0]** bits to be written.
  Read value from these bits is always 00H.
  Be sure to write the **CFDGRWC.RPAGE[1:0]** bits and the **CFDGRWC.KEY[7:0]** bits simultaneously.

- **RPAGE[1:0]**

  These bits used to select the RAM page. The address range from address 0420H to 067FH is multiplexed in to pages.

  Page 0: Global Acceptance Filter, Pretended Network Filter, and RAM Test Page Access Registers
  Page 1: FIFO Access Register, TX MB Access Register, and the TX History List Access Register
  Page 2: RX MB Access Register 0 to 7.
  Page 3: RX MB Access Register 8 to 15.

  These bits cannot be written when the RS-CANFD lite module is in GL_SLEEP mode.
  Be sure to write the **CFDGRWC.RPAGE[1:0]** bits and the **CFDGRWC.KEY[7:0]** bits simultaneously.

19.3.68 CANFD Window Access Control Register (CFDWINR)

Address: CFDWINR: F01F7H

Symbol	b7	b6	b5	b4	b3	b2	b1	b0
CFDWINR	–	–	–	–	–	–	–	CSEL0
After Reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
7 to 1	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
0	CSEL0	Channel Select	CAN Channel selection. 0: CAN0 selected 1: CAN1 selected	R/W

The CANFD Window Access Control Register selects the channel of CAN to be accessed in the SFR space.

- **CSEL0**

  When **CSEL0** is 0, the CAN0 register group is mapped onto the SFR space.
  When **CSEL0** is 1, the CAN1 register group is mapped onto the SFR space.

19.4 Identifier Bits Alignment

Standard Identifier (11 bit) format: ID28 – ID18 is aligned to b10 – b0.
Extended Identifier (29 bit) format: ID28 – ID0 is aligned to b28 – b0.
For Standard Identifier format bits 11 to 28 (b11 – b28) should be 0.

Figure 19-4. Bit Alignment for Standard and Extended Identifier Format

Standard Identifier (11 bit) Format

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE=0	RTR	–	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0	0	0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18

Extended Identifier (29 bit) Format

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE=1	RTR	–	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

19.5 Message Buffer Component Structure

The message buffer configuration consists of the following four types of message buffer components.

(1) RX Message Buffer Component (**RMBCP**)
(2) RX FIFO Access Message Buffer Component (**RFMBCP**)
(3) Common FIFO Access Message Buffer Component (**CFMBCP**)
(4) TX Message Buffer Component (**TMBCP**).

Table 19-8. Message Buffer Component Configuration

MB Component	MB Component Structure	Register Name (p = 0 to 15)
RMBCPn (n = 0 to 15)	RMBCP0	RMID0
		RMPTR0
		RMSTS0
		RMD0_p
	:	:
	RMBCP15	RMID15
		RMPTR15
		RMSTS15
		RMD15_p
RFMBCPk (k = 0, 1)	RFMBCP0	RFID0
		RFPTR0
		RFFDSTS0
		RFDF0_p
	RFMBCP1	RFID1
		RFPTR1
		RFFDSTS1
		RFDF1_p
CFMBCP	CFMBCP	CFID
		CFPTR
		CFFDCSTS
		CFDFp
TMBCPm (m = 0 to 3)	TMBCP0	TMID0
		TMPTR0
		TMFDCTR0
		TMDF0_p
	:	:
	TMBCP3	TMID3
		TMPTR3
		TMFDCTR3
		TMDF3_p

19.5.1 RX Message Buffer ID Register n (RMIDnHs, RMIDnLs) [n = 0 to 15] [s = 0, 1]

Address: RMIDnLs: F0420H + (4CH×n), RMIDnHs: F0422H + (4CH×n)
RMIDnLsL: F0420H + (4CH×n), RMIDnLsH: F0421H + (4CH×n),
RMIDnHsL: F0422H + (4CH×n), RMIDnHsH: F0423H + (4CH×n)

Conditions: n = 0 to 7, **CFDGRWCs.RPAGE[1:0]** = 10B

Address: RMIDnLs: F0420H + (4CH×(n-8)), RMIDnHs: F0422H + (4CH×(n-8))
RMIDnLsL: F0420H + (4CH×(n-8)), RMIDnLsH: F0421H + (4CH×(n-8)),
RMIDnHsL: F0422H + (4CH×(n-8)), RMIDnHsH: F0423H + (4CH×(n-8))

Conditions: n = 8 to 15, **CFDGRWCs.RPAGE[1:0]** = 11B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RMIDnHs	RM IDE	RM RTR	–	RMID[28:16]												
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RMIDnLs	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
31	RMIDE	RX Message Buffer IDE Bit	0: STD-ID is stored 1: EXT-ID is stored	R
30	RMRTR	RX Message Buffer RTR Bit	0: Data Frame 1: Remote Frame	R
29	–	Reserved	This bit is read as 0.	R
28 to 0	RMID[28:0]	RX Message Buffer ID Field	STD-ID / EXT-ID fields	R

The RX Message Buffer ID Register stores the ID field, IDE bit and RTR bit of the received message.

- **RMIDE**

  This bit shows whether message with standard identifier or extended identifier was stored in the RX Message Buffer.

- **RMRTR**

  This bit shows whether a Data Frame or a Remote Frame was stored in the RX Message Buffer.

  Caution There are no remote frames in CAN-FD format. In case a CAN-FD frame was received the register reflects the state of the received value (RRS bit in FD frame format).

- **RMID[28:0]**

  These are the bits of the STD-ID / EXT-ID fields of the message stored in the RX Message Buffer.
  For alignment of these bits in standard and extended frame format, see **19.4 Identifier Bits Alignment**.

19.5.2 RX Message Buffer Pointer Register n (RMPTRnHs, RMPTRnLs) [n = 0 to 15] [s = 0, 1]

Address: RMPTRnLs: F0424H + (4CH×n), RMPTRnHs: F0426H + (4CH×n)
RMPTRnLsL: F0424H + (4CH×n), RMPTRnLsH: F0425H + (4CH×n),
RMPTRnHsL: F0426H + (4CH×n), RMPTRnHsH: F0427H + (4CH×n)

Conditions: n = 0 to 7, **CFDGRWCs.RPAGE[1:0]** = 10B

Address: RMPTRnLs: F0424H + (4CH×(n-8)), RMPTRnHs: F0426H + (4CH×(n-8))
RMPTRnLsL: F0424H + (4CH×(n-8)), RMPTRnLsH: F0425H + (4CH×(n-8)),
RMPTRnHsH: F0427H + (4CH×(n-8))

Conditions: n = 8 to 15, **CFDGRWCs.RPAGE[1:0]** = 11B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RMPTRnHs	RMDLC[3:0]				–	–	–	–	–	–	–	–	–	–	–	–
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RMPTRnLs	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
31 to 28	RMDLC[3:0]	RX Message Buffer DLC Field	No. of Data Bytes received in a CAN Frame	R
27 to 16	–	Reserved	These bits are read as 0.	R
15 to 0	RMTS[15:0]	RX Message Buffer Timestamp Field	Timestamp value stored for the message stored in the RX Message Buffer	R

The RX Message Buffer Pointer Register stores the DLC and timestamp fields for the received message.

- **RMDLC[3:0]**

  The number of Data Bytes that were received in the RX Message Buffer is stored in these bits.

  Caution The maximum capacity of the buffer belongs to the RMNB.RMPLS[2:0] (reception message buffer payload data size)**.**

- **RMTS[15:0]**

  Indicates the timestamp value of the message stored in the RX Message Buffer. The timestamp value is obtained at the timing selected by **GFDCFG.TSCCFG[1:0]**.

19.5.3 RX Message Buffer CAN-FD Status Register n (RMSTSnHs, RMSTSnLs) [n = 0 to 15] [s = 0, 1]

Address: RMSTSnLs: F0428H + (4CH×n), RMSTSnHs: F042AH + (4CH×n)
RMSTSnLsL: F0428H + (4CH×n), RMSTSnLsH: F0429H + (4CH×n),
RMSTSnHsL: F042AH + (4CH×n), RMSTSnHsH: F042BH + (4CH×n)

Conditions: n = 0 to 7, **CFDGRWCs.RPAGE[1:0]** = 10B

Address: RMSTSnLs: F0428H + (4CH×(n-8)), RMSTSnHs: F042AH + (4CH×(n-8))
RMSTSnLsL: F0428H + (4CH×(n-8)), RMSTSnLsH: F0429H + (4CH×(n-8)),
RMSTSnHsL: F042AH + (4CH×(n-8)), RMSTSnHsH: F042BH + (4CH×(n-8))

Conditions: n = 8 to 15, **CFDGRWCs.RPAGE[1:0]** = 11B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RMSTSnHs	RMPTR[15:0]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RMSTSnLs	–	–	–	–	–	–	RMIFL[1:0]		–	–	–	–	–	RMFDF	RMBRS	RMESI
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
31 to 16	RMPTR[15:0]	RX Message Buffer Pointer Field	RX Message Buffer Pointer	R
15 to 10	–	Reserved	These bits are read as 0.	R
9, 8	RMIFL[1:0]	RX Message Buffer Information Label Field	RX Message Buffer Information Label	R
7 to 3	–	Reserved	These bits are read as 0.	R
2	RMFDF	CAN-FD Format bit	0: Non CAN-FD frame received 1: CAN-FD frame received	R
1	RMBRS	Bit Rate Switch bit	0: CAN-FD frame received with no bit rate switch 1: CAN-FD frame received with bit rate switch	R
0	RMESI	Error State Indicator bit	0: CAN-FD frame received from error active node 1: CAN-FD frame received from error passive node	R

The RX Message Buffer CAN-FD Status Register shows the status of the FDF, BRS, ESI bits, and pointer of the received CAN-FD frame.

- **RMPTR[15:0]**
  The pointer value from the related Global Acceptance Filter List entry is stored in these bits.

- **RMIFL[1:0]**
  The information label value from the related Global Acceptance Filter List entry is stored in these bits.

- **RMFDF**
  This bit is the same value as the FDF bit of the received CAN-FD frame.

- **RMBRS**

  This bit is the same value as the BRS bit of the received CAN-FD frame.
  When the received FDF bit is 0, means a CAN2.0 frame is received, 0 is always stored to this bit.

- **RMESI**

  This bit is the same value as the ESI bit of the received CAN-FD frame.
  When the received FDF bit is 0, means a CAN2.0 frame is received, 0 is always stored to this bit.

19.5.4 RX Message Buffer Data Field p Register n (RMDn_pHs, RMDn_pLs) [n = 0 to 15] [p = 0 to 15] [s = 0, 1]

Address: RMDn_pLs: F042CH + (4CH×n) + (4×p), RMDn_pHs: F042EH + (4CH×n) + (4×p)
RMDn_pLsL: F042CH + (4CH×n) + (4×p), RMDn_pLsH: F042DH + (4CH×n) + (4×p),
RMDn_pHsL: F042EH + (4CH×n) + (4×p), RMDn_pHsH: F042FH + (4CH×n) + (4×p)

Conditions: n = 0 to 7, **CFDGRWCs.RPAGE[1:0]** = 10B

Address: RMDn_pLs: F042CH + (4CH×(n-8)) + (4×p), RMDn_pHs: F042EH + (4CH×(n-8)) + (4×p)
RMDn_pLsL: F042CH + (4CH×(n-8)) + (4×p), RMDn_pLsH: F042DH + (4CH×(n-8)) + (4×p),
RMDn_pHsL: F042EH + (4CH×(n-8)) + (4×p), RMDn_pHsH: F042FH + (4CH×(n-8)) + (4×p)

Conditions: n = 8 to 15, **CFDGRWCs.RPAGE[1:0]** = 11B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RMDn_pHs	RMDBq[7:0]								RMDBq[7:0]							
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RMDn_pLs	RMDBq[7:0]								RMDBq[7:0]							
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Description	R/W
31 to 24	RMDBq[7:0]	RX Message Buffer Data Byte q	R
23 to 16	RMDBq[7:0]	RX Message Buffer Data Byte q	R
15 to 8	RMDBq[7:0]	RX Message Buffer Data Byte q	R
7 to 0	RMDBq[7:0]	RX Message Buffer Data Byte q	R

RMDn_p	p							
	0	1	2	3	4	5	6	7
RMDn_pLsL	q = 0	q = 4	q = 8	q = 12	q = 16	q = 20	q = 24	q = 28
RMDn_pLsH	q = 1	q = 5	q = 9	q = 13	q = 17	q = 21	q = 25	q = 29
RMDn_pHsL	q = 2	q = 6	q = 10	q = 14	q = 18	q = 22	q = 26	q = 30
RMDn_pHsH	q = 3	q = 7	q = 11	q = 15	q = 19	q = 23	q = 27	q = 31

RMDn_p	p							
	8	9	10	11	12	13	14	15
RMDn_pLsL	q = 32	q = 36	q = 40	q = 44	q = 48	q = 52	q = 56	q = 60
RMDn_pLsH	q = 33	q = 37	q = 41	q = 45	q = 49	q = 53	q = 57	q = 61
RMDn_pHsL	q = 34	q = 38	q = 42	q = 46	q = 50	q = 54	q = 58	q = 62
RMDn_pHsH	q = 35	q = 39	q = 43	q = 47	q = 51	q = 55	q = 59	q = 63

The RX Message Buffer Data Field p Registers store the data bytes RMDBq of the received message.

- **RMDBq[7:0]**

  Data byte q of the Message stored in the RX Message Buffer.
  Unused data bytes will be filled with 00H.

19.5.5 RX FIFO Access ID Register k (RFIDkHs, RFIDkLs) [k = 0, 1] [s = 0, 1]

Address: RFIDkLs: F0420H + (4CH×k), RFIDkHs: F0422H + (4CH×k)
RFIDkLsL: F0420H + (4CH×k), RFIDkLsH: F0421H + (4CH×k),
RFIDkHsL: F0422H + (4CH×k), RFIDkHsH: F0423H + (4CH×k)

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RFIDkHs	RF IDE	RF RTR	–	RFID[28:16]												
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RFIDkLs	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
31	RFIDE	RX FIFO Buffer IDE Bit	0: STD-ID has been received 1: EXT-ID has been received	R
30	RFRTR	RX FIFO Buffer RTR Bit	0: Data Frame 1: Remote Frame	R
29	–	Reserved	This bit is read as 0.	R
28 to 0	RFID[28:0]	RX FIFO Buffer ID Field	STD-ID / EXT-ID fields	R

The RX FIFO Access ID Register stores the ID field, IDE bit and RTR bit of the message.

- **RFIDE**

  This bit shows whether message with standard identifier or extended identifier was received in the FIFO buffer.

- **RFRTR**

  This bit shows whether a data frame or a remote frame was stored in the FIFO buffer.

  Caution There are no remote frames in CAN-FD format. In case a CAN-FD frame was received the register reflects the state of the received value (RRS bit in FD frame format).

- **RFID[28:0]**

  These are the bits of the STD-ID / EXT-ID fields of the message in the FIFO buffer.
  For alignment of these bits in standard and extended frame format, see **19.4 Identifier Bits Alignment**.

19.5.6 RX FIFO Access Pointer Register k (RFPTRkHs, RFPTRkLs) [k = 0, 1] [s = 0, 1]

Address: RFPTRkLs: F0424H + (4CH×k), RFPTRkHs: F0426H + (4CH×k)
RFPTRkLsL: F0424H + (4CH×k), RFPTRkLsH: F0425H + (4CH×k),
RFPTRkHsH: F0427H + (4CH×k)

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RFPTRkHs	RFDLC[3:0]				–	–	–	–	–	–	–	–	–	–	–	–
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RFPTRkLs	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
31 to 28	RFDLC[3:0]	RX FIFO Buffer DLC Field	No. of Data Bytes received in a CAN Frame	R
27 to 16	–	Reserved	These bits are read as 0.	R
15 to 0	RFTS[15:0]	RX FIFO Timestamp Value	Timestamp value of the received CAN Frame	R

The FIFO Access Pointer Register stores the DLC and timestamp fields for the received message.

- **RFDLC[3:0]**
  The number of data bytes that were received in the RX FIFO buffer is stored in these bits.

- **RFTS[15:0]**
  Indicates the timestamp value of the message stored in the RX FIFO buffer. The timestamp value is obtained at the timing selected by **GFDCFG.TSCCFG[1:0]**.

19.5.7 RX FIFO Access CAN-FD Status Register k (RFFDSTSkHs, RFFDSTSkLs) [k = 0, 1] [s = 0, 1]

Address: RFFDSTSkLs: F0428H + (4CH×k), RFFDSTSkHs: F042AH + (4CH×k)
RFFDSTSkLsL: F0428H + (4CH×k), RFFDSTSkLsH: F0429H + (4CH×k),
RFFDSTSkHsL: F042AH + (4CH×k), RFFDSTSkHsH: F042BH + (4CH×k)

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RFFDSTSkHs	RFPTR[15:0]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RFFDSTSkLs	–	–	–	–	–	–	RFIFL[1:0]		–	–	–	–	–	RF FDF	RF BRS	RF ESI
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
31 to 16	RFPTR[15:0]	RX FIFO Buffer Pointer Field	FIFO Buffer Pointer	R
15 to 10	–	Reserved	These bits are read as 0.	R
9, 8	RFIFL[1:0]	RX FIFO Buffer Information label Field	RX FIFO Buffer Information Label	R
7 to 3	–	Reserved	These bits are read as 0.	R
2	RFFDF	CAN-FD Format bit	0: Non CAN-FD frame received 1: CAN-FD frame received	R
1	RFBRS	Bit Rate Switch bit	0: CAN-FD frame received with no bit rate switch 1: CAN-FD frame received with bit rate switch	R
0	RFESI	Error State Indicator bit	0: CAN-FD frame received from error active node 1: CAN-FD frame received from error passive node	R

The RX FIFO Access CAN-FD Status Register shows the status of the FDF, BRS, ESI bits, and pointer of the received CAN-FD frame.

- **RFPTR[15:0]**

  The pointer value from the related Global Acceptance Filter List entry is stored in these bits.

- **RFIFL[1:0]**

  The information label value from the related Global Acceptance Filter List entry is stored in these bits.

- **RFFDF**

  This bit is the same value as the FDF bit of the received CAN-FD frame.

- **RFBRS**

  This bit is the same value as the BRS bit of the received CAN-FD frame.
  When the received FDF bit is 0, means a CAN2.0 frame is received, 0 is always stored to this bit.

- **RFESI**

  This bit is the same value as the ESI bit of the received CAN-FD frame.
  When the received FDF bit is 0, means a CAN2.0 frame is received, 0 is always stored to this bit.

19.5.8 RX FIFO Access Data Field p Register k (RFDFk_pHs, RFDFk_pLs) [k = 0, 1] [p = 0 to 15] [s = 0, 1]

Address: RFDFk_pLs: F042CH + (4CH×k) + (4×p), RFDFk_pHs: F042EH + (4CH×k) + (4×p)
RFDFk_pLsL: F042CH + (4CH×k) + (4×p), RFDFk_pLsH: F042DH + (4CH×k) + (4×p),
RFDFk_pHsL: F042EH + (4CH×k) + (4×p), RFDFk_pHsH: F042FH + (4CH×k) + (4×p)

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RFDFk_pHs	RFDBq[7:0]								RFDBq[7:0]							
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RFDFk_pLs	RFDBq[7:0]								RFDBq[7:0]							
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Description	R/W
31 to 24	RFDBq[7:0]	FIFO Buffer Data Byte q	R
23 to 16	RFDBq[7:0]	FIFO Buffer Data Byte q	R
15 to 8	RFDBq[7:0]	FIFO Buffer Data Byte q	R
7 to 0	RFDBq[7:0]	FIFO Buffer Data Byte q	R

RFDFk_p	**p**							
	0	1	2	3	4	5	6	7
RFDFk_pLsL	q = 0	q = 4	q = 8	q = 12	q = 16	q = 20	q = 24	q = 28
RFDFk_pLsH	q = 1	q = 5	q = 9	q = 13	q = 17	q = 21	q = 25	q = 29
RFDFk_pHsL	q = 2	q = 6	q = 10	q = 14	q = 18	q = 22	q = 26	q = 30
RFDFk_pHsH	q = 3	q = 7	q = 11	q = 15	q = 19	q = 23	q = 27	q = 31

RFDFk_p	p							
	8	9	10	11	12	13	14	15
RFDFk_pLsL	q = 32	q = 36	q = 40	q = 44	q = 48	q = 52	q = 56	q = 60
RFDFk_pLsH	q = 33	q = 37	q = 41	q = 45	q = 49	q = 53	q = 57	q = 61
RFDFk_pHsL	q = 34	q = 38	q = 42	q = 46	q = 50	q = 54	q = 58	q = 62
RFDFk_pHsH	q = 35	q = 39	q = 43	q = 47	q = 51	q = 55	q = 59	q = 63

The RX FIFO Access Data Field Register stores the data bytes of the received message.

- **RFDBq[7:0]**

  Data byte q of the message present in the FIFO buffer.
  Unused data bytes will be filled with 00H.

19.5.9 Common FIFO Access ID Register (CFIDHs, CFIDLs) [s = 0, 1]

Address: CFIDLs: F04B8H, CFIDHs: F04BAH
CFIDLsL: F04B8H, CFIDLsH: F04B9H,
CFIDHsL: F04BAH, CFIDHsH: F04BBH

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CFIDHs	CF IDE	CF RTR	THL EN	CFID[28:16]												
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CFIDLs	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
31	CFIDE	Common FIFO Buffer IDE Bit	0: STD-ID will be transmitted or has been received 1: EXT-ID will be transmitted or has been received	R/W
30	CFRTR	Common FIFO Buffer RTR Bit	0: Data Frame 1: Remote Frame	R/W
29	THLEN	THL Entry enable	TX FIFO Mode: 0: Entry will not be stored in THL after successful TX. 1: Entry will be stored in THL after successful TX. RX FIFO Mode: Reserved, this bit is read as 0	R/W
28 to 0	CFID[28:0]	Common FIFO Buffer ID Field	STD-ID / EXT-ID fields	R/W

The Common FIFO Access ID Register stores the ID field, IDE bit and RTR bit of the message.

In TX mode, this register can read data from FIFO buffers, only for the current entry based on the write pointer value, not for the other entries.

- **CFIDE**

  This bit selects whether a message with EXT-ID or STD-ID will be transmitted from or was received in the FIFO buffer.
  In TX mode, this bit can be read data from FIFO buffers, or be written data to FIFO buffers.
  In RX mode, this bit can only be read data from FIFO buffers.

- **CFRTR**

  This bit selects whether a data frame or a remote frame will be transmitted from or was received in the FIFO buffer. In case of CFCC.CFM is 1 (TX mode), sets the data format of the message to be sent from the Common FIFO buffer.

  Cautions 1. There are no remote frames in CAN-FD format. In case of CAN-FD frame was received (RX mode), the register reflects the state of the received value (RRS bit in FD frame format).
  2. If CFFDCSTS.CFFDF is 1 (CAN-FD frame), a dominant (data frame) is transmitted.

  In TX mode, this bit can be read data from FIFO buffers, or be written data to FIFO buffers.
  In RX mode, this bit can only be read data from FIFO buffers.

- **THLEN**

  This bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.
  In TX mode, this bit can be read data from FIFO buffers, or be written data to FIFO buffers.
  In RX mode, this bit can only be read data from FIFO buffers.

- **CFID[28:0]**

  These are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.
  For alignment of these bits in standard and extended frame format, see **19.4 Identifier Bits Alignment**.
  In TX mode, these bits can be read data from FIFO buffers, or be written data to FIFO buffers.
  In RX mode, these bits can only be read data from FIFO buffers.

19.5.10 Common FIFO Access Pointer Register (CFPTRHs, CFPTRLs) [s = 0, 1]

Address: CFPTRLs: F04BCH, CFPTRHs: F04BEH
CFPTRLsL: F04BCH, CFPTRLsH: F04BDH,
CFPTRHsH: F04BFH

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CFPTRHs	CFDLC[3:0]				–	–	–	–	–	–	–	–	–	–	–	–
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CFPTRLs	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
31 to 28	CFDLC[3:0]	Common FIFO Buffer DLC Field	No. of Data Bytes received in a CAN Frame, or to be transmitted in a CAN Frame	R/W
27 to 16	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
15 to 0	CFTS[15:0]	Common FIFO Timestamp Value	Timestamp value of the received CAN Frame (FIFO in RX Mode)	R/W

The Common FIFO Access Pointer Register stores the DLC and timestamp fields.

In TX mode, this register can read data from the FIFO buffers, only for the current entry based on the write pointer value, not for the other entries.

- **CFDLC[3:0]**

  The number of data bytes that were received in the FIFO buffer or are to be transmitted, is stored in these bits.
  In TX mode, these bits can be read data from FIFO buffers, or be written data to FIFO buffers.
  In RX mode, these bits can only be read data from FIFO buffers.
  These bits cannot read the data of other entries in the FIFO when configured in TX mode.

- **CFTS[15:0]**

  The Timestamp value taken at the capture point as configured by **GFDCFG.TSCCFG[1:0]** of the received message is stored in these bits (if FIFO is configured in RX mode).
  In TX mode, these bits can be read data from FIFO buffers, or be written data to FIFO buffers.
  In RX mode, these bits can only be read data from FIFO buffers.

19.5.11 Common FIFO Access CAN-FD Control/Status Register (CFFDCSTSHs, CFFDCSTSLs) [s = 0, 1]

Address: CFFDCSTSsL: F04C0H, CFFDCSTSHs: F04C2H
CFFDCSTSLsL: F04C0H, CFFDCSTSLsH: F04C1H,
CFFDCSTSHsL: F04C2H, CFFDCSTSHsH: F04C3H

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CFFDCSTSHs	CFPTR[15:0]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CFFDCSTSLs	–	–	–	–	–	–	CFIFL[1:0]		–	–	–	–	–	CF FDF	CF BRS	CF ESI
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
31 to 16	CFPTR[15:0]	Common FIFO Buffer Pointer Field	FIFO Message Buffer Pointer	R/W
15 to 10	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
9, 8	CFIFL[1:0]	COMMON FIFO Buffer Information label Field	COMMON FIFO Buffer Information Label	R/W
7 to 3	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
2	CFFDF	CAN-FD Format bit	0: Non CAN-FD frame received or to transmit 1: CAN-FD frame received or to transmit	R/W
1	CFBRS	Bit Rate Switch bit	0: CAN-FD frame received or to transmit with no bit rate switch 1: CAN-FD frame received or to transmit with bit rate switch	R/W
0	CFESI	Error State Indicator bit	0: CAN-FD frame received from or to transmit by error active node 1: CAN-FD frame received from or to transmit by error passive node	R/W

The Common FIFO Access CAN-FD Status Register shows the status of the FDF, BRS and ESI bits, pointer of the received CAN-FD frame or the CAN-FD frame to transmit.

In TX mode, this register can read data from FIFO buffers, only for the current entry based on the write pointer value, not for the other entries.

- **CFPTR[15:0]**

  If the Common FIFO is configured in TX Mode, the value programmed in **CFFDCSTS.CFPTR[15:0]** will be stored together with further message information, to the TX History List after successful transmission of the message.

  The pointer value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX mode).

  In TX mode, these bits can be read data from FIFO buffers, or be written data to FIFO buffers.

  In RX mode, these bits can only be read data from FIFO buffers.

- **CFIFL[1:0]**

  If the Common FIFO is configured in TX Mode, the value programmed in **CFFDCSTS.CFIFL[1:0]** will be stored together with further message information, to the TX History List after successful transmission of the message.

The Information label value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX mode).
In TX mode, these bits can be read data from FIFO buffers, or be written data to FIFO buffers.
In RX mode, these bits can only be read data from FIFO buffers.

- **CFFDF**

  In TX mode, this bit can be read data from FIFO buffers, or be written data to FIFO buffers.
  In RX mode, this bit can only be read data from FIFO buffers.
  In RX mode, this bit is updated with the FDF bit value of the CAN frame when it has been received, indicating whether it is a CAN 2.0 frame (0) or a CAN-FD frame (1).
  In TX mode, the RS-CANFD lite module will either transmit a 0 to indicate a CAN 2.0 frame is to be transmitted or a 1 to indicate a CAN-FD frame is to be transmitted.

- **CFBRS**

  In TX mode, this bit can be read data from FIFO buffers, or be written data to FIFO buffers.
  In RX mode, this bit can only be read data from FIFO buffers.
  In RX mode, this bit is updated with the BRS bit value of the CAN-FD frame when it has been received, indicating whether there is a bit rate switch (1) or not (0) on the CAN-FD frame.
  In RX mode 0 is stored to this bit when the received FDF bit is 0, that means a CAN 2.0 frame is received.
  In TX mode, the RS-CANFD lite module will either transmit a 0 to indicate no bit rate switch in the frame to be transmitted or a 1 to indicate a bit rate switch in the frame to be transmitted.

- **CFESI**

  In TX mode, this bit can be read data from FIFO buffers, or be written data to FIFO buffers.
  In RX mode, this bit can only be read data from FIFO buffers.
  In RX mode, this bit is updated with the ESI bit value of the CAN-FD frame when it has been received, indicating the error state of the transmitting node.
  In RX mode 0 is stored to this bit when the received FDF bit is 0, that means a CAN 2.0 frame is received.
  In TX mode, if the RS-CANFD lite module is not in error passive, then this bit equals the write value, else it is “don’t care” and the bit is transmitted as 1 on the CAN bus; indicating that this is an error passive node.

19.5.12 Common FIFO Access Data Field p Register (CFDFpHs, CFDFpLs) [p = 0 to 15] [s = 0, 1]

Address: CFDFpLs: F04C4H + (4×p), CFDFpHs: F04C6H + (4×p)
CFDFpLsL: F04C4H + (4×p), CFDFpLsH: F04C5H + (4×p),
CFDFpHsL: F04C6H + (4×p), CFDFpHsH: F04C7H + (4×p)

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CFDFpHs	CFDBq[7:0]								CFDBq[7:0]							
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CFDFpLs	CFDBq[7:0]								CFDBq[7:0]							
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Description	R/W
31 to 24	CFDBq[7:0]	FIFO Buffer Data Byte q	R/W
23 to 16	CFDBq[7:0]	FIFO Buffer Data Byte q	R/W
15 to 8	CFDBq[7:0]	FIFO Buffer Data Byte q	R/W
7 to 0	CFDBq[7:0]	FIFO Buffer Data Byte q	R/W

CFDFp	p							
	0	1	2	3	4	5	6	7
CFDFpLsL	q = 0	q = 4	q = 8	q = 12	q = 16	q = 20	q = 24	q = 28
CFDFpLsH	q = 1	q = 5	q = 9	q = 13	q = 17	q = 21	q = 25	q = 29
CFDFpHsL	q = 2	q = 6	q = 10	q = 14	q = 18	q = 22	q = 26	q = 30
CFDFpHsH	q = 3	q = 7	q = 11	q = 15	q = 19	q = 23	q = 27	q = 31

CFDFp	p							
	8	9	10	11	12	13	14	15
CFDFpLsL	q = 32	q = 36	q = 40	q = 44	q = 48	q = 52	q = 56	q = 60
CFDFpLsH	q = 33	q = 37	q = 41	q = 45	q = 49	q = 53	q = 57	q = 61
CFDFpHsL	q = 34	q = 38	q = 42	q = 46	q = 50	q = 54	q = 58	q = 62
CFDFpHsH	q = 35	q = 39	q = 43	q = 47	q = 51	q = 55	q = 59	q = 63

The FIFO Access Data Field p Register stores the data bytes of the message.

In TX mode, this register can be read data from FIFO buffers, only for the current entry based on the write pointer value, not for the other entries.

- **CFDBq[7:0]**

  Data Byte q of the Message present in the FIFO buffer.

  In TX mode, these bits can be read data from FIFO buffers, or be written data to FIFO buffers.

  In RX mode, these bits can only be read data from FIFO buffers.

  In RX mode, unused data bytes will be filled with 00H, according to their configured data payload size set in **CFCC.CFPLS[2:0]**.

19.5.13 CANs TX Message Buffer ID Register m (TMIDmHs, TMIDmLs) [m = 0 to 3] [s = 0, 1]

Address: TMIDmLs: F0504H + (4CH×m), TMIDmHs: F0506H + (4CH×m)
TMIDmLsL: F0504H + (4CH×m), TMIDmLsH: F0505H + (4CH×m),
TMIDmHsL: F0506H + (4CH×m), TMIDmHsH: F0507H + (4CH×m)

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TMIDmHs	TMIDE	TMRTR	THLEN	TMID[28:16]												
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TMIDmLs	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
31	TMIDE	TX Message Buffer IDE Bit	0: STD-ID will be transmitted 1: EXT-ID will be transmitted	R/W
30	TMRTR	TX Message Buffer RTR Bit	0: Data Frame 1: Remote Frame	R/W
29	THLEN	Tx History List Entry	0: Entry not stored in THL after successful TX 1: Entry stored in THL after successful TX	R/W
28 to 0	TMID[28:0]	TX Message Buffer ID Field	STD-ID / EXT-ID fields	R/W

The CANs TX Message Buffer ID Register is used to store the ID, IDE, RTR fields and history configuration of the message to be transmitted from the associated buffer.

- **TMIDE**

  This bit selects whether a message with EXT-ID or STD-ID will be transmitted from this TX Message Buffer.
  Do not write to this bit when corresponding channel is in CH_SLEEP.

- **TMRTR**

  This bit selects whether a Data Frame or a Remote Frame will be transmitted from this TX Message Buffer.
  Do not write to this bit when corresponding channel is in CH_SLEEP.

  Caution There are no remote frames in CAN-FD format. In case of CAN-FD transmission (TMFDCTRm.TMFDF = 1) the bit is always transmitted dominant (data frame).

- **THLEN**

  This bit controls the storage of an entry corresponding to the transmitted message in the TX History List at the end of a successful transmission.
  Do not write to this bit when corresponding channel is in CH_SLEEP.

- **TMID[28:0]**

  These are the bits of the STD-ID / EXT-ID fields of the message stored in this TX MB.
  For alignment of these bits in standard and extended frame format, see **19.4 Identifier Bits Alignment**.
  Do not write to these bits when corresponding channel is in CH_SLEEP.

19.5.14 CANs TX Message Buffer Pointer Register m (TMPTRmHs) [m = 0 to 3] [s = 0, 1]

Address: TMPTRmHs: F050AH + (4CH×m)
TMPTRmHsH: F050BH + (4CH×m)

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TMPTRmHs	TMDLC[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 12	TMDLC[3:0]	TX Message Buffer DLC Field	Number of Data Bytes to be transmitted in a CAN Frame.	R/W
11 to 0	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W

The CANs TX Message Buffer Pointer Register is used to store the DLC fields of the message to transmit from the associated buffer.

- **TMDLC[3:0]**
  These bits select the number of Data Bytes that will be transmitted from this TX Message Buffer if corresponding **TMIDm.TMRTR** bit is configured as 0.
  Do not write to these bits when corresponding channel is in CH_SLEEP.

19.5.15 CANs TX Message Buffer CAN-FD Control Register m (TMFDCTRmHs, TMFDCTRmLs) [m = 0 to 3] [s = 0, 1]

Address: TMFDCTRmLs: F050CH + (4CH×m), TMFDCTRmHs: F050EH + (4CH×m)
TMFDCTRmLsL: F050CH + (4CH×m), TMFDCTRmLsH: F050DH + (4CH×m),
TMFDCTRmHsL: F050EH + (4CH×m), TMFDCTRmHsH: F050FH + (4CH×m)

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TMFDCTRmHs	TMPTR[15:0]															
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TMFDCTRmLs	–	–	–	–	–	–	TMIFL[1:0]		–	–	–	–	–	TMFDF	TMBRS	TMESI
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
31 to 16	TMPTR[15:0]	TX Message Buffer Pointer Field	TX Message Buffer Pointer	R/W
15 to 10	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
9, 8	TMIFL[1:0]	TX Message Buffer Information Label Field	TX Message Buffer Information Label	R/W
7 to 3	–	Reserved	These bits are read as 0. The write value should be always 0.	R/W
2	TMFDF	CAN-FD Format bit	0: Non CAN-FD frame to transmit 1: CAN-FD frame to transmit	R/W
1	TMBRS	Bit Rate Switch bit	0: CAN-FD frame to transmit with no bit rate switch 1: CAN-FD frame to transmit with bit rate switch	R/W
0	TMESI	Error State Indicator bit	0: CAN-FD frame to transmit by error active node 1: CAN-FD frame to transmit by error passive node	R/W

The CANs TX Message Buffer CAN-FD Control Register shows the status of the FDF, BRS and ESI bits and pointer fields of the CAN-FD frame to be transmitted.

- **TMPTR[15:0]**

  The pointer value is stored in these bits. It will be copied, together with further message information, in the TX History List after successful transmission of the message.
  Do not write to these bits when corresponding channel is in CH_SLEEP.

- **TMIFL[1:0]**

  The Information label value is stored in these bits. It will be copied, together with further message information, in the TX History List after successful transmission of the message.
  Do not write to these bits when corresponding channel is in CH_SLEEP.

- **TMFDF**

  Do not write to this bit when corresponding channel is in CH_SLEEP.

- **TMBRS**

  Do not write to this bit when corresponding channel is in CH_SLEEP.

- **TMESI**

  If the channel is not in error passive, then this bit equals the write value, else it is don’t care and the bit is transmitted as 1 on the CAN bus; indicating that this is an error passive node.
  Do not write to this bit when corresponding channel is in CH_SLEEP.

19.5.16 CANs TX Message Buffer Data Field p Register m (TMDFm_pHs, TMDFm_pLs) [m = 0 to 3], [p = 0 to 15] [s = 0, 1]

Address: TMDFm_pLs: F0510H + (4CH×m) + (4×p), TMDFm_pHs: F0512H + (4CH×m) + (4×p)
TMDFm_pLsL: F0510H + (4CH×m) + (4×p), TMDFm_pLsH: F0511H + (4CH×m) + (4×p),
TMDFm_pHsL: F0512H + (4CH×m) + (4×p), TMDFm_pHsH: F0513H + (4CH×m) + (4×p)

Condition: **CFDGRWCs.RPAGE[1:0]** = 01B

Symbol	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TMDFm_pHs	TMDBq[7:0]								TMDBq[7:0]							
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TMDFm_pLs	TMDBq[7:0]								TMDBq[7:0]							
After Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Description	R/W
31 to 24	TMDBq[7:0]	TX Message Buffer Data Byte q	R/W
23 to 16	TMDBq[7:0]	TX Message Buffer Data Byte q	R/W
15 to 8	TMDBq[7:0]	TX Message Buffer Data Byte q	R/W
7 to 0	TMDBq[7:0]	TX Message Buffer Data Byte q	R/W

TMDFm_p	p							
	0	1	2	3	4	5	6	7
TMDFm_pLsL	q = 0	q = 4	q = 8	q = 12	q = 16	q = 20	q = 24	q = 28
TMDFm_pLsH	q = 1	q = 5	q = 9	q = 13	q = 17	q = 21	q = 25	q = 29
TMDFm_pHsL	q = 2	q = 6	q = 10	q = 14	q = 18	q = 22	q = 26	q = 30
TMDFm_pHsH	q = 3	q = 7	q = 11	q = 15	q = 19	q = 23	q = 27	q = 31

TMDFm_p	p							
	8	9	10	11	12	13	14	15
TMDFm_pLsL	q = 32	q = 36	q = 40	q = 44	q = 48	q = 52	q = 56	q = 60
TMDFm_pLsH	q = 33	q = 37	q = 41	q = 45	q = 49	q = 53	q = 57	q = 61
TMDFm_pHsL	q = 34	q = 38	q = 42	q = 46	q = 50	q = 54	q = 58	q = 62
TMDFm_pHsH	q = 35	q = 39	q = 43	q = 47	q = 51	q = 55	q = 59	q = 63

The CANs TX Message Buffer Data Field p Register is used to store the data bytes of the message to transmit from the associated buffer.

- **TMDBq[7:0]**

  Data byte q of the message stored in the TX Message Buffer.
  Do not write to these bits when corresponding channel is in CH_SLEEP.

19.6 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 **19.6.1 Global Modes**, and details of channel modes are described in **19.6.2 Channel Modes**.

<Global modes>

- Global sleep mode : Stops clocks of entire module to achieve low power consumption.
- Global reset mode : Performs initial settings for entire module.
- Global halt mode : All communications are suspended with no change of status and flag registers.
- Global operation mode : Makes entire module operable.

<Channel modes>

- Channel sleep mode : Stops channel clock.
- Channel reset mode : Performs initial settings for channels.
- Channel halt mode : Stops CAN communication and enables channel test.
- Channel operation mode : Performs CAN communication.

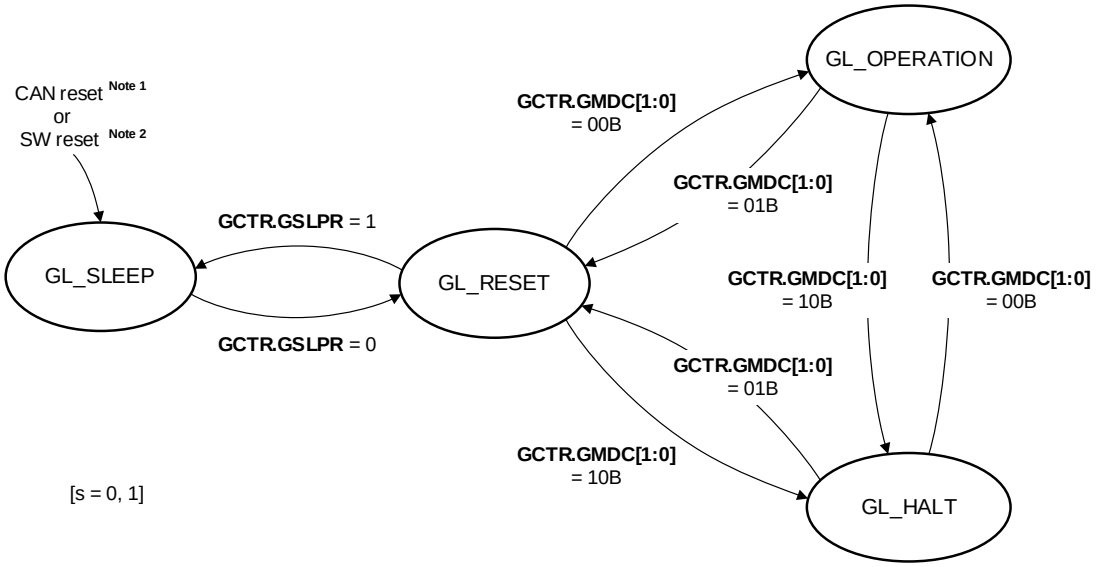
19.6.1 Global Modes

These modes are applicable for the complete RS-CANFD lite module and hence are called “global” modes. The global modes of the RS-CANFD lite module are:

- Global sleep mode : GL_SLEEP
- Global reset mode : GL_RESET
- Global halt mode : GL_HALT
- Global operation mode : GL_OPERATION

Figure 19-5 shows the transitions of global modes.

Figure 19-5. Transitions of Global Modes



Note 1. CAN reset: When the **PER2.CANsEN** bit is changed from 0 to 1.

2. SW reset: When the **GRSTC.SRST** bit is cleared to 0 after being set to 1.

Change in the Global mode can affect the Channel mode. **Table 19-9** below shows the effect of a Global mode transition on the Channel mode.

Table 19-9. Transitions of Channel Modes Depending on Global Mode Setting

Target Global Mode / Current Global Mode	GL_SLEEP	GL_RESET	GL_HALT	GL_OPERATION
GL_SLEEP		CH_SLEEP: keep CH_RESET: − CH_HALT: − CH_OPERATOPN: −		
GL_RESET	CH_SLEEP: keep CH_RESET: → CH_SLEEP CH_HALT: − CH_OPERATOPN: −		CH_SLEEP: keep CH_RESET: keep CH_HALT: − CH_OPERATOPN: −	CH_SLEEP: keep CH_RESET: keep CH_HALT: − CH_OPERATOPN: −
GL_HALT		CH_SLEEP: keep CH_RESET: keep CH_HALT: → CH_RESET CH_OPERATOPN: −		CH_SLEEP: keep CH_RESET: keep CH_HALT: keep CH_OPERATOPN: −
GL_OPERATION		CH_SLEEP: keep CH_RESET: keep CH_HALT: → CH_RESET CH_OPERATOPN : → CH_RESET	CH_SLEEP: keep CH_RESET: keep CH_HALT: keep CH_OPERATOPN: → CH_HALT	

19.6.1.1 Global Sleep Mode

After the release of the CAN reset or after setting and clearing a **GRSTC.SRST** bit, the RS-CANFD lite module automatically enters Global Sleep Mode.

The RS-CANFD lite module will also enter this mode, when the Global Sleep Request bit is set while it is in Global Reset Mode.

This control bit cannot be set in Global Halt Mode or Global Operation Mode.

Setting the Global Sleep Request bit will set all Channel Sleep Request bits and force all channels into the Channel Sleep Mode.

Sleep Mode is used for power saving purpose. When RS-CANFD lite module is in Global Sleep Mode, only the clock for CPU write access to the Global Sleep Mode Request bit is active. All other clocks are stopped and all other functions of the RS-CANFD lite module are suspended.

Read access from all registers is still possible and all register values are preserved.

After setting Global Sleep Request bit, it is necessary to confirm that the Global Sleep status has been updated indicating successful transition to Global Sleep Mode before the Global Sleep Request bit can be cleared again.

Figure 19-6. Procedure for Entering Global Sleep Mode from Global Operation or Global HALT Mode

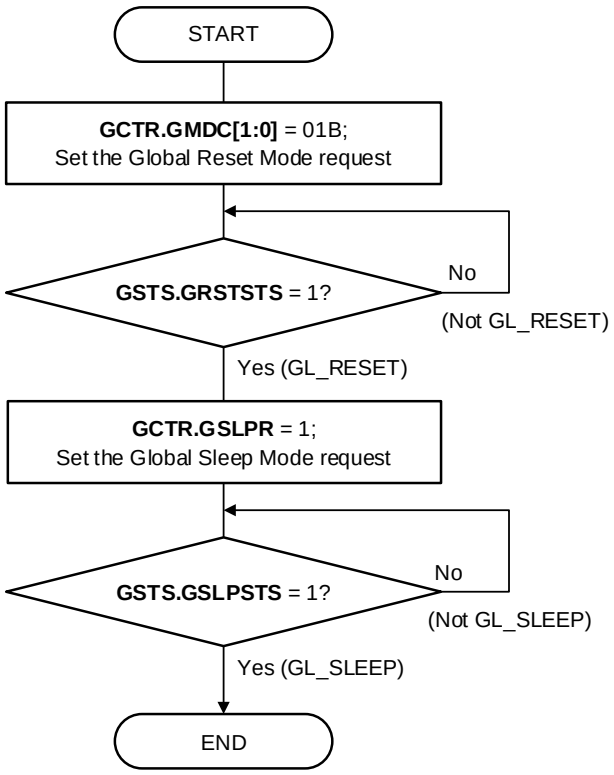
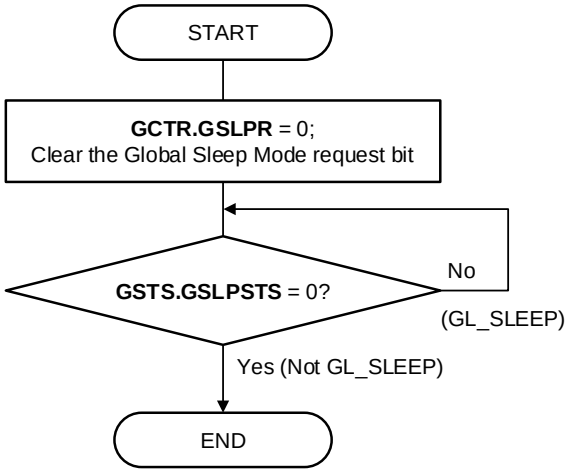


Figure 19-7. Procedure for Exiting Global Sleep Mode



19.6.1.2 Global Reset Mode

The RS-CANFD lite module enters this mode in the following ways:

Global Mode Control (**GCTR.GMDC[1:0]**) in the Global Control Register is configured for Global Reset Mode while the RS-CANFD lite module is in Global Halt Mode or Global Operation Mode

Global Sleep Mode request bit is cleared while RS-CANFD lite module is in Global Sleep Mode

In Global Reset Mode, all RS-CANFD lite module functions are suspended and all status and flag registers are initialized. Additionally, all FIFOs are disabled and transmission control bits are cleared.

Configuration registers (except the test mode registers) are not initialized in this mode to their MCU Reset values and the RS-CANFD lite module can be configured.

Refer to **19.6.3 Global Mode – Channel Mode Transition Interactions** for detailed description of the behavior of all registers when transition to Global Reset Mode is performed.

Setting the global mode to Reset by setting the Global Mode Control bits (**GCTR.GMDC[1:0]**) to 01B will set all Channel Mode Control bits (**CsCTR.CHMDC[1:0]**) to 01B and force all channels into the Channel Reset Mode.

For channels that are already in Channel Reset Mode or Channel Sleep Mode this automatic transition is not performed (**CsCTR.CHMDC[1:0]** of related channel already set to 01B).

After setting **GCTR.GMDC[1:0]** to Reset Mode, it is necessary to confirm that the Reset Mode Status (**GSTS.GRSTSTS**) has been updated indicating successful transition to Global Reset Mode before **GCTR.GMDC[1:0]** can be changed again.

Figure 19-8. Procedure for Entering Global Reset Mode

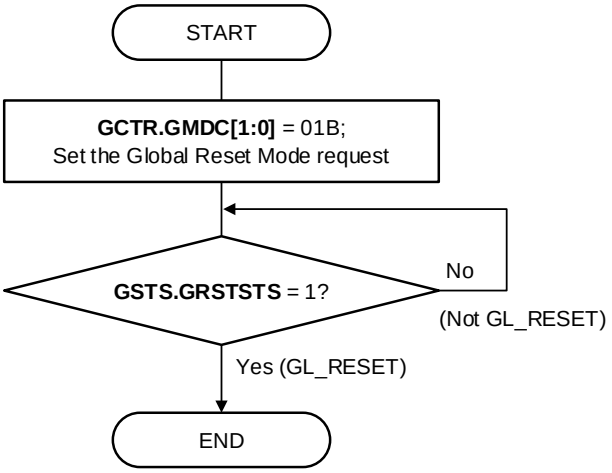
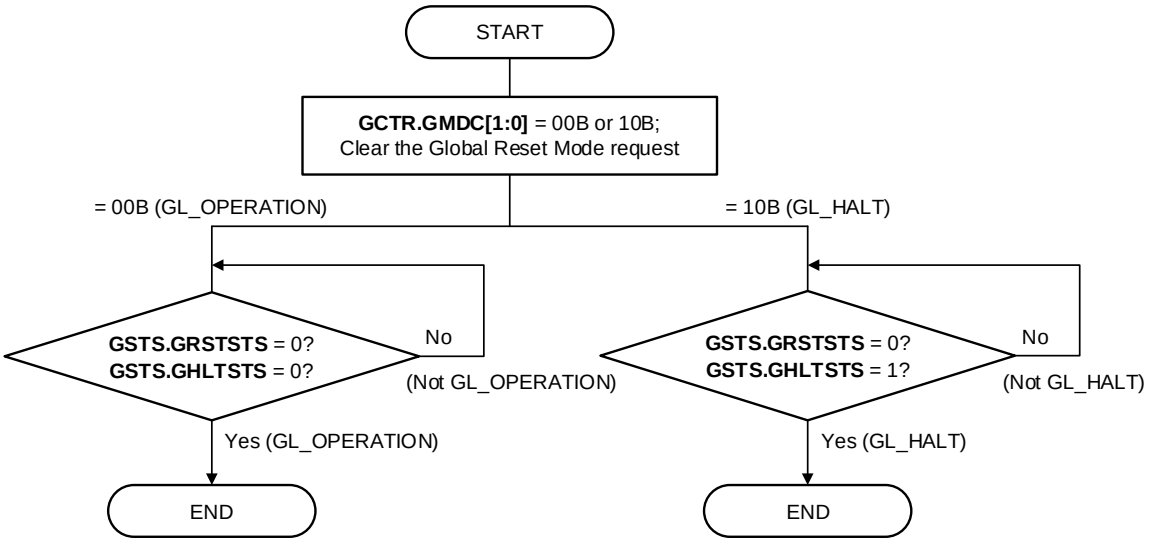


Figure 19-9. Procedure for Exiting Global Reset Mode



19.6.1.3 Global Halt Mode

The RS-CANFD lite module enters this mode in the following ways:

- Global Mode Control (**GCTR.GMDC[1:0]**) in the Global Control Register is configured for Global Halt Mode while the RS-CANFD lite module is in Global Reset Mode
  - The channels will be in either Channel Reset Mode or Channel Sleep Mode and will remain in this mode.
- Global Mode Control (**GCTR.GMDC[1:0]**) in the Global Control Register is configured for Global Halt Mode while the RS-CANFD lite module is in Global Operation Mode
  - All channels in Channel Reset Mode, Channel Halt Mode or Channel Sleep Mode will remain in this mode.
  - All channels in Channel Operation Mode will transit to Channel Halt Mode.
  - Global Halt Mode Status bit is set when all channels have left Channel Operation Mode.

If a transmission or reception is ongoing for a channel the transition to Channel Halt Mode is delayed until the completion of the communication.

Similarly, if a channel is in Bus-Off, the full Bus-Off recovery sequence may be delayed depending on the channel configuration.

In the Global Halt Mode, all communications are suspended and RS-CANFD lite logic will not cause any change to status and flag registers (only when a channel is in the Bus-Off its REC and TEC values are cleared).

Also the test mode configuration and control registers are not initialized in this mode.

The Global Halt Mode should be used to configure global module test modes.

Refer to **19.6.3.3 Register Behavior in Global/Channel Modes** for a detailed description of the behavior of all registers when transition to Global Halt Mode is performed.

Setting the global mode to Halt by setting the Global Mode Control bits **GCTR.GMDC[1:0]** in the Global Control Register to 10B will set all Channel Mode Control bits **CsCTR.CHMDC[1:0]** in the Channel Control Registers to 10B for the channels that are in Channel Operation Mode and force these channels into the Channel Halt Mode.

For channels that are already in Channel Reset Mode, Channel Halt Mode or Channel Sleep Mode this automatic transition is not performed.

Therefore, the Global Halt Mode request can be used to shut down all CAN channel communications without loss of messages and disruption on the related CAN Bus (no interruption of reception/transmission processes on the channels).

After setting Global Mode Control **GCTR.GMDC[1:0]** to Halt Mode it is necessary to confirm that the Halt Mode status **GSTS.GHLTSTS** in the Global Status Register has been updated indicating successful transition to Global Halt Mode.

Do not configure any other SFR settings until confirming **GSTS.GHLTSTS** is set.

Figure 19-10. Procedure for Entering Global Halt Mode

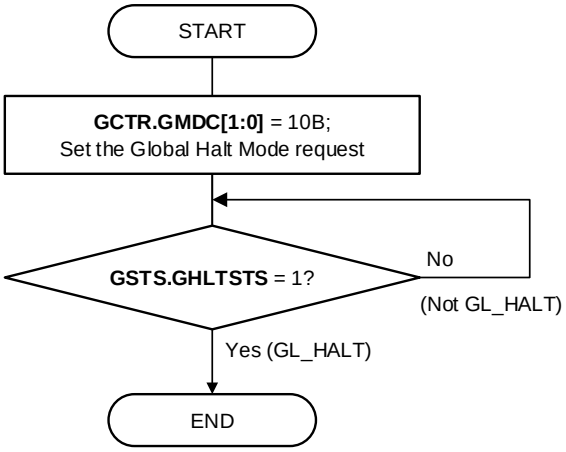
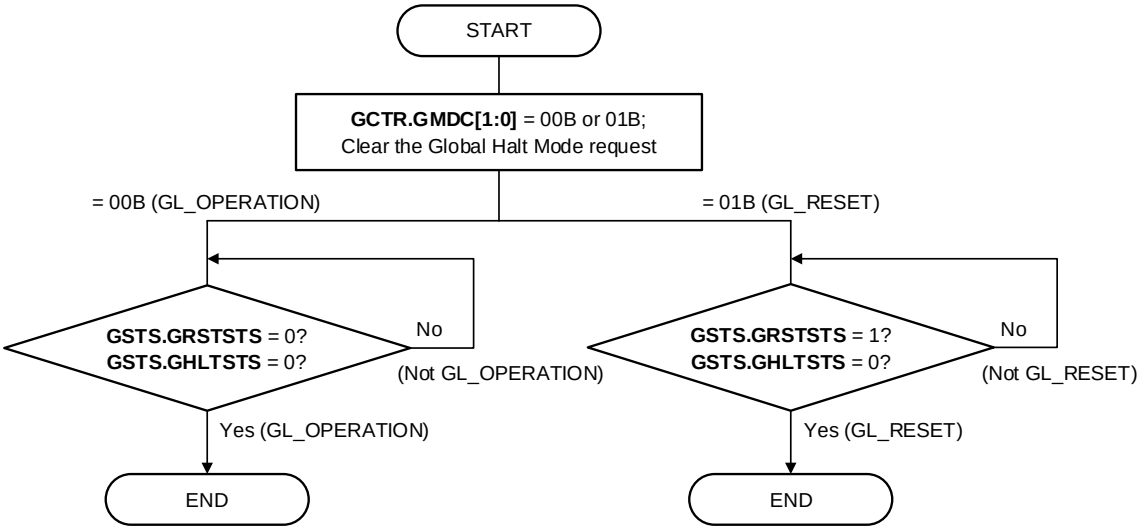


Figure 19-11. Procedure for Exiting Global Halt Mode



19.6.1.4 Global Operation Mode

The RS-CANFD lite module enters this mode when the Global Mode Configuration bits are set to Global Operation Mode.

CAN channels can only be set to Channel Operation Mode and start CAN communication when RS-CANFD lite is in Global Operation Mode.

After setting Global Mode Control **GCTR.GMDC[1:0]** to Global Operation Mode it is necessary to confirm that the Global Reset Mode status **GSTS.GRSTSTS** and the Global Halt Mode status **GSTS.GHLTSTS** in the Global Status Register have been cleared indicating successful transition to Global Operation Mode before **GCTR.GMDC[1:0]** can be changed again.

Figure 19-12. Procedure for Entering Global Operation Mode

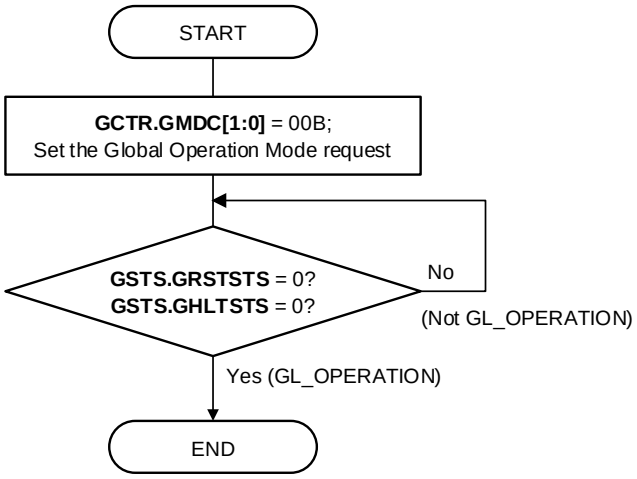
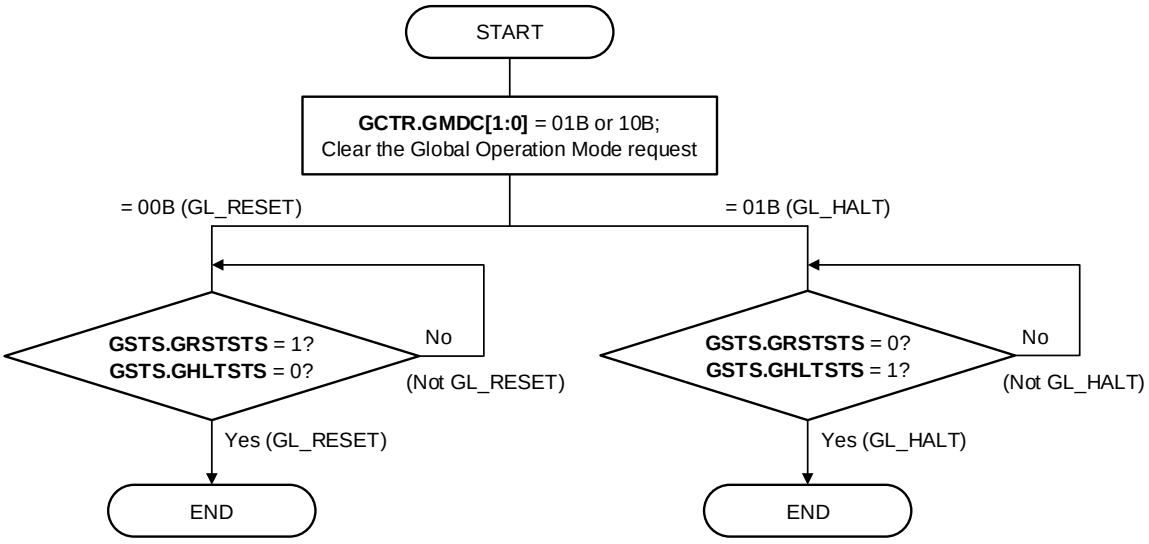


Figure 19-13. Procedure for Exiting Global Operation Mode



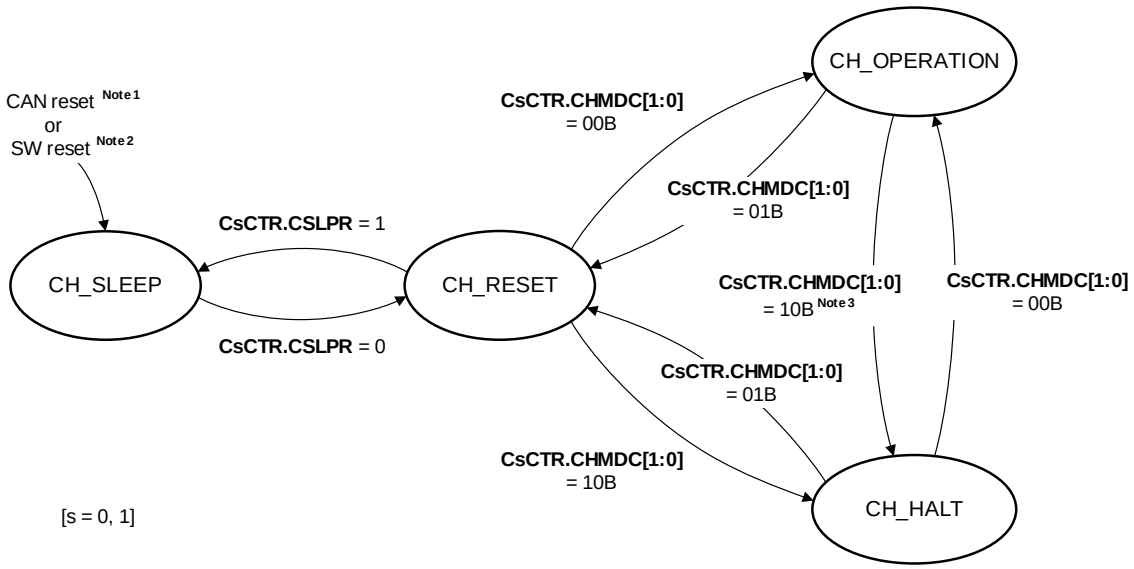
19.6.2 Channel Modes

CAN channel can be in one of the following four channel modes:

- Reset Mode : CH_RESET
- Halt Mode : CH_HALT
- Operation Mode : CH_OPERATION
- Sleep Mode : CH_SLEEP

Figure 19-14 shows the Transition of Channel Modes.

Figure 19-14. Channel Mode State Transition Chart



Note **1.** CAN reset: When the **PER2.CANsEN** bit is changed from 0 to 1.

2. SW reset: When the **GRSTC.SRST** bit is cleared to 0 after being set to 1.

3. The transition timing depends on **BOM[1:0]** setting in the CANs Control Register.

when **BOM[1:0]** = 01B, the state transition timing is immediate after entering to Bus-Off state.

when **BOM[1:0]** = 10B, the state transition timing is at the end of Bus-Off state.

when **BOM[1:0]** = 11B, the state transition timing is matching with the setting of **CsCTR.CHMDC[1:0]** to Halt Mode.

19.6.2.1 CAN Channel Sleep Mode

After the release of the CAN reset or after setting and clearing a **GRSTC.SRST** bit, CAN channel of the RS-CANFD lite module automatically enters Channel Sleep Mode.

CAN channel will also enter this mode, when the related Channel Sleep Mode Request bit is set while the CAN channel is in Channel Reset Mode.

This control bit should not be set in Channel Halt Mode or Channel Operation Mode.

Entering the CAN channel Sleep Mode instantly stops the clock supplied to the CAN channel unit and thereby reduces power consumption.

After setting Channel Sleep Mode request bit it is necessary to confirm that the Channel Sleep Mode Status has been updated indicating successful transition to Channel Sleep Mode before the Channel Sleep Mode request bit can be cleared again.

During Channel Sleep Mode, channel related reisters cannot be written.

Read operation is still possible.

19.6.2.2 CAN Channel Reset Mode

The RS-CANFD lite CAN channel enters this mode in the following ways:

Channel Mode Control **CsCTR.CHMDC[1:0]** in the CANs Control Registers is configured for Channel Reset Mode while the related CAN channel is in Channel Halt Mode or Channel Operation Mode

Channel Sleep Mode request bit is cleared while the related CAN channel is in Channel Sleep Mode

Global Mode Control **GCTR.GMDC[1:0]** is set to Global Reset Mode and CAN channel is not in Channel Sleep Mode or Channel Reset Mode

In Channel Reset Mode, all CAN channel status and flag registers are initialized.

Additionally, all channel related transmission control bits are cleared.

Configuration registers (except the Channel Test Mode registers) are not initialized in this mode and the RS-CANFD lite module CAN channel can be configured for communication.

Refer to **19.6.3 Global Mode – Channel Mode Transition Interactions** for detailed description of the behavior of all registers when transition to Channel Reset Mode is performed.

After setting Channel Mode Control **CsCTR.CHMDC[1:0]** to Channel Reset Mode, it is necessary to confirm that the Reset Mode Status **CsSTS.CRSTSTS** in the related Channel Status Registers has been updated indicating successful transition to Channel Reset Mode before the related **CsCTR.CHMDC[1:0]** can be changed again.

Refer to **Table 19-10** regarding the influence of transition to Channel Reset Mode while CAN communication is ongoing.

19.6.2.3 CAN Channel Halt Mode

The RS-CANFD lite CAN channel enters this mode in the following ways:

- Channel Mode Control **CsCTR.CHMDC[1:0]** in the Channel Control Registers is configured for Channel Halt Mode while the related CAN channel is in Channel Reset Mode or Channel Operation Mode.
- Global Mode Control **GCTR.GMDC[1:0]** is set to Global Halt Mode and CAN channel is in Channel Operation Mode.

In Channel Halt Mode, all channel CAN communication is suspended but all status and flag registers remain unchanged during Channel Halt Mode entry (except for the Bus-Off case where REC and TEC values are cleared for this channel).

In addition, the channel test mode configuration and control registers are not initialized in this mode.

The Channel Halt Mode should be used to configure channel test modes.

Refer to **19.6.3 Global Mode – Channel Mode Transition Interactions** for detailed description of the behavior of all registers when transition to Channel Halt Mode is performed.

After setting Channel Mode Control **CsCTR.CHMDC[1:0]** to Channel Halt Mode it is necessary to confirm that the Halt Mode status **CsSTS.CHLTSTS** in the related Channel Status Register has been updated indicating successful transition to Channel Halt Mode before the related **CsCTR.CHMDC[1:0]** can be changed again.

Refer to **Table 19-10** regarding the influence of transition to Channel Halt Mode while CAN communication is ongoing.

Table 19-10. Behavior in CAN Reset / Halt Mode

State / Mode	Receiver	Transmitter	Bus-Off
CAN channel Reset Mode (**CsCTR.CHMDC[1:0]** = 01B)	The CAN channel transits to Channel Reset Mode without waiting for the completion of the ongoing reception. **(Note 1)**	The CAN channel transits to Channel Reset Mode without waiting for the completion of the ongoing transmission. **(Note 1)**	The CAN channel transits to Channel Reset Mode without waiting for the completion of the Bus-Off Recovery.
CAN channel Halt Mode (**CsCTR.CHMDC[1:0]** = 10B)	CAN channel transits to Channel Halt Mode at the end of the ongoing reception or error. **(Note2)**	CAN channel transits to Channel Halt Mode after completion of the ongoing transmission.	When **CsCTR.BOM[1:0]** is set to 00B, a Channel Halt Mode request will be accepted only after the completion of the full Bus-Off Recovery sequence. When **CsCTR.BOM[1:0]** is set to 10B, then, the CAN channel transits automatically to Channel Halt Mode after waiting for the completion of the Bus-Off Recovery. When **CsCTR.BOM[1:0]** is set to 01B, then, the CAN channel transits automatically to Channel Halt Mode without waiting for the completion of the Bus-Off Recovery. When **CsCTR.BOM[1:0]** is set to 11B, the CAN channel transits to Channel Halt Mode as soon as the Channel Halt Mode is requested (without waiting for the completion of the Bus-Off Recovery).

Notes 1. If the entry to Channel Reset Mode is required only at the end of an ongoing communication, then Channel Halt Mode can be requested first to prevent interruption of CAN communication by direct transition to Channel Reset Mode. After the CAN channel enters Channel Halt Mode, the Channel Reset Mode can be requested.

2. If CAN communication is locked at dominant level after an error flag, Application SW can detect this situation by monitoring the channel related BusLock Flag and resolve lock condition by setting the CAN channel to Channel Reset Mode.

19.6.2.4 CAN Channel Operation Mode

The Channel Operation Mode is activated by setting the **CsCTR.CHMDC[1:0]** bits to 00B. If 11 consecutive recessive bits are detected after entering the CAN Operation Mode, then the **CsSTS.COMSTS** bit is set and the CAN channel:

- Enables channel's communication functions allowing the channel to become an active node on the CAN network.
- Releases the internal fault confinement logic including receive and transmit error counters.

At this point, it can start transmission and reception of CAN messages.

Within the CAN channel Operation Mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (see **Figure 19-15**):

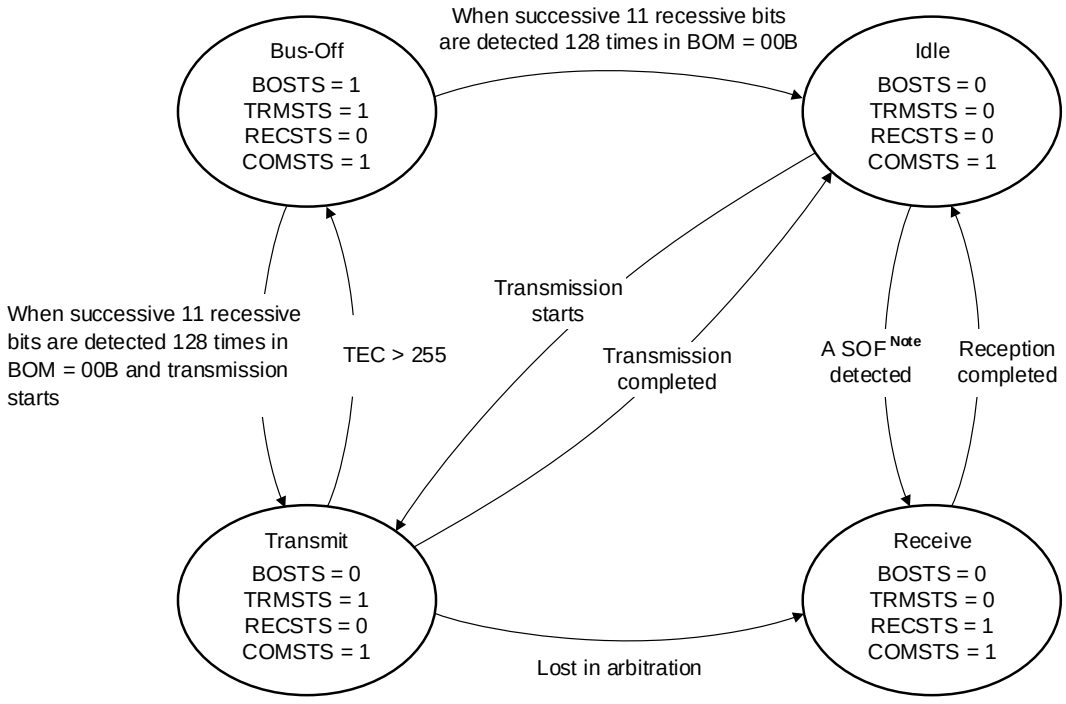
- Idle (channel idle): The CAN channel is neither receiving nor transmitting.
- Receive (channel receives): The channel is receiving a CAN message sent by another CAN node.
- Transmit (channel transmits): The channel is transmitting a CAN message.

  Note The channel may receive its own message simultaneously when Self Test Mode is enabled.

- Bus-Off (channel is in Bus-Off state): The CAN channel is cut-off from CAN Bus communication.

After setting Channel Mode Control **CsCTR.CHMDC[1:0]** to Operation Mode, it is necessary to confirm that the Channel Reset Mode Status **CsSTS.CRSTSTS** and the Channel Halt Mode Status **CsSTS.CHLTSTS** in the Channel Status Register have been updated indicating successful transition to Channel Operation Mode before the related **CsCTR.CHMDC[1:0]** can be changed again.

Figure 19-15. Sub-Modes of CAN Channel Operation Mode (only CsCTR.BOM[1:0] = 00B)



Note SOF: Start of Frame

Remark Bit symbol representing the condition:

BOSTS: **CsSTS.BOSTS** (Channel Bus-Off Status)
TRMSTS: **CsSTS.TRMSTS** (Channel Transmit Status)
RECSTS: **CsSTS.RECSTS** (Channel Receive Status)
COMSTS: **CsSTS.COMSTS** (Channel Communication Status)
BOM: **CsCTR.BOM[1:0]** (Channel Bus-Off Mode)

19.6.2.5 CAN Channel Bus-Off State

The CAN channel Bus-Off state is entered according to the fault confinement rules of the CAN specification. Following modes for returning to the CAN channel Operation Mode from the Bus-Off state can be configured:

- **CsCTR.BOM[1:0]** = 00B:
  Bus-Off recovery is compliant to ISO 11898-1, namely the CAN channel re-enters CAN communication (Error Active state) after 11 consecutive recessive bits are detected 128 times. TEC and REC counters are initialized to zero. The Bus-Off Recovery Flag **CsERFL.BORF** is set in this case.
- **CsCTR.BOM[1:0]** = 01B:
  The CAN channel changes the value of the **CsCTR.CHMDC[1:0]** bits within the CAN channel Control Register to 10B and switches immediately to Channel Halt Mode automatically after entering the Bus-Off state. TEC and REC counters are initialized to zero. The Bus-Off Recovery Flag **CsERFL.BORF** is not set in this case.
- **CsCTR.BOM[1:0]** = 10B:
  The CAN channel changes the value of the **CsCTR.CHMDC[1:0]** bits within the CAN channel Control Register to 10B as soon as it reaches the Bus-Off state and enters Channel Halt Mode automatically after the CAN channel has completed the Bus-Off recovery sequence (i.e. after 11 consecutive recessive bits are detected 128 times). TEC and REC counters are initialized to zero. The Bus-Off Recovery Flag **CsERFL.BORF** is set in this case.
- **CsCTR.BOM[1:0]** = 11B:
  Bus-Off recovery is initiated but CAN channel can enter immediately the Channel Halt Mode when still in Bus-Off state if a request is made to enter Channel Halt Mode.
  TEC and REC counters are initialized to zero. In this case, the Bus-Off Recovery Flag **CsERFL.BORF** is not set and TEC and REC counters are initialized to zero.

  Without setting **CsCTR.CHMDC[1:0]** = 10B and when 11 recessive bits is detected 128 times continuously, transition conditions become the same as **CsCTR.BOM[1:0]** = 00B.

Note, however, that if the recovery from Bus-Off occurs normally in this mode (i.e. after waiting for 128 sequences of 11 consecutive recessive bits), and no Halt request has been generated during this period, then the Bus-Off Recovery Flag **CsERFL.BORF** is set.

In the case where Application SW writes into **CsCTR.CHMDC[1:0]** at the same time as the CAN channel is due to enter Halt Mode (at the start of Bus-Off when **CsCTR.BOM[1:0]** = 01B, or at the end of Bus-Off when **CsCTR.BOM[1:0]** = 10B) then the System SW request will have the highest priority.

Note that, in the above cases, the automatic setting of the **CsCTR.CHMDC[1:0]** to Channel Halt mode request is performed when the **CsCTR.CHMDC[1:0]** value is previously 00B (Channel Operation Mode).

Additionally, it is possible to force the CAN channel to recover from the Bus-Off state by setting **CsCTR.RTBO** to 1. The error state changes from Bus-Off state to integrating state with a maximum delay of 1 CAN Bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The Bus-Off Recovery Flag is not set in this case.

TEC and REC counters are initialized to zero.

Before setting **CsCTR.RTBO** to 1, all pending transmissions from TX Message Buffers and/or Common FIFO in TX Mode should be disabled.

The disable of the pending transmission Message Buffer or FIFO must be confirmed by the corresponding acknowledge flags.

For TX Message Buffer these are the Transmission Result Flags (**TMSTSm.TMTRF[1:0]**) and for the FIFO it is the FIFO empty flag (**CFSTS.CFEMP**).

The **CsCTR.RTBO** bit should be used for Bus-Off recovery only when **CsCTR.BOM[1:0]** is set to 00B.

Setting this bit in any state (other than Bus-Off) will have no effect and the bit will be cleared immediately.

Table 19-11 summarizes the setting of Bus-Off Entry Flag **CsERFL.BOEF** and Bus-Off Recovery Flag **CsERFL.BORF** for different configurations of **CsCTR.BOM[1:0]**.

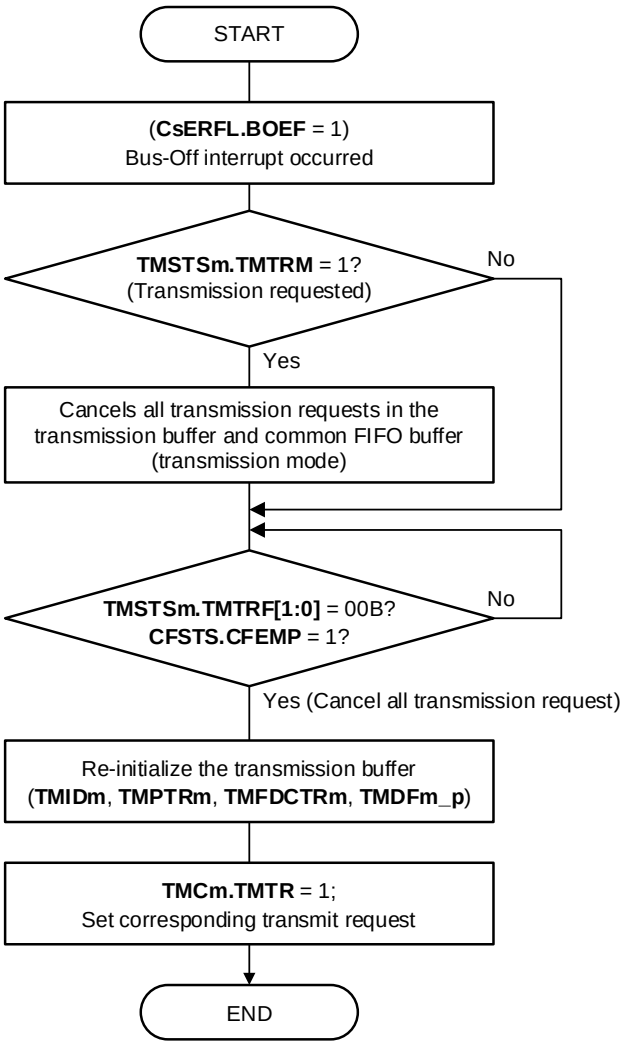
Table 19-11. Bus-Off Entry / Recovery Flag Behavior

BOM[1:0]	**BOEF** bit set	**BORF** bit set
00B	Always (on entry to Bus-Off)	Always (on exit from Bus-Off)
00B **CsCTR.RTBO** set to 1	Always (on entry to Bus-Off)	Only if normal Bus-Off recovery occurs before System SW sets **CsCTR.RTBO** to 1
01B	Always (on entry to Bus-Off)	Never
10B	Always (on entry to Bus-Off)	Always (on exit from Bus-Off)
11B	Always (on entry to Bus-Off)	Only if normal Bus-Off recovery occurs before System SW issues Halt request

To make an efficient SW procedure it is not mandatory to wait for Bus-Off recovery sequence end.

It is possible to do the transmission re-initialization during the Bus-Off recovery. To do this following SW flow is recommended to use, refer to **Figure 19-16**.

Figure 19-16. Transmission Re-Initialization During Bus-Off



19.6.3 Global Mode – Channel Mode Transition Interactions

In the following, the interaction between global mode setting and channel mode setting is summarized:

Changing of Channel Mode Control **CsCTR.CHMDC[1:0]** in the CANs Control Registers does not have any influence on the Global Mode Control **GCTR.GMDC[1:0]**.

Changing of Global Mode Control **GCTR.GMDC[1:0]** is influencing the channel mode control in the way described in **Table 19-12**.

Table 19-12. Global Mode – Channel Mode Transition Interaction

Global Mode Change	Channel Mode	Channel Mode Transition Behavior
GL_SLEEP → GL_RESET	CH_SLEEP	Channel remains in Sleep Mode
GL_SLEEP → GL_HALT	– (Global Mode unchangeable)	
GL_SLEEP → GL_OPERATION	– (Global Mode unchangeable)	
GL_RESET → GL_SLEEP	CH_SLEEP	Channel remains in Sleep Mode
	CH_RESET	Channel Sleep request bit is set automatically, channel transits to Sleep Mode
GL_RESET → GL_HALT	CH_SLEEP	Channel remains in Sleep Mode
	CH_RESET	Channel remains in Reset Mode
GL_RESET → GL_OPERATION	CH_SLEEP	Channel remains in Sleep Mode
	CH_RESET	Channel remains in Reset Mode
GL_HALT → GL_SLEEP	– (Global Mode unchangeable)	
GL_HALT → GL_RESET	CH_SLEEP	Channel remains in Sleep Mode
	CH_RESET	Channel remains in Reset Mode
	CH_HALT	Channel mode control is set to Reset Mode, channel transits to Reset Mode
GL_HALT → GL_OPERATION	CH_SLEEP	Channel remains in Sleep Mode
	CH_RESET	Channel remains in Reset Mode
	CH_HALT	Channel remains in Halt Mode
GL_OPERATION → GL_SLEEP	– (Global Mode unchangeable)	
GL_OPERATION → GL_RESET	CH_SLEEP	Channel remains in Sleep Mode
	CH_RESET	Channel remains in Reset Mode
	CH_HALT	Channel mode control is set to Reset Mode, channel transits to Reset Mode
	CH_OPERATION	Channel mode control is set to Reset Mode, channel transits to Reset Mode
GL_OPERATION → GL_HALT	CH_SLEEP	Channel remains in Sleep Mode
	CH_RESET	Channel remains in Reset Mode
	CH_HALT	Channel remains in Halt Mode
	CH_OPERATION	Channel mode control is set to Halt Mode, channel transits to Halt Mode after communication finished

19.6.3.1 Global Mode Change Timing

The transition time for the Global mode changes is shown below.

Table 19-13. Transition Time for The Global Mode Changes

From	To	Max. Transition Time
GL_SLEEP	GL_RESET	3 peripheral clock cycle Note 2
GL_RESET	GL_SLEEP	3 peripheral clock cycle
GL_RESET	GL_HALT	10 peripheral clock cycle
GL_RESET	GL_OPERATION	10 peripheral clock cycle
GL_HALT	GL_RESET	2 CAN bit times Note 3
GL_HALT	GL_OPERATION	3 peripheral clock cycle
GL_OPERATION	GL_RESET	2 CAN bit times Note 3
GL_OPERATION	GL_HALT	3 CAN frames Notes 1, 3

Notes 1. The given transition time is the time without any errors on the bus. In case of error condition, the transition time could lengthen to an uncalculated result. As well it could also come to stuck condition in case of locked RX lines or continues error conditions.

2. When transitioning from GL_SLEEP mode, **GSTS.GRAMINIT** should be 0 (initialization of RAM for CAN is complete).

3. CAN frame and CAN bit time depend on the communication speed setting of the CAN channel.

19.6.3.2 Channel Mode Change Timing

The transition time for the channel mode changes is shown below.

Table 19-14. Transition Time for The Channel Mode Changes

From	To	Max. Transition Time
CH_SLEEP	CH_RESET	3 peripheral clock cycle
CH_RESET	CH_SLEEP	3 peripheral clock cycle
CH_RESET	CH_HALT	3 CAN bit times
CH_RESET	CH_OPERATION	4 CAN bit times
CH_HALT	CH_RESET	2 CAN bit times
CH_HALT	CH_OPERATION	4 CAN bit times Note 3
CH_OPERATION	CH_RESET	2 CAN bit times
CH_OPERATION	CH_HALT	2 CAN frames Notes 1, 2

Notes 1. The time specified for this transition does not include the case where channel enters Bus-Off state. In case of Bus-Off the timing depends upon the configuration of the **CsCTR.BOM[1:0]** bits.

2. The given transition time is the time without any errors on the bus. In case of error condition, the transition time could lengthen to an uncalculated result. As well it could also come to a stuck condition in case of locked RX lines or continues error conditions.

3. In general, if the baudrate prescaler value **CsNCFG.NBRP[9:0]** is changed in CH_HALT then the transition time could be deviate from above. As the internal prescaler is a free running down counter to create the Tq clock. And new BRP value will earliest be captured when the counter reached the value zero.

19.6.3.3 Register Behavior in Global/Channel Modes

The tables on the following pages show the behavior of the bits when the mode of the RS-CANFD lite module changes.

Remark Meaning of symbols in the table:
unch: un-changed
und: undefined
MCU_RESET: CAN reset (When the **PER2.CANsEN** bit is changed from 0 to 1)
SW_RESET: SW reset (When the **GRSTC.SRST** bit is cleared to 0 after being set to 1.)

Table 19-15. Register Behavior in Global / Channel Modes (1/8)

Register Symbol	Bit Symbol	Initial Value (after transition, in the state: Initialized value is kept in the state)							
		MCU_ RESET	SW_ RESET	GL_SLEEP	GL_RESET	GL_HALT	CH_SLEEP	CH_RESET	CH_HALT
CsNCFG	**NBRP[9:0]**	all “0”	all “0”	unch	unch	unch	unch	unch	unch
	NSJW[6:0]	all “0”	all “0”	unch	unch	unch	unch	unch	unch
	NTSEG1[7:0]	all “0”	all “0”	unch	unch	unch	unch	unch	unch
	NTSEG2[6:0]	all “0”	all “0”	unch	unch	unch	unch	unch	unch
CsCTR	**CHMDC[1:0]**	01B	01B	unch	01B	In channel Sleep, Reset or Halt: unch, Others: 10B	unch	01B	10B
	CSLPR	1	1	1	unch	unch	unch	unch	unch
	RTBO	0	0	unch	unch	unch	unch	unch	unch
	BEIE	0	0	unch	unch	unch	unch	unch	unch
	EWIE	0	0	unch	unch	unch	unch	unch	unch
	EPIE	0	0	unch	unch	unch	unch	unch	unch
	BOEIE	0	0	unch	unch	unch	unch	unch	unch
	BORIE	0	0	unch	unch	unch	unch	unch	unch
	OLIE	0	0	unch	unch	unch	unch	unch	unch
	BLIE	0	0	unch	unch	unch	unch	unch	unch
	ALIE	0	0	unch	unch	unch	unch	unch	unch
	TAIE	0	0	unch	unch	unch	unch	unch	unch
	EOCOIE	0	0	unch	unch	unch	unch	unch	unch
	SOCOIE	0	0	unch	unch	unch	unch	unch	unch
	TDCVFIE	0	0	unch	unch	unch	unch	unch	unch
	BOM[1:0]	00B	00B	unch	unch	unch	unch	unch	unch
	ERRD	0	0	unch	unch	unch	unch	unch	unch
	CTME	0	0	unch	0	unch	unch	0	unch
	CTMS[1:0]	00B	00B	unch	00B	unch	unch	00B	unch
	BFT	0	0	unch	0	unch	unch	0	unch
	ROM	0	0	unch	0	unch	unch	0	unch

(**Note** is listed at the end of the table.) “unch”: un-changed, “und”: undefined

Table 19-15. Register Behavior in Global / Channel Modes (2/8)

Register Symbol	Bit Symbol	Initial Value (after transition, in the state: Initialized value is kept in the state)							
		MCU_ RESET	SW_ RESET	GL_SLEEP	GL_RESET	GL_HALT	CH_SLEEP	CH_RESET	CH_HALT
CsSTS	**CRSTSTS**	1	1	unch	1	unch	unch	1	0
	CHLTSTS	0	0	unch	0	In channel Sleep, Reset or Halt: unch, Others: 1	unch	0	1
	CSLPSTS	1	1	1	unch	unch	1	unch	unch
	EPSTS	0	0	unch	0	unch	unch	0	unch
	BOSTS	0	0	unch	0	0	unch	0	0
	TRMSTS	0	0	unch	0	0	unch	0	0
	RECSTS	0	0	unch	0	0	unch	0	0
	COMSTS	0	0	unch	0	0	unch	0	0
	ESIF	0	0	unch	0	unch	unch	0	unch
	REC[7:0]	00H	00H	unch	00H	unch	unch	00H	unch
	TEC[7:0]	00H	00H	unch	00H	unch	unch	00H	unch
CsERFL	**BEF**	0	0	unch	0	unch	unch	0	unch
	EWF	0	0	unch	0	unch	unch	0	unch
	EPF	0	0	unch	0	unch	unch	0	unch
	BOEF	0	0	unch	0	unch	unch	0	unch
	BORF	0	0	unch	0	unch	unch	0	unch
	OVLF	0	0	unch	0	unch	unch	0	unch
	BLF	0	0	unch	0	unch	unch	0	unch
	ALF	0	0	unch	0	unch	unch	0	unch
	SERR	0	0	unch	0	unch	unch	0	unch
	FERR	0	0	unch	0	unch	unch	0	unch
	AERR	0	0	unch	0	unch	unch	0	unch
	CERR	0	0	unch	0	unch	unch	0	unch
	B1ERR	0	0	unch	0	unch	unch	0	unch
	B0ERR	0	0	unch	0	unch	unch	0	unch
	ADERR	0	0	unch	0	unch	unch	0	unch
	CRCREG[14:0]	all “0”	all “0”	unch	all “0”	unch	unch	all “0”	unch
GIPV	**IPV[7:0]**	43H	43H	43H	43H	43H	–	–	–
	IPT[1:0]	01B	01B	01B	01B	01B	–	–	–
	CPUBUS	1	1	1	1	1	–	–	–
	PSI[15:0]	3C8BH	3C8BH	3C8BH	3C8BH	3C8BH	–	–	–
GCFG	**TPRI**	0	0	unch	unch	unch	–	–	–
	DCE	0	0	unch	unch	unch	–	–	–
	DRE	0	0	unch	unch	unch	–	–	–
	MME	0	0	unch	unch	unch	–	–	–
	DCS	0	0	unch	unch	unch	–	–	–
	CMPOC	0	0	unch	unch	unch	–	–	–
	TSP[3:0]	0000B	0000B	unch	unch	unch	–	–	–
	TSSS	0	0	unch	unch	unch	–	–	–
	ITRCP[15:0]	0000H	0000H	unch	unch	unch	–	–	–

(Note is listed at the end of the table.) “unch”: un-changed, “und”: undefined

Table 19-15. Register Behavior in Global / Channel Modes (3/8)

Register Symbol	Bit Symbol	Initial Value (after transition, in the state: Initialized value is kept in the state)							
		MCU_ RESET	SW_ RESET	GL_SLEEP	GL_RESET	GL_HALT	CH_SLEEP	CH_RESET	CH_HALT
GCTR	**GMDC[1:0]**	01B	01B	unch	unch	unch	–	–	–
	GSLPR	1	1	unch	unch	unch	unch	unch	unch
	DEIE	0	0	unch	unch	unch	–	–	–
	MEIE	0	0	unch	unch	unch	–	–	–
	THLEIE	0	0	unch	unch	unch	–	–	–
	CMPOFIE	0	0	unch	unch	unch	–	–	–
	TSRST	0	0	unch	unch	unch	–	–	–
GSTS	**GRSTSTS**	1	1	unch	1	0	–	–	–
	GHLTSTS	0	0	unch	0	1	–	–	–
	GSLPSTS	1	1	1	unch	unch	–	–	–
	GRAMINIT	1	1	1 to 0	0	0	–	–	–
GERFL	**DEF**	0	0	unch	0	unch	–	–	–
	MES	0	0	unch	0	unch	–	–	–
	THLES	0	0	unch	0	unch	–	–	–
	CMPOF	0	0	unch	0	unch	–	–	–
	EEF	0	0	unch	0	unch	–	–	–
GTSC	**TS[15:0]**	0000H	0000H	unch	0000H	unch	–	–	–
GAFLECTR	**AFLDAE**	0	0	unch	unch	unch	–	–	–
GAFLCFG	**RNC[4:0]**	all “0”	all “0”	unch	unch	unch	–	–	–
RMNB	**NRXMB[4:0]**	all “0”	all “0”	unch	unch	unch	–	–	–
	RMPLS[2:0]	000B	000B	unch	unch	unch	–	–	–
RMND	**RMNS[15:0]**	0000H	0000H	unch	0000H	unch	–	–	–
RMIEC	**RMIE[15:0]**	0000H	0000H	unch	unch	unch	–	–	–
RFCCk	**RFE**	0	0	unch	0	unch	–	–	–
	RFIE	0	0	unch	unch	unch	–	–	–
	RFPLS[2:0]	000B	000B	unch	unch	unch	–	–	–
	RFDC[2:0]	000B	000B	unch	unch	unch	–	–	–
	RFIM	0	0	unch	unch	unch	–	–	–
	RFIGCV[2:0]	000B	000B	unch	unch	unch	–	–	–
RFSTSk	**RFEMP**	1	1	unch	1	unch	–	–	–
	RFFLL	0	0	unch	0	unch	–	–	–
	RFMLT	0	0	unch	0	unch	–	–	–
	RFIF	0	0	unch	0	unch	–	–	–
	RFMC[5:0]	all “0”	all “0”	unch	all “0”	unch	–	–	–
RFPCTRk	**RFPC[7:0]**	–	–	–	–	–	–	–	–

(Note is listed at the end of the table.) “unch”: un-changed, “und”: undefined

Table 19-15. Register Behavior in Global / Channel Modes (4/8)

Register Symbol	Bit Symbol	Initial Value (after transition, in the state: Initialized value is kept in the state)							
		MCU_ RESET	SW_ RESET	GL_SLEEP	GL_RESET	GL_HALT	CH_SLEEP	CH_RESET	CH_HALT
CFCC	**CFE**	0	0	unch	0	unch	unch	TX FIFO: 0, RX FIFO: unch	unch
	CFRXIE	0	0	unch	unch	unch	unch	unch	unch
	CFTXIE	0	0	unch	unch	unch	unch	unch	unch
	CFPLS[2:0]	000B	000B	unch	unch	unch	–	–	–
	CFM	0	0	unch	unch	unch	unch	unch	unch
	CFITSS	0	0	unch	unch	unch	unch	unch	unch
	CFITR	0	0	unch	unch	unch	unch	unch	unch
	CFIM	0	0	unch	unch	unch	unch	unch	unch
	CFIGCV[2:0]	000B	000B	unch	unch	unch	unch	unch	unch
	CFTML[1:0]	00B	00B	unch	unch	unch	unch	unch	unch
	CFDC[2:0]	000B	000B	unch	unch	unch	unch	unch	unch
	CFITT[7:0]	00H	00H	unch	unch	unch	unch	unch	unch
CFSTS	**CFEMP**	1	1	unch	1	unch	–	TX FIFO: 1, RX FIFO: unch	–
	CFFLL	0	0	unch	0	unch	–	TX FIFO: 0, RX FIFO: unch	–
	CFMLT	0	0	unch	0	unch	–		–
	CFRXIF	0	0	unch	0	unch	–		–
	CFTXIF	0	0	unch	0	unch	–		–
	CFMC[5:0]	all “0”	all “0”	unch	all “0”	unch	–	TX FIFO: all “0”, RX FIFO: unch	–
CFPCTR	**CFPC[7:0]**	–	–	–	–	–	–	–	–
FESTS	**RF0EMP**	1	1	unch	1	unch	–	–	–
	RF1EMP	1	1	unch	1	unch	–	–	–
	CFEMP	1	1	unch	1	unch	–	Mirror of COM FIFO status	–
FFSTS	**RF0FLL**	0	0	unch	0	unch	–	–	–
	RF1FLL	0	0	unch	0	unch	–	–	–
	CFFLL	0	0	unch	0	unch	–	Mirror of COM FIFO status	–
FMSTS	**RF0MLT**	0	0	unch	0	unch	–	–	–
	RF1MLT	0	0	unch	0	unch	–	–	–
	CFMLT	0	0	unch	0	unch	–	–	–
RFISTS	**RF0IF**	0	0	unch	0	unch	–	–	–
	RF1IF	0	0	unch	0	unch	–	–	–
TMCm	**TMTR**	0	0	unch	0	unch	unch	0	unch
	TMTAR	0	0	unch	0	unch	unch	0	unch
	TMOM	0	0	unch	0	unch	unch	0	unch
TMSTSm	**TMTSTS**	0	0	unch	0	0	unch	0	0
	TMTRF[1:0]	00B	00B	unch	00B	unch	unch	00B	unch
	TMTRM	0	0	unch	0	unch	unch	0	unch
	TMTARM	0	0	unch	0	unch	unch	0	unch

(Note is listed at the end of the table.) “unch”: un-changed, “und”: undefined

Table 19-15. Register Behavior in Global / Channel Modes (5/8)

Register Symbol	Bit Symbol	Initial Value (after transition, in the state: Initialized value is kept in the state)							
		MCU_ RESET	SW_ RESET	GL_SLEEP	GL_RESET	GL_HALT	CH_SLEEP	CH_RESET	CH_HALT
TMTRSTS	**TMTRSTS[3:0]**	0000B	0000B	unch	0000B	unch	unch	Partially cleared by channel reset	unch
TMTARSTS	**TMTARSTS[3:0]**	0000B	0000B	unch	0000B	unch	unch		unch
TMTCSTS	**TMTCSTS[3:0]**	0000B	0000B	unch	0000B	unch	unch		unch
TMTASTS	**TMTASTS[3:0]**	0000B	0000B	unch	0000B	unch	unch		unch
TMIEC	**TMIE[3:0]**	0000B	0000B	unch	unch	unch	unch	unch	unch
THLCC	**THLE**	0	0	unch	0	unch	unch	0	unch
	THLIE	0	0	unch	unch	unch	unch	unch	unch
	THLIM	0	0	unch	unch	unch	unch	unch	unch
	THLDTE	0	0	unch	unch	unch	unch	unch	unch
THLSTS	**THLEMP**	1	1	unch	1	unch	unch	1	unch
	THLFLL	0	0	unch	0	unch	unch	0	unch
	THLELT	0	0	unch	0	unch	unch	0	unch
	THLIF	0	0	unch	0	unch	unch	0	unch
	THLMC[3:0]	0000B	0000B	unch	0000B	unch	–	0000B	–
THLPCTR	**THLPC[7:0]**	–	–	–	–	–	–	–	–
GTINTSTS	**TSIF**	0	0	unch	0	unch	unch	0	unch
	TAIF	0	0	unch	0	unch	unch	0	unch
	CFTIF	0	0	unch	0	unch	unch	0	unch
	THIF	0	0	unch	0	unch	unch	0	unch
GTSTCFG	**RTMPS[3:0]**	0000B	0000B	unch	0000B	unch	–	–	–
GTSTCTR	**RTME**	0	0	unch	0	unch	–	–	–
GFDCFG	**RPED**	0	0	unch	unch	unch	–	–	–
	TSCCFG[1:0]	00B	00B	unch	unch	unch	–	–	–
GLOCKK	**LOCK[15:0]**	–	–	–	–	–	–	–	–
GAFLIGENT	**IRN[3:0]**	0000B	0000B	unch	unch	unch	–	–	–
GAFLIGCTR	**IREN**	0	0	unch	unch	unch	–	–	–
	KEY[7:0]	–	–	–	–	–	–	–	–
GPFLECTR	**PFLDAE**	0	0	unch	unch	unch	–	–	–
GPFLCFG	**RNC[1:0]**	00B	00B	unch	unch	unch	–	–	–
GRSTC	**SRST**	0	unch	unch	unch	unch	–	–	–
	KEY[7:0]	–	–	–	–	–	–	–	–
CFDGRWC	**RPAGE[1:0]**	00B	00B	unch	unch	unch	–	–	–
	KEY[7:0]	–	–	–	–	–	–	–	–
CsDCFG	**DBRP[7:0]**	00H	00H	unch	unch	unch	unch	unch	unch
	DTSEG1[4:0]	all “0”	all “0”	unch	unch	unch	unch	unch	unch
	DTSEG2[3:0]	0000B	0000B	unch	unch	unch	unch	unch	unch
	DSJW[3:0]	0000B	0000B	unch	unch	unch	unch	unch	unch
CsFDCFG	**EOCCFG[2:0]**	000B	000B	unch	unch	unch	unch	unch	unch
	TDCOC	0	0	unch	unch	unch	unch	unch	unch
	TDCE	0	0	unch	unch	unch	unch	unch	unch
	ESIC	0	0	unch	unch	unch	unch	unch	unch
	RPNMD[1:0]	00B	00B	unch	unch	unch	unch	unch	unch
	TDCO[7:0]	00H	00H	unch	unch	unch	unch	unch	unch
	FDOE	0	0	unch	unch	unch	unch	unch	unch
	REFE	0	0	unch	unch	unch	unch	unch	unch
	CLOE	0	0	unch	unch	unch	unch	unch	unch

(Note is listed at the end of the table.)

“unch”: un-changed, “und”: undefined

Table 19-15. Register Behavior in Global / Channel Modes (6/8)

Register Symbol	Bit Symbol	Initial Value (after transition, in the state: Initialized value is kept in the state)							
		MCU_ RESET	SW_ RESET	GL_SLEEP	GL_RESET	GL_HALT	CH_SLEEP	CH_RESET	CH_HALT
CsFDCTR	**EOCCLR**	0	0	unch	0	unch	unch	0	unch
	SOCCLR	0	0	unch	0	unch	unch	0	unch
	PNMDC[1:0]	00B	00B	unch	00B	unch	unch	00B	unch
	KEY[7:0]	–	–	–	–	–	–	–	–
CsFDSTS	**TDCR[7:0]**	00H	00H	unch	00H	unch	unch	00H	unch
	EOCO	0	0	unch	0	unch	unch	0	unch
	SOCO	0	0	unch	0	unch	unch	0	unch
	PNSTS[1:0]	00B	00B	unch	00B	unch	unch	00B	unch
	TDCVF	0	0	unch	0	unch	unch	0	unch
	EOC[7:0]	00H	00H	unch	00H	unch	unch	00H	unch
	SOC[7:0]	00H	00H	unch	00H	unch	unch	00H	unch
CsFDCRC	**CRCREG[20:0]**	all “0”	all “0”	unch	all “0”	unch	unch	all “0”	unch
	SCNT[3:0]	0000B	0000B	unch	0000B	unch	unch	0000B	unch
GAFLIDi	**GAFLID[28:0]**	und	unch	unch	unch	unch	unch	unch	unch
	GAFLLB	und	unch	unch	unch	unch	unch	unch	unch
	GAFLRTR	und	unch	unch	unch	unch	unch	unch	unch
	GAFLIDE	und	unch	unch	unch	unch	unch	unch	unch
GAFLMi	**GAFLIDM[28:0]**	und	unch	unch	unch	unch	unch	unch	unch
	GAFLIFL1	und	unch	unch	unch	unch	unch	unch	unch
	GAFLRTRM	und	unch	unch	unch	unch	unch	unch	unch
	GAFLIDEM	und	unch	unch	unch	unch	unch	unch	unch
GAFLP0i	**GAFLDLC[3:0]**	und	unch	unch	unch	unch	unch	unch	unch
	GAFLIFL0	und	unch	unch	unch	unch	unch	unch	unch
	GAFLRMDP[4:0]	und	unch	unch	unch	unch	unch	unch	unch
	GAFLRMV	und	unch	unch	unch	unch	unch	unch	unch
	GAFLPTR[15:0]	und	unch	unch	unch	unch	unch	unch	unch
GAFLP1i	**GAFLFDP0**	und	unch	unch	unch	unch	unch	unch	unch
	GAFLFDP1	und	unch	unch	unch	unch	unch	unch	unch
	GAFLFDP8	und	unch	unch	unch	unch	unch	unch	unch
GPFLIDj	**GPFLID[28:0]**	und	unch	unch	unch	unch	unch	unch	unch
	GPFLLB	und	unch	unch	unch	unch	unch	unch	unch
	GPFLRTR	und	unch	unch	unch	unch	unch	unch	unch
	GPFLIDE	und	unch	unch	unch	unch	unch	unch	unch
GPFLMj	**GPFLIDM[28:0]**	und	unch	unch	unch	unch	unch	unch	unch
	GPFLIFL1	und	unch	unch	unch	unch	unch	unch	unch
	GPFLRTRM	und	unch	unch	unch	unch	unch	unch	unch
	GPFLIDEM	und	unch	unch	unch	unch	unch	unch	unch
GPFLP0j	**GPFLDLC[3:0]**	und	unch	unch	unch	unch	unch	unch	unch
	GPFLIFL0	und	unch	unch	unch	unch	unch	unch	unch
	GPFLRMDP[4:0]	und	unch	unch	unch	unch	unch	unch	unch
	GPFLRMV	und	unch	unch	unch	unch	unch	unch	unch
	GPFLPTR[15:0]	und	unch	unch	unch	unch	unch	unch	unch
GPFLP1j	**GPFLFDP0**	und	unch	unch	unch	unch	unch	unch	unch
	GPFLFDP1	und	unch	unch	unch	unch	unch	unch	unch
	GPFLFDP8	und	unch	unch	unch	unch	unch	unch	unch

(Note is listed at the end of the table.) “unch”: un-changed, “und”: undefined

Table 19-15. Register Behavior in Global / Channel Modes (7/8)

Register Symbol	Bit Symbol	Initial Value (after transition, in the state: Initialized value is kept in the state)							
		MCU_ RESET	SW_ RESET	GL_SLEEP	GL_RESET	GL_HALT	CH_SLEEP	CH_RESET	CH_HALT
GPFLPTj	**GPFLOFFSET1[3:0]**	und	unch	unch	unch	unch	unch	unch	unch
	GPFLOUT1	und	unch	unch	unch	unch	unch	unch	unch
	GPFLRANG1	und	unch	unch	unch	unch	unch	unch	unch
	GPFLOFFSET0[3:0]	und	unch	unch	unch	unch	unch	unch	unch
	GPFLOUT0	und	unch	unch	unch	unch	unch	unch	unch
	GPFLRANG0	und	unch	unch	unch	unch	unch	unch	unch
	GPFLANDOR	und	unch	unch	unch	unch	unch	unch	unch
GPFLPD0j	**FDATA0[31:0]**	und	unch	unch	unch	unch	unch	unch	unch
GPFLPM0j	**FMASK0[31:0]**	und	unch	unch	unch	unch	unch	unch	unch
GPFLPD1j	**FDATA1[31:0]**	und	unch	unch	unch	unch	unch	unch	unch
GPFLPM1j	**FMASK1[31:0]**	und	unch	unch	unch	unch	unch	unch	unch
RPGACCr	**RDTA[31:0]**	und	unch	unch	unch	unch	unch	unch	unch
RMIDn	**RMID[28:0]**	und	unch	unch	unch	unch	unch	unch	unch
	RMRTR	und	unch	unch	unch	unch	unch	unch	unch
	RMIDE	und	unch	unch	unch	unch	unch	unch	unch
RMPTRn	**RMTS[15:0]**	und	unch	unch	unch	unch	unch	unch	unch
	RMDLC[3:0]	und	unch	unch	unch	unch	unch	unch	unch
RMSTSn	**RMESI**	und	unch	unch	unch	unch	unch	unch	unch
	RMBRS	und	unch	unch	unch	unch	unch	unch	unch
	RMFDF	und	unch	unch	unch	unch	unch	unch	unch
	RMIFL[1:0]	und	unch	unch	unch	unch	unch	unch	unch
	RMPTR[15:0]	und	unch	unch	unch	unch	unch	unch	unch
RMDn_p	**RMDBq[7:0]**	und	unch	unch	unch	unch	unch	unch	unch
	RMDBq[7:0]	und	unch	unch	unch	unch	unch	unch	unch
	RMDBq[7:0]	und	unch	unch	unch	unch	unch	unch	unch
	RMDBq[7:0]	und	unch	unch	unch	unch	unch	unch	unch
RFIDk	**RFID[28:0]**	und	unch	unch	unch	unch	unch	unch	unch
	RFRTR	und	unch	unch	unch	unch	unch	unch	unch
	RFIDE	und	unch	unch	unch	unch	unch	unch	unch
RFPTRk	**RFTS[15:0]**	und	unch	unch	unch	unch	unch	unch	unch
	RFDLC[3:0]	und	unch	unch	unch	unch	unch	unch	unch
RFFDSTSk	**RFESI**	und	unch	unch	unch	unch	unch	unch	unch
	RFBRS	und	unch	unch	unch	unch	unch	unch	unch
	RFFDF	und	unch	unch	unch	unch	unch	unch	unch
	RFIFL[1:0]	und	unch	unch	unch	unch	unch	unch	unch
	RFPTR[15:0]	und	unch	unch	unch	unch	unch	unch	unch
RFDFk_p	**RFDBq[7:0]**	und	unch	unch	unch	unch	unch	unch	unch
	RFDBq[7:0]	und	unch	unch	unch	unch	unch	unch	unch
	RFDBq[7:0]	und	unch	unch	unch	unch	unch	unch	unch
	RFDBq[7:0]	und	unch	unch	unch	unch	unch	unch	unch
CFID	**CFID[28:0]**	und	unch	unch	unch	unch	unch	unch	unch
	THLEN	und	unch	unch	unch	unch	unch	unch	unch
	CFRTR	und	unch	unch	unch	unch	unch	unch	unch
	CFIDE	und	unch	unch	unch	unch	unch	unch	unch
CFPTR	**CFTS[15:0]**	und	unch	unch	unch	unch	unch	unch	unch
	CFDLC[3:0]	und	unch	unch	unch	unch	unch	unch	unch

(Note is listed at the end of the table.) “unch”: un-changed, “und”: undefined

Table 19-15. Register Behavior in Global / Channel Modes (8/8)

Register Symbol	Bit Symbol	Initial Value (after transition, in the state: Initialized value is kept in the state)							
		MCU_ RESET	SW_ RESET	GL_SLEEP	GL_RESET	GL_HALT	CH_SLEEP	CH_RESET	CH_HALT
CFFDCSTS	**CFESI**	und	unch	unch	unch	unch	unch	unch	unch
	CFBRS	und	unch	unch	unch	unch	unch	unch	unch
	CFFDF	und	unch	unch	unch	unch	unch	unch	unch
	CFIFL[1:0]	und	unch	unch	unch	unch	unch	unch	unch
	CFPTR[15:0]	und	unch	unch	unch	unch	unch	unch	unch
CFDFp	**CFDBq[7:0]**	und	unch	unch	unch	unch	unch	unch	unch
	CFDBq[7:0]	und	unch	unch	unch	unch	unch	unch	unch
	CFDBq[7:0]	und	unch	unch	unch	unch	unch	unch	unch
	CFDBq[7:0]	und	unch	unch	unch	unch	unch	unch	unch
TMIDm	**TMID[28:0]**	und	unch	unch	unch	unch	unch	unch	unch
	THLEN	und	unch	unch	unch	unch	unch	unch	unch
	TMRTR	und	unch	unch	unch	unch	unch	unch	unch
	TMIDE	und	unch	unch	unch	unch	unch	unch	unch
TMPTRm	**TMDLC[3:0]**	und	unch	unch	unch	unch	unch	unch	unch
TMFDCTRm	**TMESI**	und	unch	unch	unch	unch	unch	unch	unch
	TMBRS	und	unch	unch	unch	unch	unch	unch	unch
	TMFDF	und	unch	unch	unch	unch	unch	unch	unch
	TMIFL[1:0]	und	unch	unch	unch	unch	unch	unch	unch
	TMPTR[15:0]	und	unch	unch	unch	unch	unch	unch	unch
TMDFm_p	**TMDBq[7:0]**	und	unch	unch	unch	unch	unch	unch	unch
	TMDBq[7:0]	und	unch	unch	unch	unch	unch	unch	unch
	TMDBq[7:0]	und	unch	unch	unch	unch	unch	unch	unch
	TMDBq[7:0]	und	unch	unch	unch	unch	unch	unch	unch
THLACC0	**BT[1:0]**	und	unch	unch	unch	unch	unch	unch	unch
	BN[1:0]	und	unch	unch	unch	unch	unch	unch	unch
	TMTS[15:0]	und	unch	unch	unch	unch	unch	unch	unch
THLACC1	**TID[15:0]**	und	unch	unch	unch	unch	unch	unch	unch
	TIFL[1:0]	und	unch	unch	unch	unch	unch	unch	unch

“unch”: un-changed, “und”: undefined

Note After HW reset, the **GSTSs.GRAMINIT** bit is set to indicated that the RS-CANFD lite module is initializing the RAM. This bit is cleared automatically when the RAM initialization is completed and will not set again until H/W reset is activated.

19.7 Acceptance Filtering Function using Global Acceptance Filter List (AFL)

19.7.1 Overview

The RS-CANFD lite module can handle message acceptance filtering for all channels with a global Acceptance Filter List (called AFL). Each element of the AFL defines a filter rule for messages received on a specific channel.

Following actions will be done based on the AFL entries:

- Acceptance filtering based on received CAN Identifier and masking.
- DLC filtering based on received DLC value.
- Message data payload according to the **GCFG.CMPOC** bit.
- Storage of accepted messages in the Message Buffer objects defined in the related AFL entry.
- Attaching a 16 bits pointer to the stored messages defined in the related AFL entry e.g. to support AUTOSAR applications.
- Attaching a 2 bits information label to the stored messages defined in the related AFL entry.

The RS-CANFD lite module allows a maximum of 16 AFL entries.

During acceptance filtering process, each AFL entry in a channel is checked against the received message by the acceptance filter unit. Check is starting from the lowest AFL entry number for this channel.

AFL search is stopped when a match of the received Identifier with a configured Identifier/Mask combination occurs or when the received Identifier has been compared against all AFL entries defined for the related channel. If no match occurs, then the received message is rejected. No notification is given to the application in this case.

Additionally, an automatic DLC filtering is performed for each accepted message if DLC Check is globally enabled. If the DLC value of the received message is equal to or higher than the configured DLC value in the matching AFL entry, then the DLC check is passed.

If DLC replacement (**GCFG.DRE** bit) is enabled, DLC value configured in the matching AFL entry is greater than 0H and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination RXMB or FIFO Buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional Data Bytes received on the CAN Bus are not stored in the destination RXMB or FIFO Buffer. These additional Data Bytes will be stored as 0H in the destination RXMB or FIFO Buffer.

If DLC replacement is enabled and DLC value of matching AFL entry is 0H, then the received value of DLC will be stored in the destination RX MB or FIFO Buffer.

If DLC replacement (**GCFG.DRE** bit) is disabled and DLC check passes, then the received value of DLC on the CAN Bus is stored in the destination RXMB or FIFO Buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional Data Bytes received from the CAN Bus are also stored in the destination RXMB or FIFO Buffer.

If DLC value of the received message is less than the configured DLC value in the matching AFL entry, then DLC check fails. In this case, the received message is rejected and it is not stored in any RXMB or FIFO Buffer.

Additionally, DLC check failure will be flagged by the DLC Error Flag in the Global Error Flag Register.

If configured, an error interrupt will also be generated.

The DLC replacement configuration has no impact if the DLC check fails.

If a message has passed both acceptance filtering and DLC filtering, it is stored in a single reception Message Buffer and/or in FIFO buffers configured for reception function.

This message storage target information is also defined in the same AFL entry. Do not set a target at the AFL entry which is not configured.

Each accepted received message could be stored into a maximum of 2 different target destinations (single reception Message Buffer and/or FIFO buffers).

(Programming of more than 2 target destinations are not allowed. In the case more destinations are programmed then internal timing race condition can occur and received message may not be stored to the Message RAM. Correct configuration of the numbers of Target destination is the responsibility of the application.)

Further protection mechanism is made for the case when a received message contains more data payload Bytes than possible to store in the target destination (**RMNB.RMPLS**, **RFCCk.RFPLS** or **CFCC.CFPLS**).

If the **GCFG.CMPOC** = 0 then the message is completely rejected and will not be stored in the target destination. In the case of **GCFG.CMPOC** = 0 and RX or Common FIFO full and the received message contains more data payload Bytes than possible to store in the target destination (**RMNB.RMPLS**, **RFCCk.RFPLS** or **CFCC.CFPLS**), the corresponding **FMSTS.RFxMLT** (x=0,1) or **FMSTS.CFMLT** will not be set to 1 respectively.

If the **GCFG.CMPOC** = 1 then the received Data Bytes greater than **RMNB.RMPLS** will be rejected. In the case of **GCFG.CMPOC** = 1 and RX or Common FIFO full and the received message contains more data payload Bytes than possible to store in the target destination (**RMNB.RMPLS**, **RFCCk.RFPLS** or **CFCC.CFPLS**), the corresponding **FMSTS.RFxMLT** (x=0,1) or **FMSTS.CFMLT** will be set to 1 respectively.

Depending on the **GCFG.DRE** the original received DLC or the DLC value configured at the AFL entry will be stored.

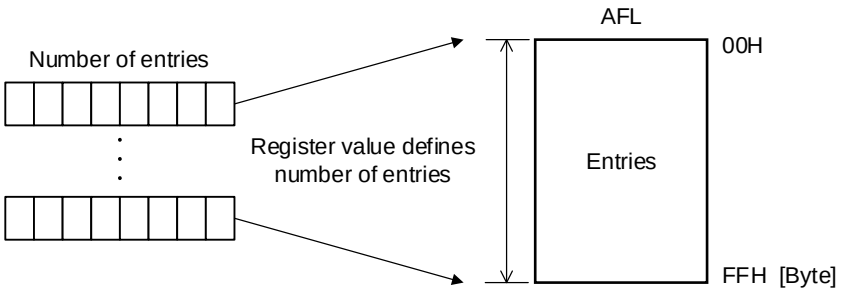
Regardless of **GCFG.CMPOC** configuration, **GERFL.CMPOF** will be set to 1 if payload overflow condition is detected.

The DLC filtering is performed before the payload overflow function. Hence for one reception frame only one flag could be set at the same time **GERFL.DEF** or **GERFL.CMPOF.**

19.7.2 Allocation of AFL Entries

The number of AFL entries can be configured using the dedicated field in the related Global Acceptance Filter Configuration Registers (See figure below).

Figure 19-17. Configuration of AFL



The minimum number of entries is 0 (no entries defined), the maximum number of entries is 16.

All entries are unique for a channel and overlapping or sharing of entries is not supported. Correct configuration of the AFL is the responsibility of the application.

The RS-CANFD lite module will not flag errors related to the configuration of the AFL.

19.7.3 AFL Entry Description

Each AFL entry consists of 16 Bytes. The fields in all entries are identical.

Each entry contains the following information for acceptance filtering and DLC filtering:

- ID bit (11 bits for Standard Frame Format, 29 bits for Extended Frame Format):

  Acceptance filter unit will check Identifier field of received message against Identifier field of each AFL entry (full 29 bits masking of Identifier bits possible, see below).
- IDE bit:

  Acceptance filter unit will check IDE bit of received message against this bit and selects the relevant part of the Identifier field for acceptance filtering (masking of IDE bit possible, see below).
- RTR bit:

  Acceptance filter unit will only accept Data Frames (RTR = 0) or Remote Frames (RTR = 1) according to the setting of this bit (masking of RTR bit possible, see below).
- Loopback configuration bit:

  This bit can enable/disable the AFL entry depending on the Loopback Configuration or Mirror Mode condition.
- Mask for Identifier bits (29 bits):

  Each bit in the Identifier Mask can mask the corresponding Identifier bit in the AFL entry during acceptance filtering (See **Figure 19-18**).
- Mask for IDE bit:

  If this Mask bit masks the IDE bit of the AFL entry both Standard Identifier and Extended Identifier format messages can be accepted by this AFL entry. The identifier of the received message is compared against the Standard Identifier part of the AFL entry for Standard Identifier format messages and against the Extended Identifier part of the AFL entry for Extended Identifier format messages.
- Mask for RTR bit:

  If this Mask bit masks the RTR bit of the AFL entry both frame formats ‘Data Frame’ and ‘Remote Frame’ will be accepted by this AFL entry.
- Pointer information (16 bits):

  This 16 bits pointer will be attached to a received message accepted by the related AFL entry. The Pointer will be added during message storage in the Message Buffer area and can be used by application as support function. The pointer information could be used for example to support PDU Identifier allocation for the received message in AUTOSAR systems.
- Information Label (2 bits):

  This 2 bits label will be attached to a received message accepted by the related AFL entry. The label will be added during message storage in the Message Buffer area and can be used by application as support function.
- DLC value for automatic DLC filtering:

  If the DLC value of the received message is equal or higher than the configured DLC value the DLC check is passed. If the DLC value in this AFL entry is configured to 0 DLC filtering is effectively disabled for this entry (all accepted messages will pass DLC filtering).

Each AFL entry contains the following information for the handling of received messages.

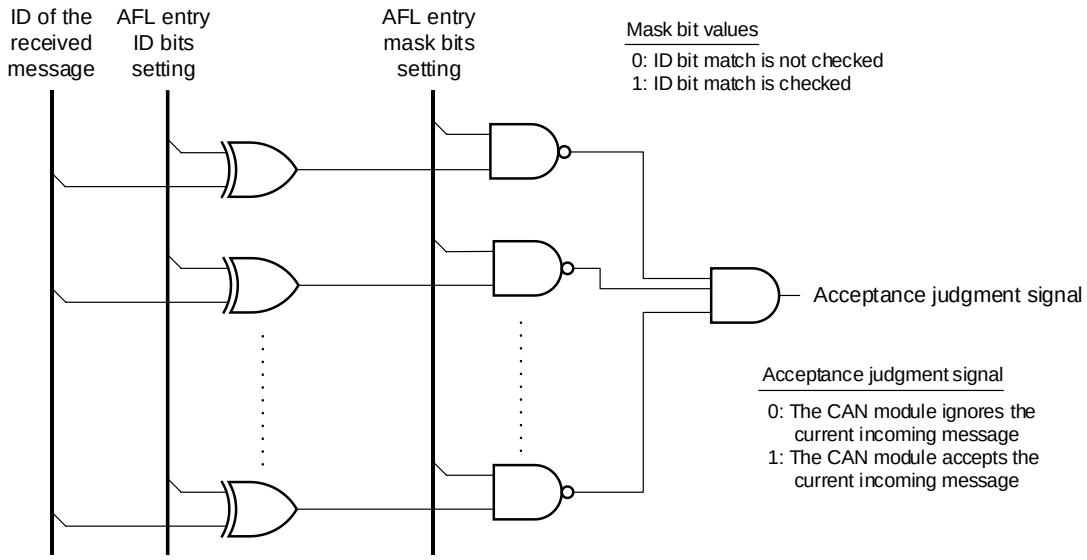
Message Buffer number of one single reception Message Buffer as target for received message storage.

Single reception Message Buffer enable bit to configure the single reception Message Buffer number as valid/invalid as target for received message storage.

FIFO direction pointer: each bit of the FIFO direction pointer configures a dedicated FIFO as possible target for a received message.

There is no hardware protection against such storage of message. Hence the FIFO direction pointer should be configured carefully.

Figure 19-18. Acceptance Filter Function



19.7.4 Entering Entries in the AFL

Application SW can enter one full entry into the AFL via following registers:

- Global AFL ID Entry Register: Part 1 of the AFL entry
- Global AFL Mask Entry Register: Part 2 of the AFL entry
- Global AFL Pointer 0 Entry Register: Part 3 of the AFL entry
- Global AFL Pointer 1 Entry Register: Part 4 of the AFL entry

16 sets of these registers form a group of AFL entries. Each group can be accessed via a page mechanism. The AFL should only be configured in CH_RESET or CH_HALT.

GAFLECTR register has following fields:

- **AFLDAE** (Acceptance Filter List Data Access Enable): 1 bit to enable / disable the AFL Data access to prevent unwanted write access to the AFL.

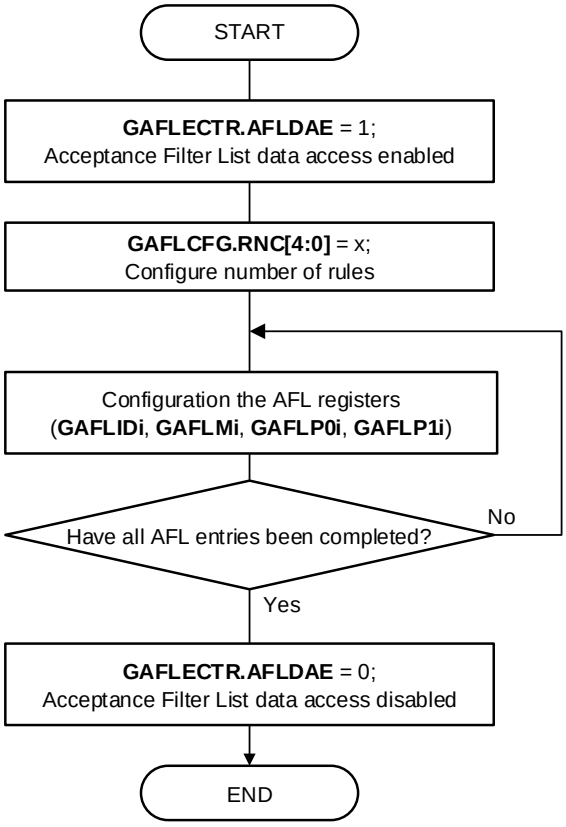
The configuration flow shown in **Figure 19-19** should be followed to program the AFL.

After entering all entries in Configuration Mode, locking of the AFL access should be performed to protect unwanted write access to the AFL.

Write protection is active during all global modes if the lock bit is set (GL_RESET, GL_HALT, and GL_OPERATION).

Read access to AFL is still possible during all global modes even if AFL data access is disabled (consistency check of AFL contents possible during runtime).

Figure 19-19. AFL Configuration Flow



19.7.5 Loopback Modes

If the Loopback Configuration bit is set, the AFL entry is only valid in loopback test mode (Self test mode 0 or Self test mode 1) or in mirror mode when receiving messages that were transmitted by the respective CAN channel itself.

The AFL entry is not valid for received messages in loopback mode transmitted by other CAN nodes on the bus (the expression ‘valid or invalid’ for the related entry means that this AFL entry ‘will or will not’ be compared against the received message ID respectively).

If the Loopback Configuration bit is 0 the AFL entry is only valid for

- Received messages transmitted by other CAN nodes on the bus in normal (non-loopback mode) and mirror mode.
- Received messages transmitted by other CAN nodes or the CAN channel itself in loopback test mode.

The mirror mode can be enabled via the **GCFG.MME** bit. If **GCFG.MME** bit is set, then a successfully Transmitted message can be stored back in an RX MB or FIFO Buffer if a matching Entry is configured in the AFL for that channel.

The Loopback configuration bit in the matching AFL Entry must be set for storing this Frame.

If mirror mode and loopback test mode are configured at the same time the loopback test mode behavior applies.

Table 19-16 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 19-16. Acceptance Filter Behavior Based on Loopback Configuration Setting in AFL Entry

GCFG.MME (Mirror Mode Enable)	**CsCTR.CTMS[1:0]** (Self-test Mode Selection)	Channel Mode	Loopback Configuration Bit in AFL Entry	AFL Entry Note
0	≠ 10B, 11B	Receiver	0	valid
			1	invalid
		Transmitter	0	invalid
			1	invalid
	= 10B, 11B	Receiver	0	valid
			1	invalid
		Transmitter	0	valid
			1	valid
1	≠ 10B, 11B	Receiver	0	valid
			1	invalid
		Transmitter	0	invalid
			1	valid
	= 10B, 11B	Receiver	0	valid
			1	invalid
		Transmitter	0	valid
			1	valid

Note The expression ‘valid’ or ‘invalid’ for the related entry means that this AFL entry ‘will or will not’ be compared against the received message ID respectively.

19.7.6 IDE Masking

When the **GAFLMi.GAFLIDEM** bit is 0 in an AFL entry, then the IDE bit configured in the AFL Entry is not considered for ID matching. In this case the decision of ID[10:0] or ID[28:0] matching is based on the received IDE bit.

Consider the following Example:

The ID & Mask fields of an AFL Entry “x” is configured as follows:

GAFLID [x] = C0553A20H → IDE = 1, RTR = 1, LLB = 0, ID[10:0] = 220H / ID[28:0] = 0553A20H

GAFLMi = 0000FFFFH → IDEM = 0. RTRM = 0, IDM[10:0] = 7FFH / IDM[28:0] = 0000FFFFH

The result of comparison of 4 different received Ids with AFL Entry X is described below:

- If a frame with IDE = 0 & ID = 220H is received, then this is considered as a match.
- If a frame with IDE = 0 & ID = 320H is received, then this is not a match.
- If a frame with IDE = 1 & ID = 1FFF3A20H is received, then this is considered as a match.
- If a frame with IDE = 1 & ID = 08803220H is received, then this is not a match.

19.7.7 Updating AFL Entry During Communication

AFL entries can be updated without disabling all CAN communications.

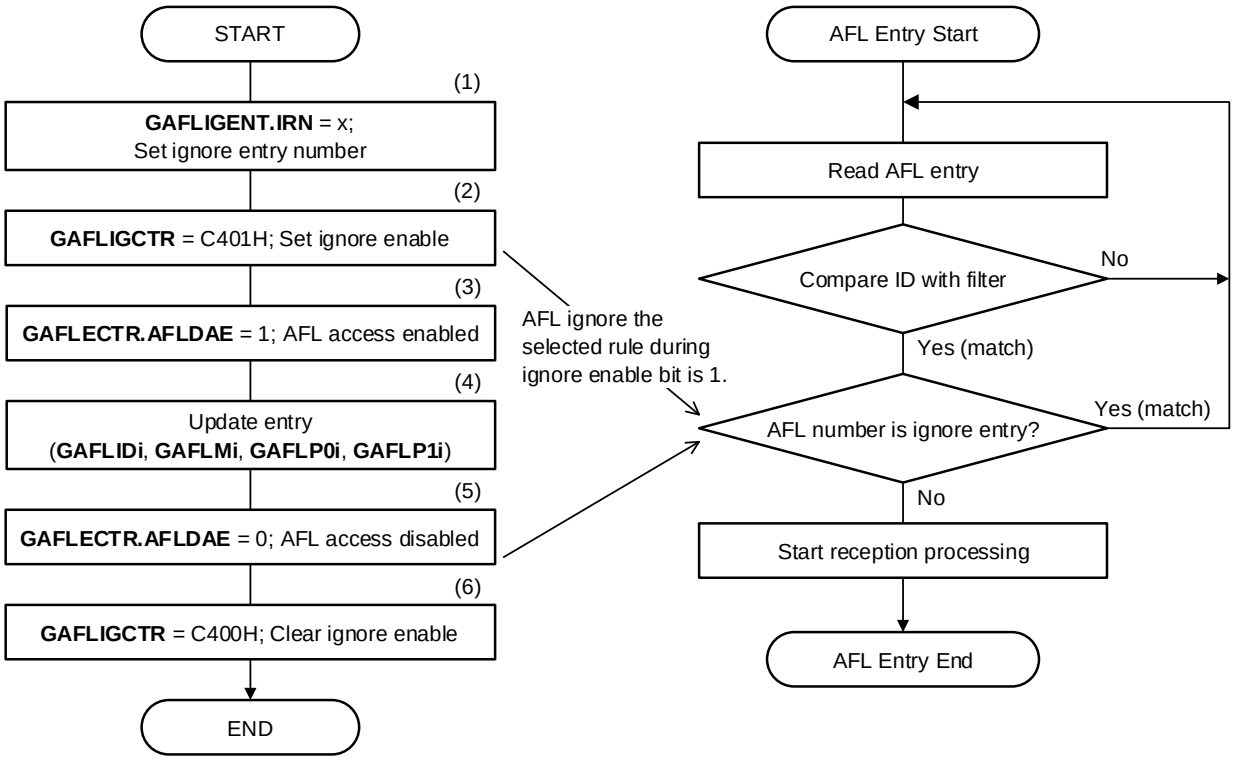
Select the entry number to be updated.

Set AFL entry number and ignore enable bit.

This entry number is ignored from AFL matching while updating the entry.

Figure below shows the AFL entry update flow.

Figure 19-20. AFL Entry Update Flow



The method of update of an AFL entry is shown below.

(1) Set entry number to **GAFLIGENT** register.
(2) Set the value C401H (key code & enable bit) to **GAFLIGCTR** register.
(3) **GAFLECTR.AFLDAE** is set to 1.
(4) Set the new rule to **GAFLIDi**, **GAFLMi**, **GAFLP0i**, and **GAFLP1i** registers.
(5) **GAFLECTR.AFLDAE** is cleared to 0.
(6) Set the value C400H (key code & clear enable bit) to **GAFLIGCTR** register.

(*) This entry number becomes ignoring for RXSCAN during the periods from (2) to (5).

AFL filter can be used to the range set **GAFLCFG**. An addition and deletion of an entry are possible in it. Therefore, it is necessary to set the maximum number to be used to **GAFLCFG**.

For example: case1

Delete one of the entry.

Current entry is follows. Delete the entry3.

		entry number		
total entry = 6	entry0	0	ID=050H	
	entry1	1	ID=051H	
	entry2	2	ID=052H	
	entry3	3	ID=053H	← delete rule
	entry4	4	ID=054H	
	entry5	5	ID=055H	

How to delete the entry

(1) Set 0003H to **GAFLIGENT** register.
(2) Set C401H to **GAFLIGCTR** register.
(3) Set 0100H to **GAFLECTR** register.
(4) Set same rule as previous rule by accessing to **GAFLIDi, GAFLMi, GAFLP0i, GAFLP1i**. (i = 3, this is entry3)
(5) Set 0000H to **GAFLECTR** register.
(6) Set C400H to **GAFLIGCTR** register.

Finish to delete the entry3. Current entry is follows.

		entry number		
total entry = 6	entry0	0	ID=050H	
	entry1	1	ID=051H	
	entry2	2	ID=052H	
	entry3	3	ID=052H	← set rule same as previous rule
	entry4	4	ID=054H	
	entry5	5	ID=055H	

		entry number		
total entry = 5 entry2 = entry3	entry0	0	ID=050H	
	entry1	1	ID=051H	
	entry2	2	ID=052H	
	entry3	3	ID=052H	← set rule same as previous rule
	entry4	4	ID=054H	
	entry5	5	ID=055H	

For example: case2

Add one of the entry.

Current entry is follows. New entry is added to entry3.

		entry number		
total entry = 6	entry0	0	ID=050H	
	entry1	1	ID=051H	
	entry2	2	ID=052H	
	entry3	3	ID=052H	← add new rule in this position
	entry4	4	ID=054H	
	entry5	5	ID=055H	

		entry number		
total entry = 5 entry2 = entry3	entry0	0	ID=050H	
	entry1	1	ID=051H	
	entry2	2	ID=052H	
	entry3	3	ID=052H	← add new rule in this position
	entry4	4	ID=054H	
	entry5	5	ID=055H	

How to add entry

(1) Set 0003H to **GAFLIGENT** register.
(2) Set C401H to **GAFLIGCTR** register.
(3) Set 0100H to **GAFLECTR** register.
(4) Set new rule by accessing to **GAFLIDi**, **GAFLMi**, **GAFLP0i**, **GAFLP1i**. (i = 3, this is entry3)
(5) Set 0000H to **GAFLECTR** register.
(6) Set C400H to **GAFLIGCTR** register.

Finish to add entry. Current entry is follows.

		entry number		
total entry = 6	entry0	0	ID=050H	
	entry1	1	ID=051H	
	entry2	2	ID=052H	
	entry3	3	ID=056H	← add new rule
	entry4	4	ID=054H	
	entry5	5	ID=055H	

19.8 FIFO Buffers and Normal MB Configuration

This Section describes the process for configuration of the number of RX Message Buffers, the FIFO Buffers and the Flat TX Message Buffers in the RS-CANFD lite module. The Message Buffers are mapped as shown in **Figure 19-21** below.

The RX Message Buffers can be accessed via RX Message Buffer Registers.

The RX FIFO buffers and the Common FIFO buffers configured in RX mode or TX mode can only be accessed via the FIFO Access Registers.

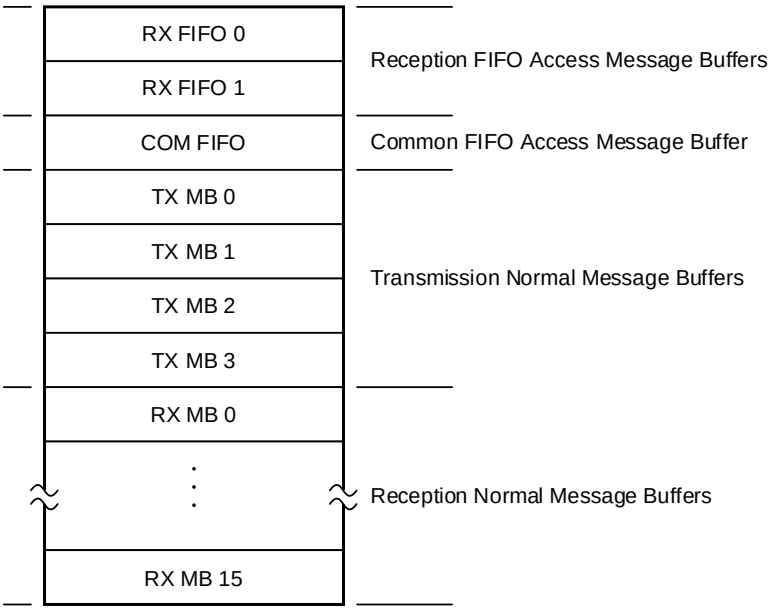
If the Common FIFO is configured in TX mode, then the data can only be written to the FIFO via FIFO Access registers.

If the Common FIFO is configured in RX mode, then the data can only be read from the FIFO Access Registers.

The TX Message Buffers can be accessed via the TX Message Buffer Registers.

When reading the unused Message Buffer, the MB locations are read as unknown values.

Figure 19-21. Message Buffer Configuration



19.8.1 Normal RX Message Buffers

In RS-CANFD lite module, the frames received can be stored in Normal RX Message Buffers based on the configuration of the AFL entries.

Additionally, the number of Normal RX Message Buffers required in the system can be chosen up to a fixed maximum limit.

19.8.1.1 Normal RX Message Buffer Configuration

In RS-CANFD lite module, the number of normal RX Message Buffers can be configured by writing to the RX Message Buffer Number Register.

The limiting values for the configuration of number of Message Buffers are:

Minimum Value = 00H (no Normal RX MB)

Maximum Value = 10H (16)

Do not use values outside these limits.

The AFL entries for routing the received messages to normal RX Message Buffers should be configured to match the requirements of the system.

The AFL entries should also be configured properly. An AFL entry for normal RX Message Buffers should not exceed the number of Message Buffers configured in the RX Message Buffer Number Register.

Note There is no internal check procedure provided in RS-CANFD lite module against wrong configuration of the AFL.

The data field size of the RX Message Buffer can be configured via the **RMNB.RMPLS**. The default size is 8 Bytes. The max data payload size is 64 Bytes.

In case the receiving frame exceeds the data field size then the acceptance depends on the configuration of the **GCFG.CMPOC** (message rejecting or data payload cut).

19.8.2 FIFO Buffers

The RS-CANFD lite module provides a fixed number of FIFO Buffers to support storage of frames for reception and transmission functions.

Number of reception-only FIFO Buffers is fixed to 2.

However, Common FIFO Buffer can be configured for storing messages for transmission or reception function.

These FIFO Buffers can be enabled or disabled and the size, Interrupt structure and Message Lost mechanism and Message overwrite mechanism of the FIFO Buffers, as well as the location of the TX FIFO can be configured to match the system requirements.

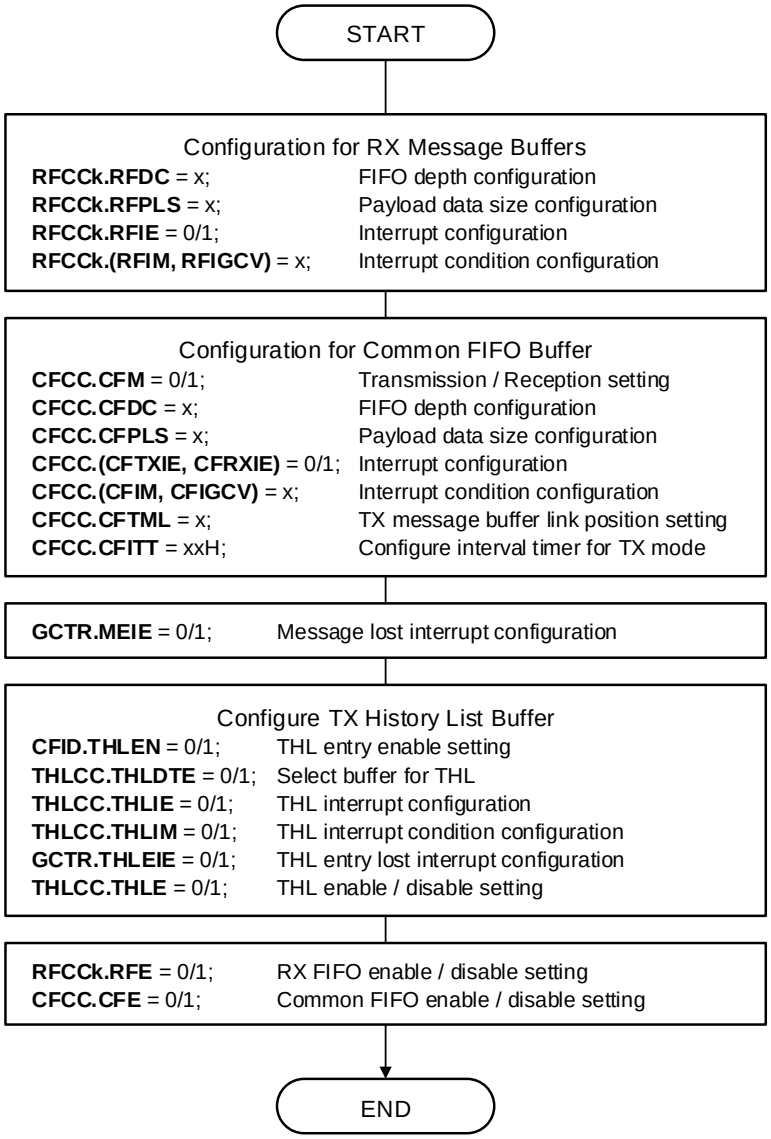
In case the receiving frame exceeds the data field size then the acceptance depends on the configuration of the **GCFG.CMPOC** (message rejecting or data payload cut).

19.8.2.1 FIFO Buffers Configuration

In RS-CANFD lite module, the FIFO Buffers can be configured to match the system requirements.

The total number of FIFO Buffers = 2 RX FIFO Buffers + 1 Common FIFO Buffer = 3 FIFO Buffers.

Figure 19-22. FIFO Buffer Configuration Flow in RS-CANFD lite Module



As shown in **Figure 19-22**, the various FIFO Buffers can be configured by writing to the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

For the 2 RX FIFO Buffers, the Interrupts, the FIFO depth and the FIFO payload data size can be configured.

For the Common FIFO Buffer, the mode, Interrupts FIFO depth, the FIFO payload data size and the FIFO TX link position can be configured.

19.8.2.1.1 FIFO Mode Configuration of Common FIFO Buffers

The mode of the Common FIFO Buffer can be configured by writing to the **CFCC.CFM** bits. The possible modes of configuration for Common FIFO Buffer are:

0: RX FIFO mode (default mode after CAN reset)
1: TX FIFO mode

Messages can only be read from the RX FIFO Buffers and the Common FIFO Buffer configured in RX Mode.

Messages are stored by the CAN module in these FIFO buffers based on the AFL entries.

Messages can be read and written into the Common FIFO Buffer configured in TX mode. These messages will be transmitted on the appropriate CAN channel.

The pointers can only be incremented when a new message is stored in the FIFO Buffer and decremented when a message is transmitted on the corresponding CAN channel by the RS-CANFD lite module.

After HW reset, the Common FIFO Buffer are configured in RX mode by default.

The FIFO buffers should be enabled only after configuring the Common FIFO buffer in the required modes.

19.8.2.1.2 FIFO TX-Message Buffer Link Configuration

When the Common FIFO is configured as TX FIFO, then the FIFO Buffer must be linked to a normal TX Message Buffer to participate in the transmission scan.

Do not write data into a TX Message Buffer that is linked to a Common FIFO buffer.

The TX Message Buffer link of each Common FIFO Buffer can be configured by writing to the **CFCC.CFTML** bits.

Available options for TX Message Buffer link configuration are:

00B: TX Message Buffer 0
01B: TX Message Buffer 1
10B: TX Message Buffer 2
11B: TX Message Buffer 3

19.8.2.1.3 FIFO Depth Configuration

The depth of each FIFO Buffer can be configured by writing to the **RFCCk.RFDC** bits and **CFCC.CFDC** bits. The 4 available options for depth configuration are:

000B: 0 Messages (FIFO Buffer cannot be enabled)
001B: 4 Messages
010B: 8 Messages
011B: 16 Messages
100B to 111B: Reserved

The RAM allocation for RX Message Buffers along with FIFO Buffers is limited to 16 messages with 64 Data Bytes. Configuration of the RX Message Buffers, along with FIFO Buffers, that exceeds this maximum limit should not be done.

RS-CANFD lite module logic will not check the validity of the configuration.

Caution If the FIFO depth of a Common FIFO is 4 messages or more (001B ≤ CFCC.CFDC[2:0]), then the common FIFO TX Message Buffer link is valid when the FIFO is disabled as well as enabled.
If the FIFO depth is 0 messages, then the common FIFO TX Message Buffer link is not valid when the FIFO is disabled as well as enabled.

19.8.2.1.4 FIFO Payload Size Configuration

The data size of each FIFO Buffer can be configured by writing to the **RFCCk.RFPLS** bits and **CFCC.CFPLS** bits. The 8 available options for size configuration are:

000B: 8 Bytes
001B: 12 Bytes
010B: 16 Bytes
011B: 20 Bytes
100B: 24 Bytes
101B: 32 Bytes
110B: 48 Bytes
111B: 64 Bytes

The RAM allocation for RX Message Buffers along with FIFO Buffers is limited to 16 messages with 64 Data Bytes. Configuration of the RX Message Buffers, along with FIFO Buffers, that exceeds this maximum limit should not be done.

RS-CANFD lite module logic will not check the validity of the configuration.

19.8.2.1.5 FIFO Interrupt Configuration

The Interrupt generation conditions for the FIFO Buffers can be configured by writing to the **RFCCk.RFIM** and **CFCC.CFIM** bit.

The 2 available options are:

0: RX FIFO Mode: Interrupt generated when Common FIFO counter reaches **RFCCk.RFIGCV**/**CFCC.CFIGCV** value from below values;
TX FIFO Mode: Interrupt generated when Common FIFO transmits last message successfully;

1: RX FIFO Mode: Interrupt generated at the end of storage of every received message;
TX FIFO Mode: Interrupt generated for every successfully transmitted message;

If the interrupt mode bit is 0 for a RX FIFO, then interrupt is generated based on the configuration of the **RFCCk.RFIGCV** bits.

Similarly, if the interrupt mode bit is 0 for a Common FIFO configured in RX mode, then interrupt is generated based on the configuration of **CFCC.CFIGCV** bits.

The 8 available options for configuring the FIFO counter value for generation of an interrupt are:

000B: Interrupt generated when FIFO is 1/8th Full
001B: Interrupt generated when FIFO is 1/4th Full
010B: Interrupt generated when FIFO is 3/8th Full
011B: Interrupt generated when FIFO is 1/2 Full
100B: Interrupt generated when FIFO is 5/8th Full
101B: Interrupt generated when FIFO is 3/4th Full
110B: Interrupt generated when FIFO is 7/8th Full
111B: Interrupt generated when FIFO is Full

In this case, an interrupt is generated when the Message Count matches the configured value.

However, there are some limitations on the configuration of the **RFCCk.RFIGCV** and **CFCC.CFIGCV** bits depending upon the **RFCCk.RFDC** and **CFCC.CFDC** bits (FIFO Depth Configuration) see **Table 19-17**.

Table 19-17. FIFO Interrupt Generation Counter vs. FIFO Depth Configuration

RFCCk.RFDC[2:0], CFCC.CFDC[2:0]	RFCCk.RFIGCV[2:0], CFCC.CFIGCV[2:0]							
	111B	110B	101B	100B	011B	010B	001B	000B
000B	don't care (FIFO cannot be enabled)							
001B	allowed	not allowed	allowed	not allowed	allowed	not allowed	allowed	not allowed
010B	allowed							
011B	allowed							

19.8.2.2 FIFO Buffers control

The FIFO Interrupt should be enabled by setting the **RFCCk.RFIE** and **CFCC.CFRXIE** or **CFCC.CFTXIE** bit.

After configuration is complete, each FIFO can be enabled by setting the **RFCCk.RFE** and **CFCC.CFE** bit to allow transmission and reception of messages.

19.9 Reception Function

In the RS-CANFD lite module, CAN messages received on any of the channels, will be stored in RX Message Buffers or in RX FIFO Buffers or Common FIFO Buffers configured in RX Mode depending upon the Acceptance Filter List entries:

- Up to 16 RX Message Buffers can be configured.
- 2 RX FIFO Buffers available.
- 1 Common FIFO Buffer can be configured in RX mode.

19.9.1 Message Storage in RX Message Buffers

When a message is successfully received and stored in a RX Message Buffer, the corresponding Newdata Flag is set in the RX Message Buffer Newdata Register.

The CAN Message can be read from the corresponding RX Message Buffer.

If a new message is stored into a RX Message Buffer before the previous message in this Message Buffer can be read, then the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the RX Message Buffer. If such a loss of messages is not acceptable, then RX FIFO should be used for storing related messages.

Cautions 1. When using interrupts, it is necessary to perform the same processing as the existing SW flow. See Figure 19-23.

2. Unused Data Bytes will be filled with 00H depending upon the DLC value.

Figure 19-23. RX Message Buffer Message Access Flow (Polling)

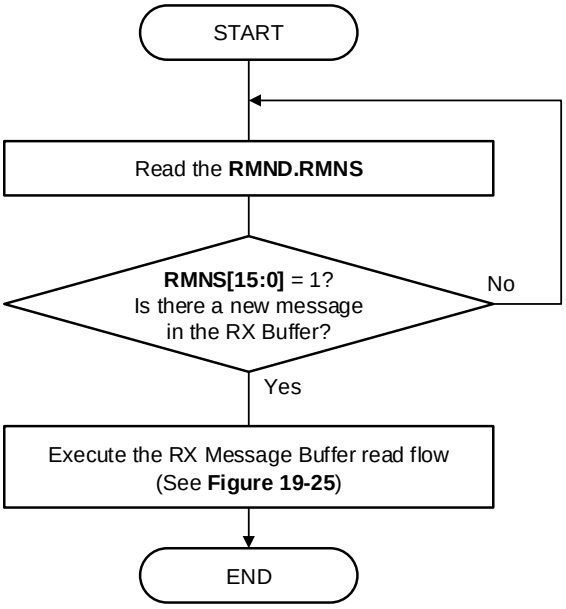


Figure 19-24. RX Message Buffer Message Access Flow (Interrupt)

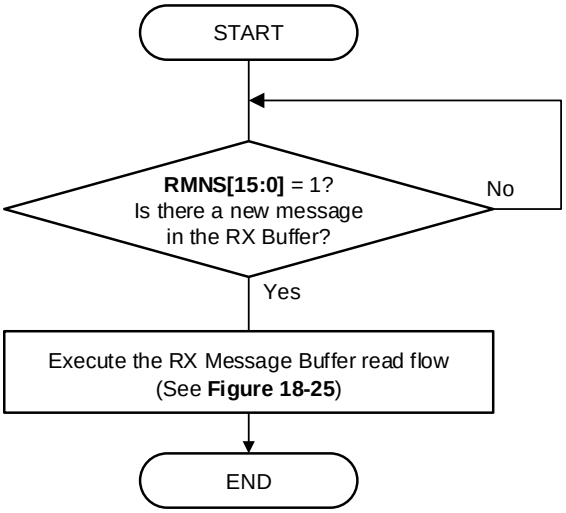
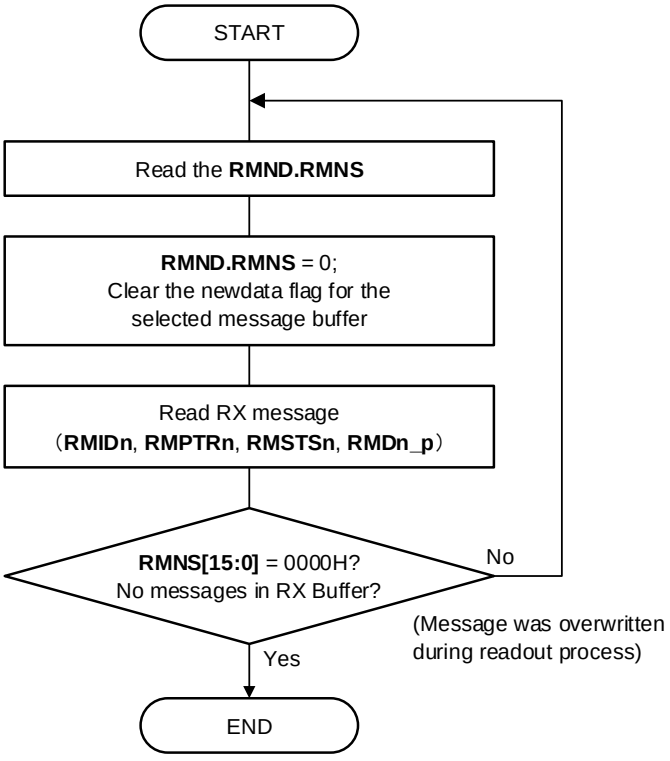


Figure 19-25. RX Message Buffer Read Flow



19.9.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to RX FIFO buffers or Common FIFO Buffer configured in RX Mode should be configured based on the requirements of the system.

The **GAFLP1i.GAFLFDP[8], [1:0]** field in the matching AFL entry selects the FIFO Buffers to which the related reception message will be stored.

When the received message is stored in one or more RX FIFO Buffers or Common FIFO Buffer configured in RX Mode, then the Message counter value is incremented in the corresponding RX FIFO Status Registers or Common FIFO Status Register.

Depending upon the configuration of the FIFO Buffers, an Interrupt may also be generated.

The message can be read from the corresponding FIFO access registers.

Caution Since many messages can be stored in the FIFO Buffers, reading more than 1 message may be required to read the latest message stored in a FIFO Buffer.

If the Message count value matches the FIFO depth, then the FIFO Full Flag is set.

When the value FFH is written to the corresponding FIFO Pointer Control Register, then the Message Count is decremented by 1.

Write only the value of FFH for FIFO Pointer Control Register after reading the complete message from the FIFO Access registers of the corresponding FIFO.

When all the messages stored in the FIFO are read, then the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO Message count matches the FIFO Depth (FIFO Full condition), the FIFO Message Lost flag is set and the new message will be lost (no overwrite of already stored messages will take place).

An appropriate value can be configured as warning level to generate an interrupt before the FIFO full condition occurs to avoid loss of a Message due to Overrun condition.

Caution Message lost can be set only in RX Mode by CAN side, the flag will not be set when the CPU side is overloading the FIFO buffers.

The RX FIFO Buffers and the Common FIFO Buffers configured in RX Mode can be disabled at any time by clearing the **RFCCk.RFE** or **CFCC.CFE** bit.

When the **RFCCk.RFE** or **CFCC.CFE** bit is cleared, then the message read and write pointers of the FIFO are cleared and are no longer active. Hence, all messages in the FIFO Buffers will be lost and no further messages can be stored into the FIFO.

Caution If the interrupt flag is set for a FIFO Buffer and then the FIFO is disabled, then the interrupt flag will not be cleared automatically. The interrupt flag should be cleared by SW before disabling the FIFO.

Figure 19-26. FIFO Buffer Message Access Flow (Example for Polling Case)

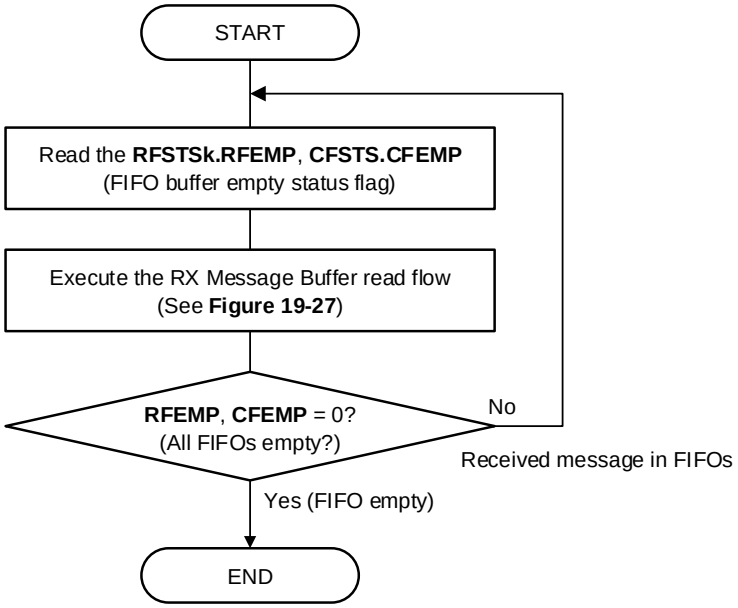
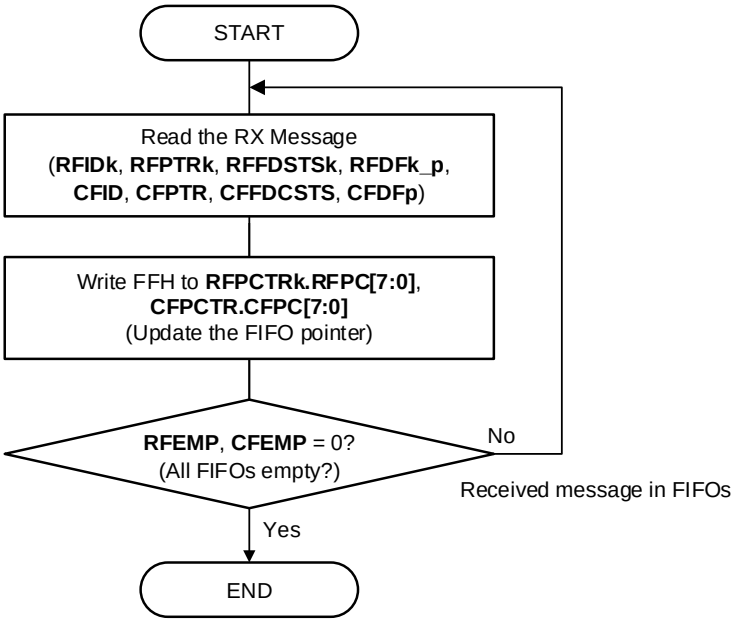


Figure 19-27. RX FIFO Buffer Read Flow (Example for Polling Case)



Caution When the next frame is received before clearing the completion interrupt flag of reception, the completion interrupt of reception is not set again. Check message reception within the process as shown in Figure 19-27.

Setting the interrupt request flag to 0 (no interrupt request present) clears the current interrupt request.

The next interrupt request is not generated until the interrupt request is cleared.

Even in the polling case, after checking that the receive FIFO buffer is empty, clear the interrupt request flag and check that the receive FIFO buffer is empty again.

19.9.3 Timestamp

The Timestamp counter is a free-running counter that can be used to check reception time of an incoming message or transmission time of successful transmitted messages. The Timestamp counter value will be captured based on the **GFDCFG.TSCCFG** configuration (at the sample point of Start of Frame, point in time when the frame is valid, or for CAN-FD frames also at the sample point of the res bit). For reception it is stored together with the message ID and Data into the target RX Message Buffer or RX FIFO.

For transmit message the Timestamp counter value will be stored as part of the TX History List entry.

The counter can be clocked with the peripheral clock or with the CAN channel bit timing clock. The counter source clock can be configured via the **GCFG.TSSS** bit. If it is 0, the peripheral clock is used. If it is 1, the selected CAN channel bit time clock is used.

The channel selection is done via the **GCFG.TSBTCS** Bit**.**

Care has to be taken when using selected CAN channel bit time clock as clock source. In case of entering Channel Halt Mode or Channel Reset Mode, for this channel, the Timestamp counter is stopped. So, also for other CAN channels the Timestamp counter value will not be updated.

If peripheral clock is selected as Timestamp counter clock source Channel Modes are not influencing the Timestamp counter function.

The source clock for the Timestamp counter can be divided by a factor defined by the **GCFG.TSP** bits.

The Timestamp counter can be reset to 0000H via the **GCTR.TSRST** bit.

19.10 Transmission Functions

There are several possible transmission configurations:

- Normal transmission
- FIFO transmission

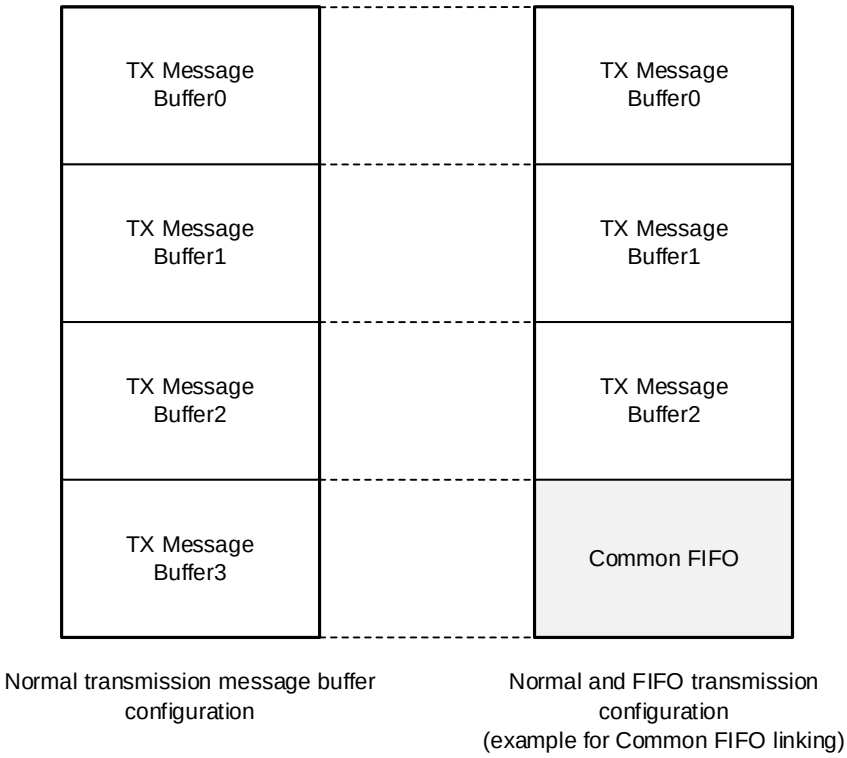
Common FIFO (TX mode): Common FIFO in TX mode is linked to a dedicated channel. Channel has a fixed number of 1 Common FIFO assigned to it. Within the channel, a Common FIFO configured in TX mode, can be freely linked (assigned) between 0 and 3 transmission Message Buffers (only one FIFO to one transmission Message Buffer).

The Common FIFO Buffer then replaces the transmission Message Buffer linked to it.

Transmission control and status registers of these transmission Message Buffers should not be used.

Refer to **Figure 19-28** for information about Common FIFO Buffer assignment to related channel.

Figure 19-28. Channel Transmission Message Buffer Configuration



19.10.1 Transmission Priority

If two or more transmission Message Buffers of a channel are configured for transmission, then the transmission priority in the RS-CANFD lite module can be selected from the following two modes:

- CAN ID priority
- Message Buffer number priority

The transmission priority mode is common for all Message Buffers. It can be configured via the **GCFG.TPRI** bit.

For Message Buffer number priority transmission, the smallest Message Buffer number with transmission request has the highest priority for transmission. This also includes the TX Message Buffers linked to the Common FIFO Buffer configured in TX mode.

For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All TX Message Buffers can enter the ID priority comparison for Message Buffers configured for transmission. This also includes the TX Message Buffers linked to the Common FIFO Buffer configured in TX mode.

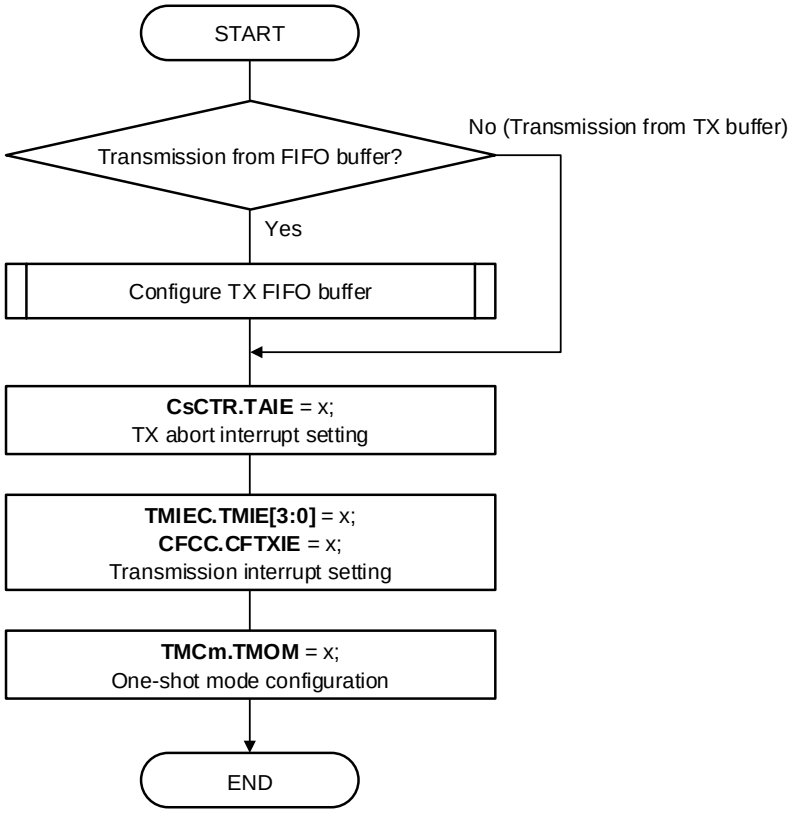
If the ID of two or more Message Buffers is the same, then the smaller Message Buffer number will have higher priority for transmission.

Caution For Common FIFO Buffer configured in TX mode, only the message currently being pointed to by the FIFO Read Pointer can be included in the transmission arbitration.

If the message is being transmitted from the FIFO, then the next pending Message within the same FIFO will be considered in the transmission arbitration.

Figure 19-29 below shows the transmission configuration flow.

Figure 19-29. Transmission Configuration Flow



19.10.2 Normal Transmission

Each transmission Message Buffer has two modes of message transmission:

(1) Regular Transmission Mode

If the Message Buffer is placed in regular transmission mode, the data frame or remote frame set in that Message Buffer can be transmitted.

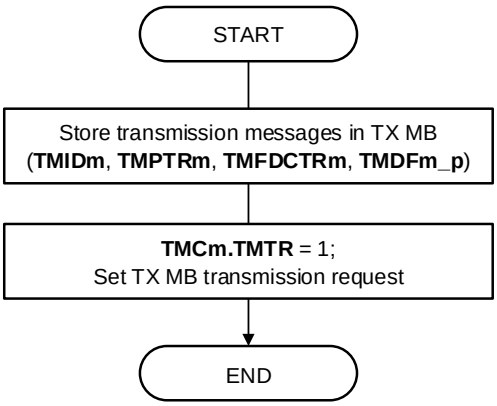
Completion of regular transmission can be checked through the related TX Message Buffer Transmission Result Flag (**TMSTSm.TMTRF[1:0]**). These bits are set to 10B or 11B when the regular transmission is successful.

When arbitration is lost or an error occurs, message transmission will be attempted further if no transmission abort request is set for this transmission Message Buffer.

New internal transmission arbitration for this channel will be performed considering all Message Buffers with transmission request.

The regular transmission request procedure after a configuration is shown in **Figure 19-30**.

Figure 19-30. Transmission Request Processing using Normal TX Message Buffer Mode



(2) One-shot Transmission Mode

When the **TMCm.TMOM** bit is set for a transmission Message Buffer, then the Message Buffer is placed in one-shot transmission mode and attempts to transmit a message only once.

Completion of one-shot transmission can be checked through the related TX Message Buffer Transmission Result Flag (**TMSTSm.TMTRF[1:0]**). The **TMSTSm.TMTRF[1:0]** bits are set to 10B or 11B when the one-shot transmission is successful.

The **TMSTSm.TMTRF[1:0]** bits are set to 01B when arbitration is lost or an error occurs during the transmission of the related Message Buffer.

Further message transmission will not be attempted in this case.

19.10.2.1 TX Message Buffer Control Register Setting

Table 19-18 shows configuration of the normal CAN transmission mode.

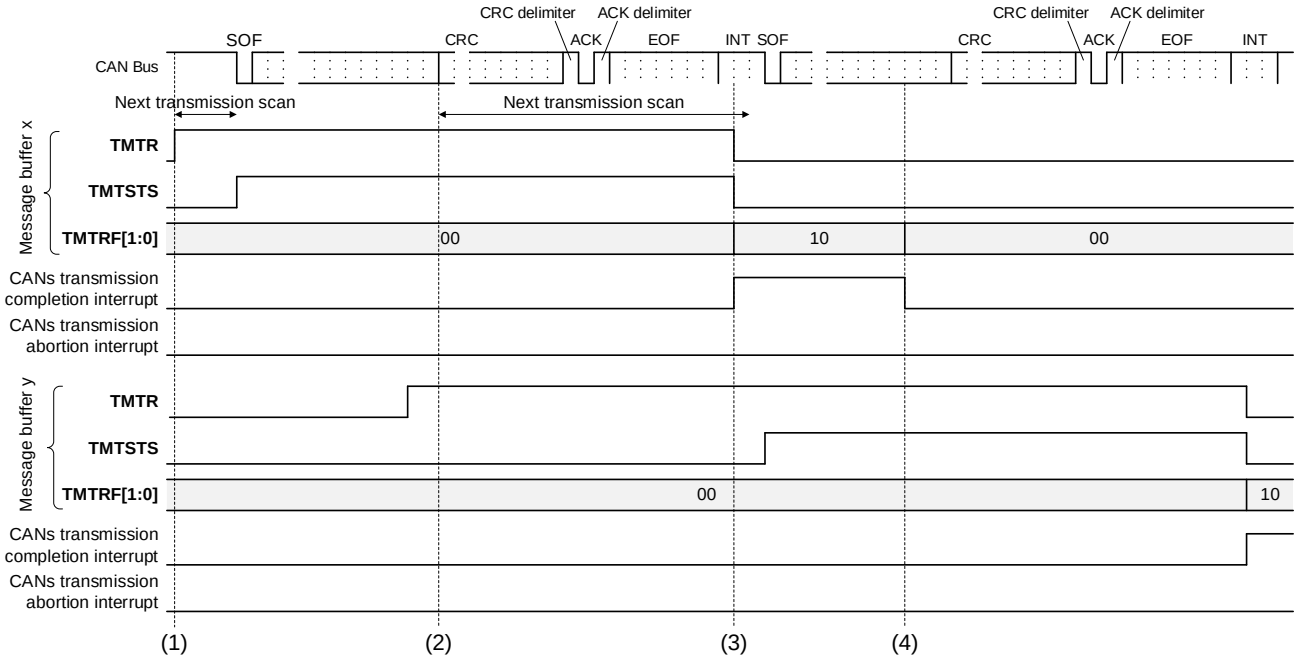
Table 19-18. Configuration of CAN Transmission Mode

Transmission Request **TMCm.TMTR**	Transmission Abortion Request **TMCm.TMTAR**	One-shot Enable **TMCm.TMOM**	Communication Activity
0	0	0	Message Buffer disabled.
0	0	1	Message Buffer disabled.
1	0	0	Configured as a transmission Message Buffer for a data frame or a remote frame.
1	0	1	Configured as an one-shot transmission Message Buffer for a data frame or a remote frame.
1	1	0	Transmission abortion requested.
1	1	1	One-shot transmission abortion requested.

The Configuration bits can be configured in the TX Message Buffer Control Registers.

The **Figure 19-31** shows timings for successful transmission for two Message Buffers.

Figure 19-31. Bit Status of Successful Transmission Timing



Description of (1) to (4) in the above figure:

(1) If the **TMCm.TMTR** bit is set in the bus idle state, Message Buffer scanning procedure starts to decide the highest priority Message Buffer for transmission.
When the transmission Message Buffer is decided, the **TMSTSm.TMTSTS** bit is set, and CAN channel starts the transmission **Note**.

(2) At 1st bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.

(3) If the message has been successfully transmitted, the **TMSTSm.TMTRF[1:0]** bits are set to 10B and **TMSTSm.TMTSTS** and the **TMCm.TMTR** bits are cleared.
When the **TMIEC.TMIE[3:0]** bits is set, the CAN successful transmission interrupt request is generated.
To clear the related interrupt line the **TMSTSm.TMTRF** have to be cleared.

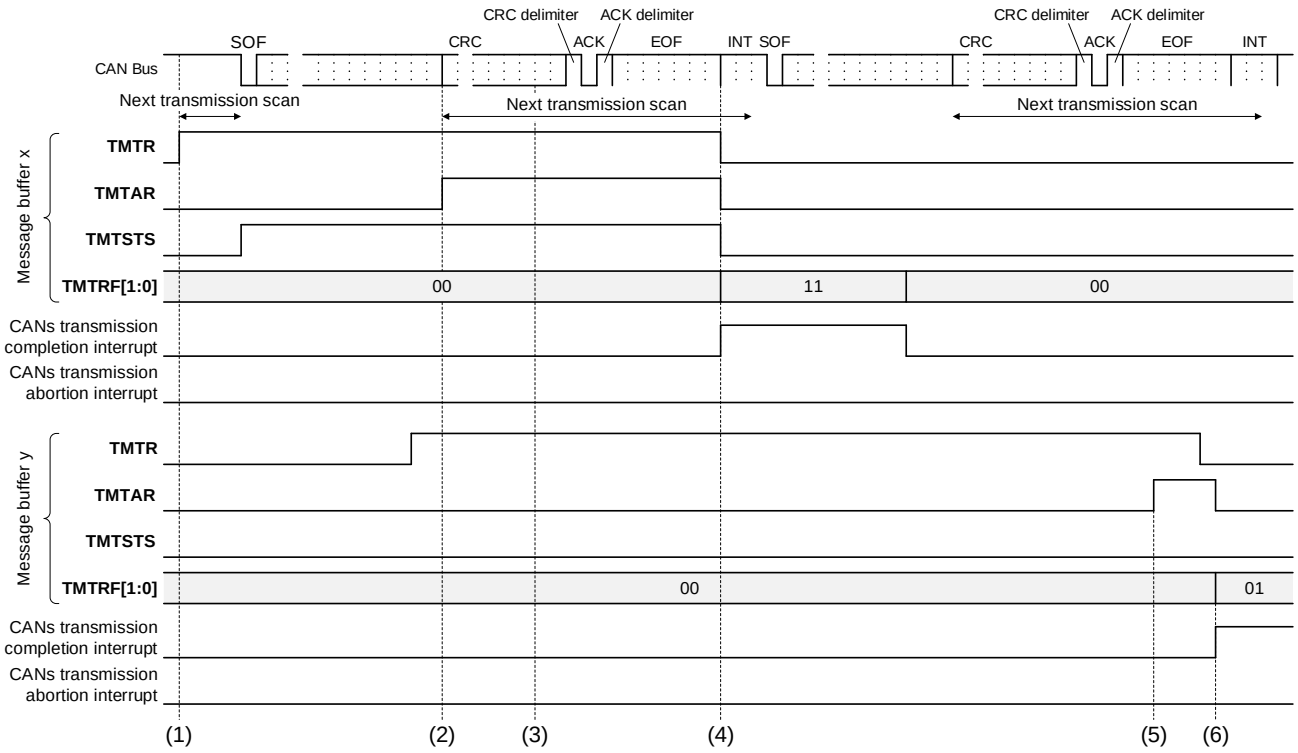
(4) Before starting the next transmission, clear the **TMSTSm.TMTRF[1:0]** bits. Load the next message in the transmission Message Buffer and set the **TMCm.TMTR** bit again.
TMCm.TMTR bit cannot be set again before **TMSTSm.TMTRF[1:0]** bits are cleared.

Note If arbitration is lost after the CAN channel starts the transmission, the **TMSTSm.TMTSTS** bit is cleared. The transmission scanning procedure is performed again to search for the highest priority transmission Message Buffer from the beginning of the 1st CRC bit.
If an error occurs either during the transmission or following the loss of arbitration, then during Error Frame, the transmission scanning procedure is performed again to search for the highest priority transmission Message Buffer.

Caution The setting point of TMSTSm.TMTSTS is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID due to the synchronisation logic implemented for the PLL bypass.

The **Figure 19-32** shows timings for transmission abort for two Message Buffers.

Figure 19-32. Bit Status of Transmission Abort Timing



Description of (1) to (6) in the above figure:

(1) If the **TMCm.TMTR** bit is set in the bus idle state, Message Buffer scanning procedure starts to decide the highest priority Message Buffer for transmission.
If transmit buffer a is determined to be the highest-priority transmit buffer, the **TMSTSm.TMTSTS** bit is set to 1 (transmission is in progress) and the CAN channel starts transmitting data **Note**.

(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 **TMCm.TMTAR** bit is set to 1 (transmit abort is requested).

(3) At 1st CRC bit, the transmission scanning procedure starts for the next transmission. In this example timing chart Message Buffer y is not selected as next transmission Message Buffer.

(4) If the message has been successfully transmitted, the **TMSTSm.TMTRF[1:0]** bits are set to 11B and **TMSTSm.TMTSTS** and the **TMCm.TMTR** bits are cleared.
When the **TMIEC.TMIE** bits is set, the CAN successful transmission interrupt request is generated.
To clear the related interrupt line the **TMSTSm.TMTRF[1:0]** bits has to be cleared.

(5) Another CAN node is transmitting on the CAN bus (**TMSTSm.TMTSTS** not set): if the **TMCm.TMTAR** bit is set when the related channel is under transmission scan then the transmission request cannot be cleared.

(6) After internal processing time the transmission is aborted and the **TMSTSm.TMTRF[1:0]** bits are set to 01B.
If the Message Buffer is not transmitting or selected as next transmission Message Buffer or under transmit scan, then the abort is immediately accepted and the corresponding **TMSTSm.TMTRF[1:0]** bits are set to 01B.
In addition, **TMCm.TMTR**, and **TMCm.TMTAR** bits are cleared automatically.
When the **CsCTR.TAIE** bit is set then an interrupt is generated for successful transmission abort.
To clear the related interrupt line the **TMSTSm.TMTRF[1:0]** bits have to be cleared.

Note If arbitration is lost after the CAN channel starts the transmission, the **TMSTSm.TMTSTS** is cleared. The transmission scanning procedure is performed again to search for the highest priority transmission Message Buffer from the beginning of the 1st CRC bit.
If an error occurs, either during the transmission, or following the loss of arbitration, then during Error Frame, the transmission scanning procedure is performed again to search for the highest priority transmission Message Buffer.

19.10.3 TX FIFO Transmission

One Common FIFO buffer is assigned to RS-CANFD lite module. The FIFO buffer could be linked to any normal TX Message Buffer position for this channel by the **CFCC.CFTML** bits if configured in TX mode.

When the transmission scan starts and the FIFO Buffer corresponding to this TX Message Buffer is enabled, then the relevant message in the FIFO Buffer will participate in the transmission scan.

Configuration of a TX Message Buffer linked to a FIFO Buffer configured in TX mode should not be done.

19.10.3.1 TX FIFO Operation

CAN Messages can be written into the TX FIFO by writing to the corresponding FIFO Access registers.

When the value FFH is written into the corresponding FIFO Pointer Control Register, then the Message Count of the related FIFO is incremented by 1.

Write only the value of FFH for FIFO Pointer Control Register after writing the complete message to the corresponding FIFO Access registers.

If the Message count matches the FIFO Depth, then the FIFO Full flag is set.

The oldest message in the TX FIFO is included in the scan for transmission by the corresponding RS-CANFD lite module channel logic.

When a message is successfully transmitted from the TX FIFO, the Message Count value is decremented by 1.

When all the messages from the FIFO are transmitted, the FIFO Empty flag is set.

The Interrupt generation conditions for the TX FIFO buffer can be configured by configuring the **CFCC.CFIM** bit in the corresponding Common FIFO Configuration / Control Register.

- If **CFCC.CFIM** bit is 0, then interrupt is generated when last message is successfully transmitted from the TX FIFO buffer.
- If **CFCC.CFIM** bit is 1, then interrupt is generated for every successfully transmitted message from the TX FIFO buffer.

Common FIFO can set interrupt when CAN frame transmitted is completed.

The Common FIFO Buffer configured in TX Mode can be disabled by clearing the **CFCC.CFE** bit in the Common FIFO Configuration / Control Register. If this bit is cleared to 0, the FIFO empty flag is set as described below:

- The flag is set immediately if a message from the TX FIFO is not scheduled for the next or subsequent transmission.
- The flag is set on error detection on the CAN bus, arbitration lost, transition to Channel or Global HALT mode if transmission from the TX FIFO is already scheduled or in progress.

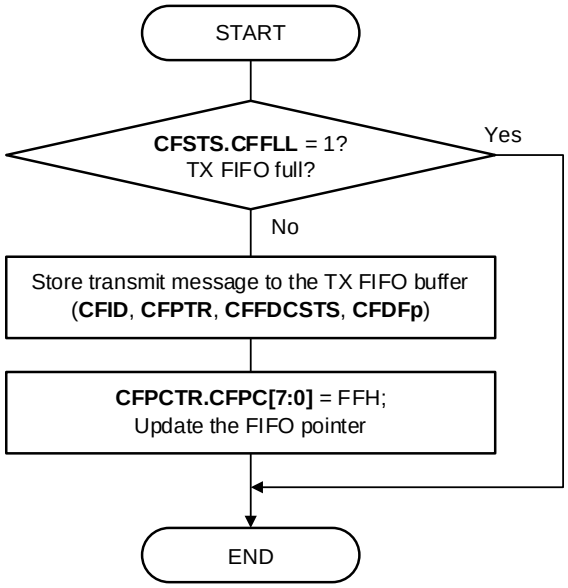
Caution Common FIFO buffer is disabled after clearing the CFCC.CFE bit only if the Empty flag (CFSTS.CFEMP) is set for the corresponding Common FIFO Buffer.

Other possible messages pending from the TX FIFO are lost and their transmission needs to be requested again. Before **CFCC.CFE** is set again ensure that **CFSTS.CFEMP** bit is set and that there is no pending abort from the TX FIFO.

When the **CFCC.CFE** bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Hence, all messages in the FIFO buffers will be lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after a configuration is shown in **Figure 19-33**.

Figure 19-33. TX FIFO Transmission Request Procedure



19.10.3.2 Interval Timer for FIFO Transmission

For each Common FIFO in TX mode it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is, called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the **CFCC.CFE** bit is set.

When the Common FIFO in TX mode is enabled, then the first message will be transmitted without considering this interval time.

The Interval Timer will stop counting when:

- FIFO is disabled by clearing the **CFCC.CFE** bit.
- CAN channel is in CH_RESET mode.

The interval time is specified by the **CFCC.CFITT** value and can be specified from 0 to 255 timer units.

The timer unit can be defined based on two different source clocks for the interval timer. To disable the interval timer for FIFO transmission a value of 0 should be selected.

The timer source can be selected by the configuration bit (**CFCC.CFITSS**). For the timer source the CAN Bit Timing clock of the FIFO related channel or a global reference clock could be selected.

If CAN channel bit time clock is configured as clock source and the CAN channel enters CH_HALT or CH_RESET or CH_SLEEP mode, then the Interval Timer is stopped for that channel.

If peripheral clock is selected as Interval Timer clock source, then the Interval Timer is stopped only when the CAN channel is in CH_RESET or CH_SLEEP mode.

The reference clock can be used to configure the interval time in fixed time units. It is based on the peripheral clock. The reference clock prescaler value (**GCFG.ITRCP**) defines the relation between the peripheral clock frequency/period and the reference clock period.

Refer to **Table 19-19** for **GCFG.ITRCP** configuration values to achieve different reference clock periods based on the peripheral clock frequency/period.

Table 19-19. Configuration Example for the FIFO Interval Timer Reference Clock (GCFG.ITRCP Setting Value)

Reference Clock Period / Peripheral Clock Frequency/Period	1 µs	100 µs	500 µs
16 MHz / 62.5 ns	16	1600	8000
20 MHz / 50 ns	20	2000	10000
32 MHz / 31.25 ns	32	3200	16000
40 MHz / 25 ns	40	4000	20000

Additionally the reference clock resolution can be specified by the Interval Timer Reference Clock Resolution value (**CFCC.CFITR**).

The interval time is based on the reference clock period multiplied by the configured value (x1 or x10).

The reference clock based interval timer can be used to follow the requirements of the ISO 15765-2 Separation Time. The whole range for the Separation Time from 100 µs to 127 ms can be covered.

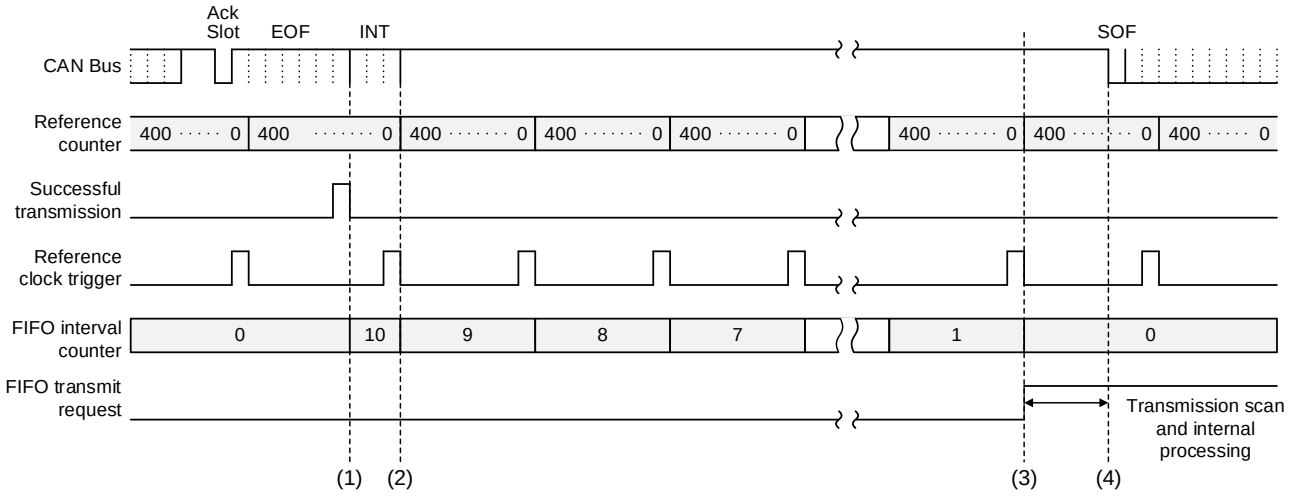
The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).

When the interval time has elapsed, the next transmission request is raised by the related TX FIFO. Hence, the interval time defines the minimum time between two messages transmitted from one FIFO.

The next message will earliest be sent after this interval time.

The **Figure 19-34** shows an example timing of the internal processing.

Figure 19-34. Example for Interval Processing Time



The configuration for this timing above is following:

- Peripheral clock frequency = 40 MHz
- Interval Timer Reference Clock (**GCFG.ITRCP**) = 400 times
- Reference clock due to the settings above = 10 µs
- Common FIFO interval Timer Source Selection (**CFCC.CFITSS**) = 0
- Common FIFO Interval Timer Resolution (**CFCC.CFITR**) = 0
- Common FIFO Interval Transmission Time (**CFCC.CFITT**) = 10 times
- Theoretical message separation interval = 100 µs

(1) Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the Reference clock trigger. Therefore the first interval is counting less or equal to one Reference clock interval.

(2) With the next reference clock trigger the FIFO interval timer is decremented and so on.

(3) When the FIFO interval timer reached the value “zero” the FIFO Transmit request is set.

(4) When the FIFO is selected for transmission then the transmission will start soon. Due to internal processing this usually takes less than 3 CAN bit time, between internal FIFO transmit request set (3) and actual transmission.

In worst case when multi events like reception scan, internal message routing, transmit scan on happen, then it could take up to 120 peripheral clock cycles.

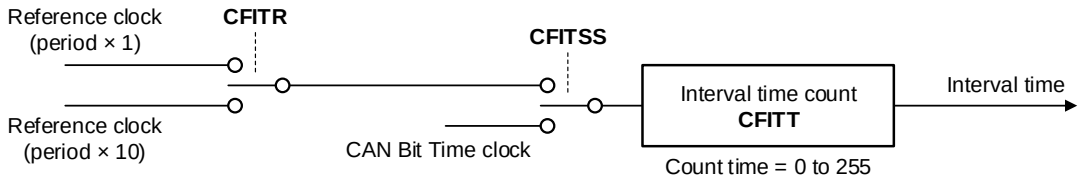
As shown in **Figure 19-34**, it is not guaranteed that the minimum interval is always equal to the configured value.

If a minimum time must never be breached, **CFCC.CFITT** should be configured with the required minimum value + 1.

If further TX Message Buffers or TX FIFO are configured for transmission for the same channel the real delay between two messages transmitted from a TX FIFO can be much longer than specified by the interval time due to higher priority message transmission from these TX Message Buffers or TX FIFO.

Figure 19-35 shows a block diagram of the FIFO interval time generation circuit.

Figure 19-35. Block Diagram of FIFO Interval Timer



CFITR: **CFCC.CFITR** (Common FIFO Interval Timer Resolution Selection)
CFITSS: **CFCC.CFITSS** (Common FIFO Interval Timer Source Selection)
CFITT: **CFCC.CFITT[7:0]** (Common FIFO Interval Transmission Time Count)

19.10.4 TX History List

The TX History List function records the information of the successfully transmitted message in the TX History List Buffers. THL Buffer can store up to 8 THL entries.

The **THLCC.THLDTE** bit can be used to configure if only message information from TX FIFO is stored, TX FIFO or normal TX Message Buffers should be stored in the TX History List.

Each transmit message can be individually configured for acceptance to the TX History List by the **CFID.THLEN** bit or the **TMIDm.THLEN** bit.

The message information is stored to the TX History List Buffer after the message is successfully transmitted.

Storing to the List is not synchronized with the status of **TMSTSm.TMTRF** bits.

Due to internal processing, the storage to the List could happen with a delay after the successful transmission indication.

Storing the TX History List data can be recognized by the condition that the **THLSTS.THLIF** bit is set to 1 when the **THLCC.THLIE** bit is configured to 1 or when the TX History List counter **THLSTS.THLMC** is increased.

In worst case when multi events like reception scan, internal message routing on happen.

Maximum delay time from setting the **TMSTSm.TMTRF** to storing the TX History List data is 70 peripheral clock cycles.

The History list records following information of the transmitted message:

- Buffer Type (**THLACC0.BT**):

  01B: TX Message Buffer
  10B: TX FIFO

- Buffer Number (**THLACC0.BN**):

  TX Message Buffer, or TX Message Buffer Link for the Common FIFO Buffer from which the transmission occurred. The number depends upon the Buffer Type, refer to **Table 19-20**.

- Transmission ID (**THLACC1.TID**):

  Transmission Pointer stored in the transmission message.

- Transmit Timestamp (**THLACC0.TMTS**):

  Message timestamp captured at capture point as configured by **GFDCFG.TSCCFG**.

- Transmission Information Label (**THLACC1.TIFL**):

  Transmission information label stored in the transmission message.

Table 19-20. TX History List Buffer Number Entry

Buffer Type / Buffer Number	01B	10B
00B	Message Buffer 0	Numbers of transmit buffers linked to Common FIFO buffers by the **CFCC.CFTML** bits.
01B	Message Buffer 1	
10B	Message Buffer 2	
11B	Message Buffer 3	

The Transmission ID entry is used to identify which message of a TX FIFO has been successfully transmitted because the TX FIFO number alone is not sufficient.

Therefore, a unique number can be attached to each transmission message stored in a TX FIFO. This unique identification number should be written to the **CFFDCSTS.CFPTR[15:0]** for a TX FIFO.

When the message is successfully transmitted then this identification number is stored together with the other message related information to the TX History List and can be read via the Transmission ID (TID) of the TX History List Access Register.

Also for normal TX Message Buffers, the **TMFDCTR.TMPTR[15:0]** will be stored in the Transmission History List. Information label is the same.

Figure 19-36 shows a transmission preparation flow when TX History List is used.

Read access to the TX History List Access Register will be done for every single entry.

After reading one entry, FFH has to be written to the corresponding TX History List Pointer Control Register to be able to access the next entry until TX History List is empty.

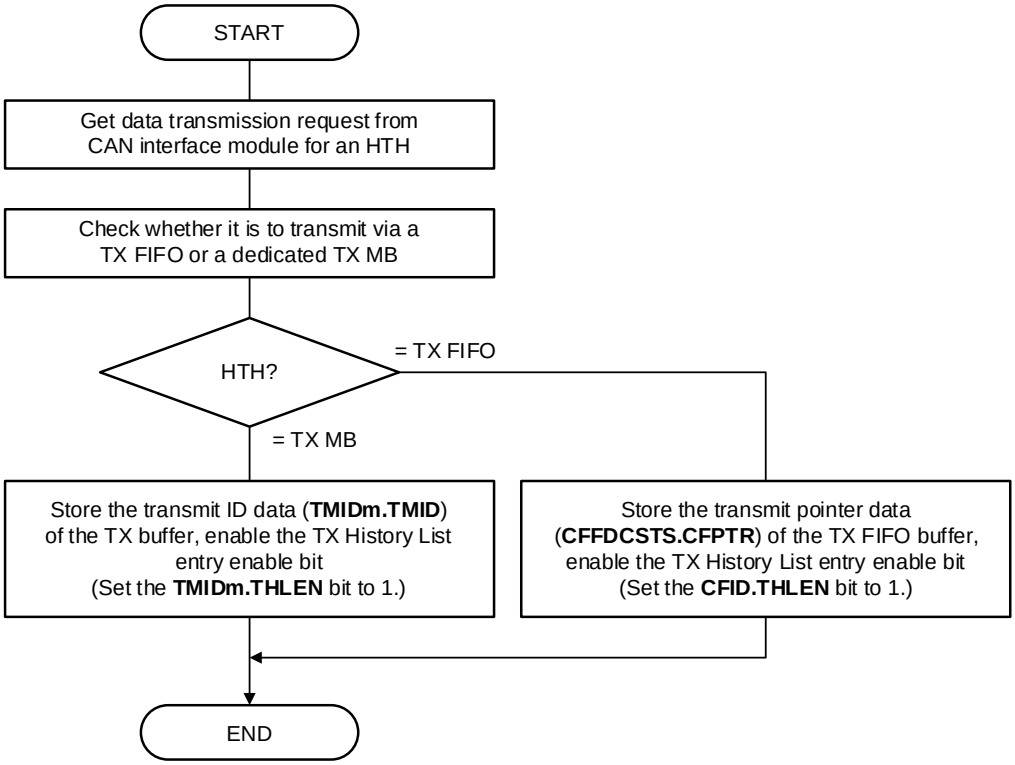
Figure 19-37 shows an example flow for processing the TX History List information.

The TX History Lists have dedicated interrupts, which can be configured with the **THLCC.THLIM** bit and enabled with the **THLCC.THLIE** bit, either to generate an interrupt when the History List reached a filling level of 75% or for every new TX History List entry.

An entry lost indication is flagged by the **THLSTS.THLELT** bit.

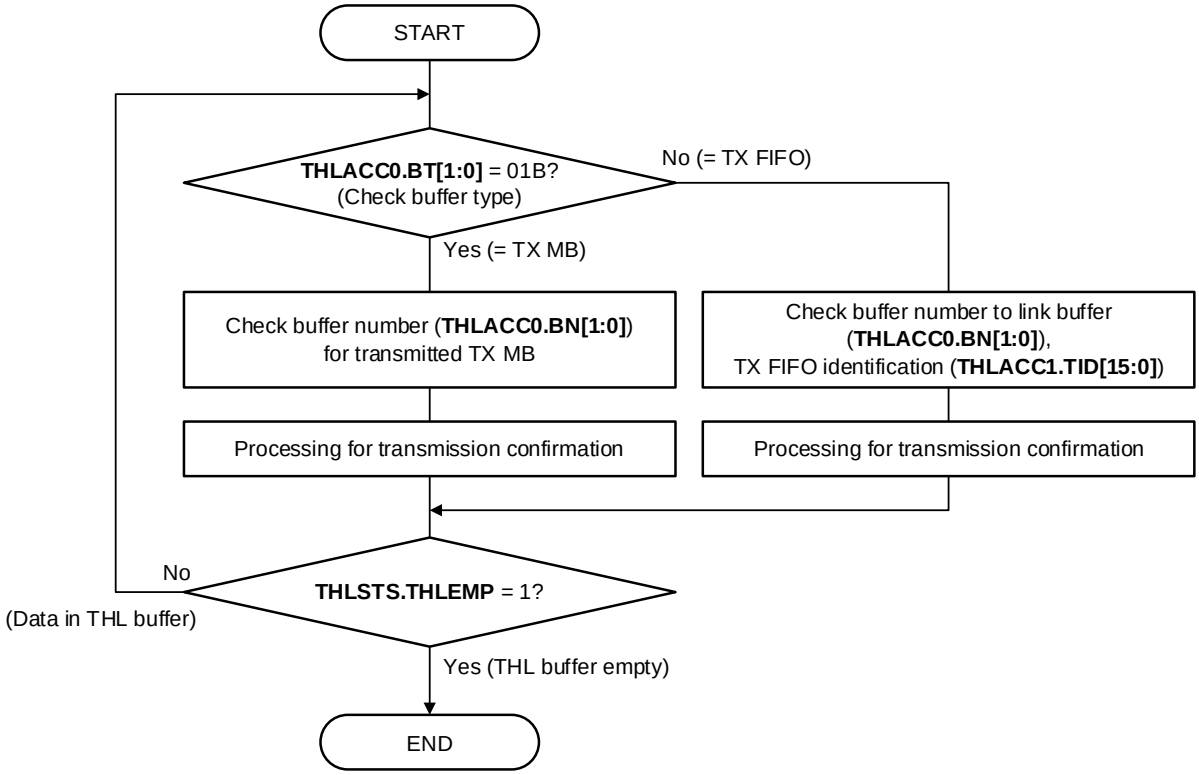
Status of this bit is also shown by the **GERFL.THLES** bit.

Figure 19-36. TX History List Preparation Flow



Remark HTH: Hardware Transmission Handler
TMIDm.THLEN: TX History List entry bit in the Transmission Buffer
CFID.THLEN: TX History List entry enable bit in the Common FIFO
CFFDCSTS.CFPTR: Pointer field of the Common FIFO

Figure 19-37. TX History List Processing Flow



19.10.5 TX Data Padding

If the data length code (DLC) of the transmitting message is higher number of data bytes than the buffer size. Then the data bytes beyond the restricted range will be replaced by bytes with the value of CC HEX.

This could happen for Common FIFO configured as (TX) when the transmit message DLC is higher than the **CFCC.CFPLS[2:0].**

This could also happen in FD only mode, if a Classical-CAN Frame is configured with a DLC bigger than 8.

19.11 Test Function

The RS-CANFD lite module can be configured into Test Modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the RS-CANFD lite module in the test modes.

Note that all Test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable the various Test modes combinations specified in this Chapter.

The Test modes can be broadly split into 2 groups:

- Channel specific test modes
- Global test modes

19.11.1 Channel Specific Test Modes

CAN channel can be configured into following test modes:

- Basic Test Mode
- Listen-Only Mode
- Self Test Mode 0 (External Loopback Mode)
- Self Test Mode 1 (Internal Loopback Mode)
- Restricted Operation Mode

19.11.1.1 Basic Test Mode

CRC testing is enabled in basic test mode. The CRC value calculated by the RS-CANFD module based on the transmit message or receive message is stored in the register. This CRC value is stored in the **CsERFL.CRCREG[14:0]** bits when the message is a Classical-CAN frame (CRC length = 15 bits) or in the **CsFDCRC.CRCREG[20:0]** bits when the message is a CAN-FD frame (CRC length = 17 or 21 bits).

Use the bit flip test function for CRC error tests (intentionally generate a CRC error). For details, see **19.11.2.2 Bit Flip Test**.

19.11.1.2 Listen-only Mode

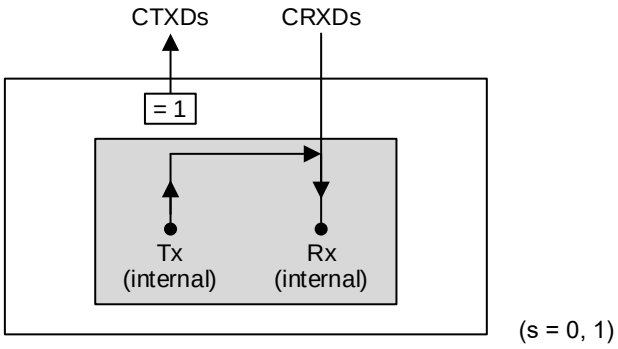
The ISO 11898-1 recommends an optional bus-monitoring mode. In this mode, the CAN channel is able to receive valid data frames and valid remote frames. However, it sends only recessive bits on the CAN bus and is not allowed to transmit.

If the CAN engine is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is routed internally so that the CAN engine monitors this as dominant. The external CTXDs pin will remain in recessive state.

This mode can be used for baud rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.

In this mode, it is not permitted to request transmission from any normal TX Message Buffer or TX FIFO of this channel.

Figure 19-38. Operation of CTXDs and CRXDs Pins in Listen-Only Mode



19.11.1.3 Self Test Mode 0 (External Loopback Mode)

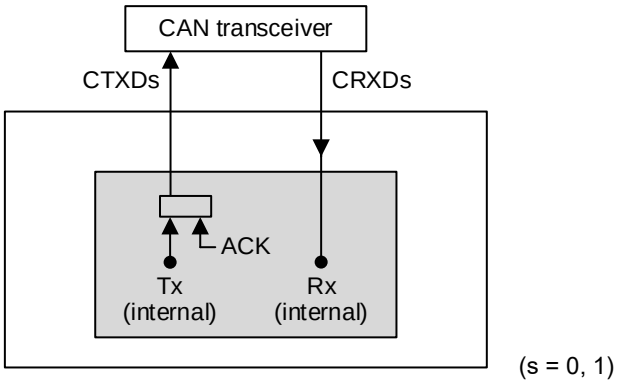
In Self Test Mode 0, the CAN engine treats its own transmitted messages as received messages via the CAN transceiver and can store them into Receive Message Buffers.

To be independent from external stimulation the engine generates its own acknowledge bit.

This test can be used for CAN transceiver tests.

The CRXDs/CTXDs pins should be connected to the transceiver.

Figure 19-39. Operation of CTXDs and CRXDs Pins in Self Test Mode 0 (External Loopback Mode)



19.11.1.4 Self Test Mode 1 (Internal Loopback Mode)

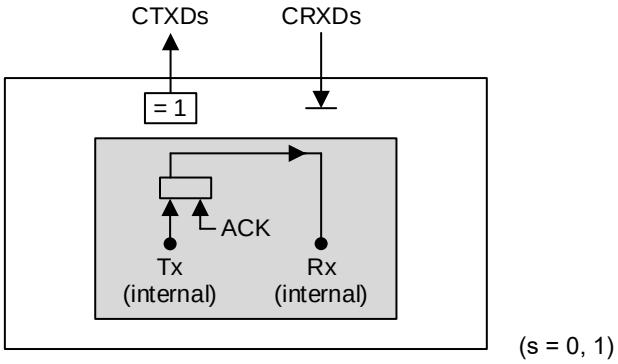
In Self Test Mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent from external stimulation the CAN engine generates its own acknowledge bit. In this mode the CAN engine performs an internal feedback from Tx internal to Rx internal. The actual value of the external CRXDs input is disregarded by the CAN engine.

The external CTXDs pin outputs only recessive bits.

The CRXDs/CTXDs pins do not need to be connected to the CAN bus or any external device.

Caution The channel pins are disconnected from the internal CAN bus communication lines, so internal signals cannot be monitored by external pins.

Figure 19-40. Operation of CTXDs and CRXDs Pins in Self Test Mode 1 (Internal Loopback Mode)



(s = 0, 1)

19.11.1.5 Restricted Operation Mode

In Restricted Operation Mode, the CAN node is able to receive valid data and remote frames generating the acknowledge bit.

Active error and overload frames cannot be transmitted instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.

Moreover, the receive and transmit error counter (REC and TEC) are frozen independently from the occurrence of errors.

The mode is specified as in ISO 11898-1; however, it is permitted to set any transmit requested.

19.11.2 Global Test Modes

The RS-CANFD lite module can be configured into following test modes:

- RAM Test Mode
- Bit Flip Test

For RAM test mode is protected by a special SW procedure to enable the mode. This SW procedure enables the write access to the test mode by specific unlock key, the related unlock key can be seen in the table below:

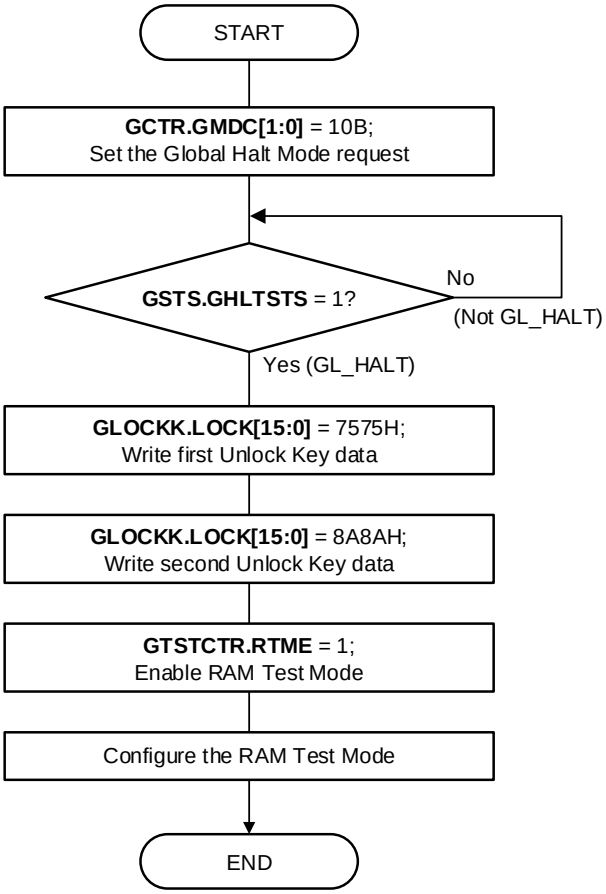
Table 19-21. Unlock Key in RAM Test Mode

Test Mode	GLOCKK.LOCK	
	Unlock Key 1	Unlock Key 2
RAM Test Mode	7575H	8A8AH

If the SW sequence of the two consecutive unlock key write accesses (16-bit accesses) is interrupted by any other write access to the SFR or if incorrect data is written to the Global Lock Key register then the corresponding Test mode cannot be set and the sequence should be re-started.

After the two unlock key write accesses, the next write access should be to set the corresponding test mode enable bit. If this is not followed, the unlock mechanism resets and the test mode enable bit cannot be set and then the unlock sequence should be restarted.

Figure 19-41. Unlock SW Protection Routine



19.11.2.1 RAM Test Mode

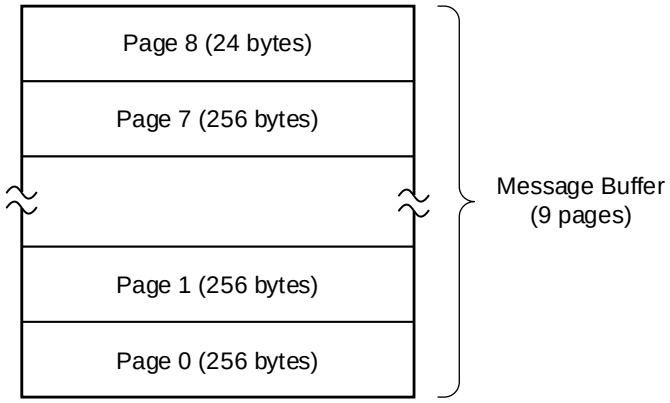
The RS-CANFD lite module can be configured in RAM Test Mode by setting the **GTSTCTR.RTME** bit when the corresponding lock key is written before. This is a special test mode, in which, the complete RAM area can be accessed.

In this mode, the RAM area is split into number of pages (**pn**) of 256 Bytes each. Which can be accessed via **RPGACCr** register.

The page should be selected for read / write access by writing to the **GTSTCFG.RTMPS[3:0]** bits. Then, data can be read from or written to the RAM Test Page Access Registers.

Figure 19-42 shows the structure of the pages in the RAM when performing a RAM Test Mode.

Figure 19-42. RAM Page Structure



Page: **GTSTCFG.RTMPS[3:0]**

The total available RAM size is 2072 Bytes for the Message Buffer RAM.

pn = ceil (Total RAM size in Bytes / Number of Bytes per page)

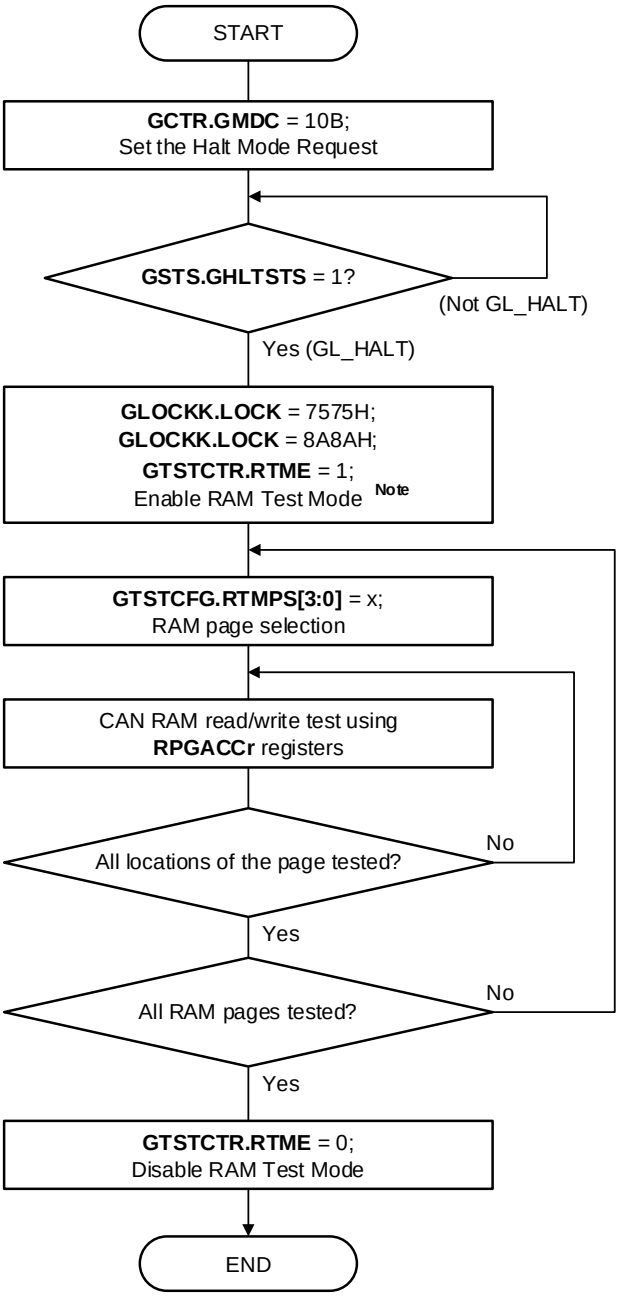
MB RAM:

pn = ceil (2072 / 256) = 9 pages

GTSTCFG.RTMPS[3:0] = 0 to 8 inclusive

Figure 19-43 below shows the SW flow for RAM Test mode.

Figure 19-43. RAM Test Mode SW Flow



Note Change into the following status before changing to RAM Test.

- Cancel of a request of transmission.
- Disable of all the FIFO.
- Clear of the receiving flag of a receiving buffer.

To exit this Test mode, the **GTSTCTR.RTME** bit must be cleared. The **GTSTCTR.RTME** bit is cleared by writing 0 to it.

The **GTSTCTR.RTME** bit is cleared automatically when the RS-CANFD lite module enters Global Reset mode from the Test mode.

19.11.2.2 Bit Flip Test

Bit Flip Test can invert the bit (the 1st bit of ID) of the beginning of the bit stream to receive.

If this function is used by a transmitting node, a bit error or an arbitration lost will occur.

If this function is used by a receiving node, a CRC error or a stuff error will occur.

Refer to the bit stuffing rule when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The following sequence should be used to perform CRC error testing. In the sequence below RS-CANFD lite module is the receiver.

1. Set the **CsCTR.BFT** bit to 1, in order to invert the first bit of the incoming bit stream from sending node.
2. Wait for the INTRCANsERR (CANs channel error interrupt) output signal to set to 1.
3. Read either the **CsERFL.CRCREG** (for classical CAN frame) or the **CsFDCRC.CRCREG** (for CAN-FD frame). The value should be different from the received CRC value of the reference message from sending node.
4. Check that **CsERFL.CERR** is 1 (CRC error detection).

As the CRC generator logic is shared for RX and TX there is no need to create a separate TX CRC error test.

19.12 Interrupt

19.12.1 Interrupt Sources

The RS-CANFD lite module generates several Interrupts.

The interrupt output, which is connected to the Interrupt Controller Unit, can be controlled by the corresponding interrupt enable bit.

The status flag will be set independent from this enable bit.

The channel Transmission Interrupt has an additional Status flag register; these Status bits will only be set when the corresponding interrupt enables are set.

This register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.

The Interrupts in the RS-CANFD lite module can be classified into 2 groups, Global Interrupts and Channel Interrupts:

1. Global Interrupts:

The RS-CANFD lite module can generate 3 types of Global Interrupts:

(1) Global Interrupt for successful reception into the 2 RX FIFO buffers
(2) Global Interrupt for successful reception into the 16 RX Message Buffers
(3) Global Error Interrupt

2. Channel Interrupts:

Each channel of the RS-CANFD lite module can generate 3 types of Channel Interrupts:

(1) CANs Channel Transmission Interrupts
- CANs transmission completion from channel
- CANs transmission abort from channel
- CANs channel THL Interrupt
- CANs successful transmission from a Common FIFO in TX mode for a channel

(2) Channel Error Interrupt
(3) Successful Reception in a Common FIFO in RX mode for a channel

3. Other Interrupts:

(1) CANs wakeup

The CANs wakeup interrupt is generated in every mode when a falling edge in the CRXDs pin is detected. When the CANs wakeup interrupt is used, set the function of the corresponding port to CRXDs. The CANs wakeup interrupt is controlled by the interrupt function.

(2) CAN RAM ECC: 1-bit correction/2-bit error detection [Note].

Note For details on the CAN RAM ECC function, refer to **29.3.4 CAN RAM-ECC Function (RL78/F25 Only)**.

The interrupts are cleared when the corresponding flag bits are cleared or Interrupt enable bits are cleared.

If the set from the RS-CANFD lite module occurs simultaneously with the clear by the write access, then each flag bit is set. The set condition of the flag bits is prioritized.

Table 19-22. Interrupt Source Overview

Item	Interrupt Name	Interrupt Source		CAN Corresponding Interrupt Request Flag Note 1	CAN Corresponding Interrupt Enable Bit Note 1
Global Interrupts	INTRCANsGRFR	CAN Global Receive FIFO	Receive FIFO 0	**RFSTS0s.RFIF**	**RFCC0s.RFIE**
			Receive FIFO 1	**RFSTS1s.RFIF**	**RFCC1s.RFIE**
	INTRCANsGRVC	CAN Global Receive Message Buffer Interrupt		**RMNDs.RMNS[15:0]**	**RMIECs.RMIE[15:0]**
	INTRCANsGERR	CAN Global Error	DLC Error	**GERFLs.DEF**	**GCTRs.DEIE**
			Message Lost Error	**GERFLs.MES**	**GCTRs.MEIE**
			TX History Entry Lost Error	**GERFLs.THLES**	**GCTRs.THLEIE**
			CAN-FD Message Payload Overflow	**GERFLs.CMPOF**	**GCTRs.CMPOFIE**
Channel Interrupts	INTRCANsTRM	CANs Channel Transmit	CANs Transmit Complete	**TMSTSms.TMTRF[1:0]**	**TMIECs.TMIE[3:0]**
			CANs Transmit Abort	**TMSTSms.TMTRF[1:0]**	**CsCTR.TAIE**
			CANs Transmit / Receive FIFO Transmit Complete	**CFSTSs.CFTXIF**	**CFCCs.CFTXIE**
			CANs Transmit History	**THLSTSs.THLIF**	**THLCCs.THLIE**
	INTRCANsCFR	CANs Transmit / Receive FIFO Receive		**CFSTSs.CFRXIF**	**CFCCs.CFRXIE**
	INTRCANsERR	CANs Channel Error	Bus Error	**CsERFL.BEF**	**CsCTR.BEIE**
			Error Warning	**CsERFL.EWF**	**CsCTR.EWIE**
			Error Passive	**CsERFL.EPF**	**CsCTR.EPIE**
			Bus-Off Entry	**CsERFL.BOEF**	**CsCTR.BOEIE**
			Bus-Off Recovery	**CsERFL.BORF**	**CsCTR.BORIE**
			Overload	**CsERFL.OVLF**	**CsCTR.OLIE**
			Bus Lock	**CsERFL.BLF**	**CsCTR.BLIE**
			Arbitration Lost	**CsERFL.ALF**	**CsCTR.ALIE**
			Error Occurrence Counter Overflow	**CsFDSTS.EOCO**	**CsCTR.EOCOIE**
			Successful Occurrence Counter Overflow	**CsFDSTS.SOCO**	**CsCTR.SOCOIE**
			Transceiver Delay Compensation Violation	**CsFDSTS.TDCVF**	**CsCTR.TDCVFIE**
Other Interrupts	INTRCANsWUP	CANs Wakeup		none	none
	INTCRAM	CAN RAM ECC 1-bit Correction/2-bit Error Detection Note 2		none	none

Notes 1. For details on the interrupt requests flag and interrupt enable bits, refer to **Chapter 22. Interrupt Functions**.

2. For details on the CAN RAM ECC function, refer to **29.3.4 CAN RAM-ECC Function (RL78/F25 Only)**.

Figure 19-44. Global Interrupt Block Diagram

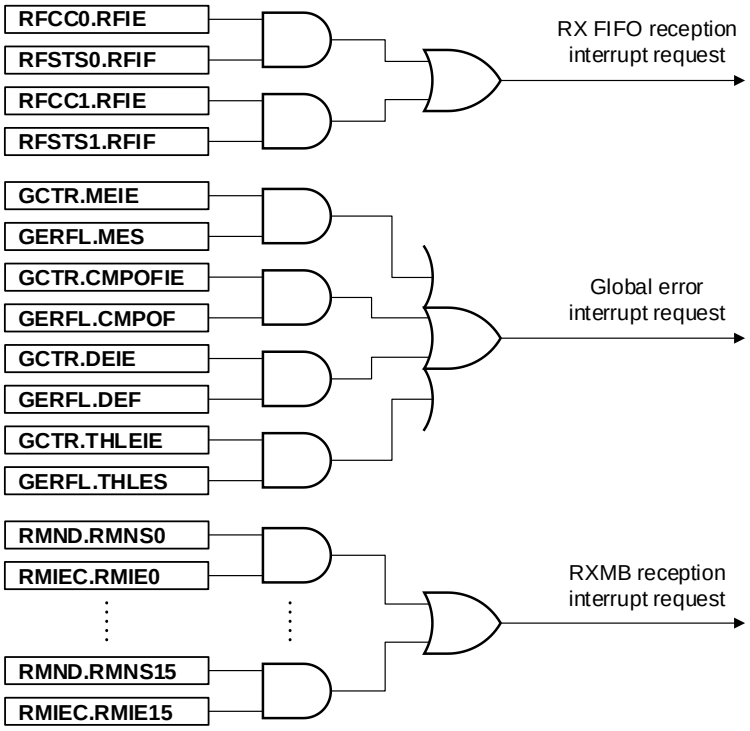
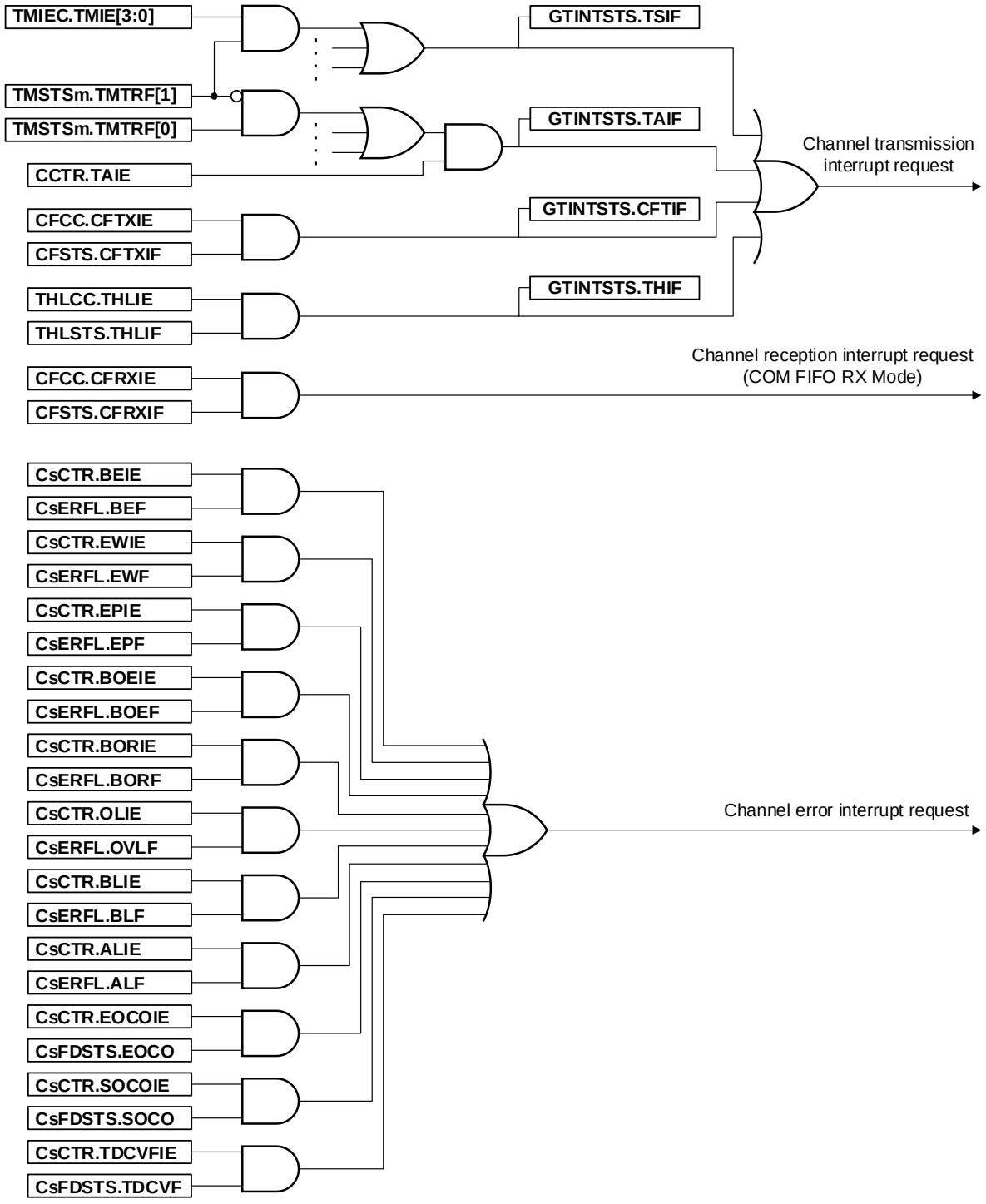


Figure 19-45. Channel Interrupt Block Diagram



19.12.2 Interrupt Processing Flow

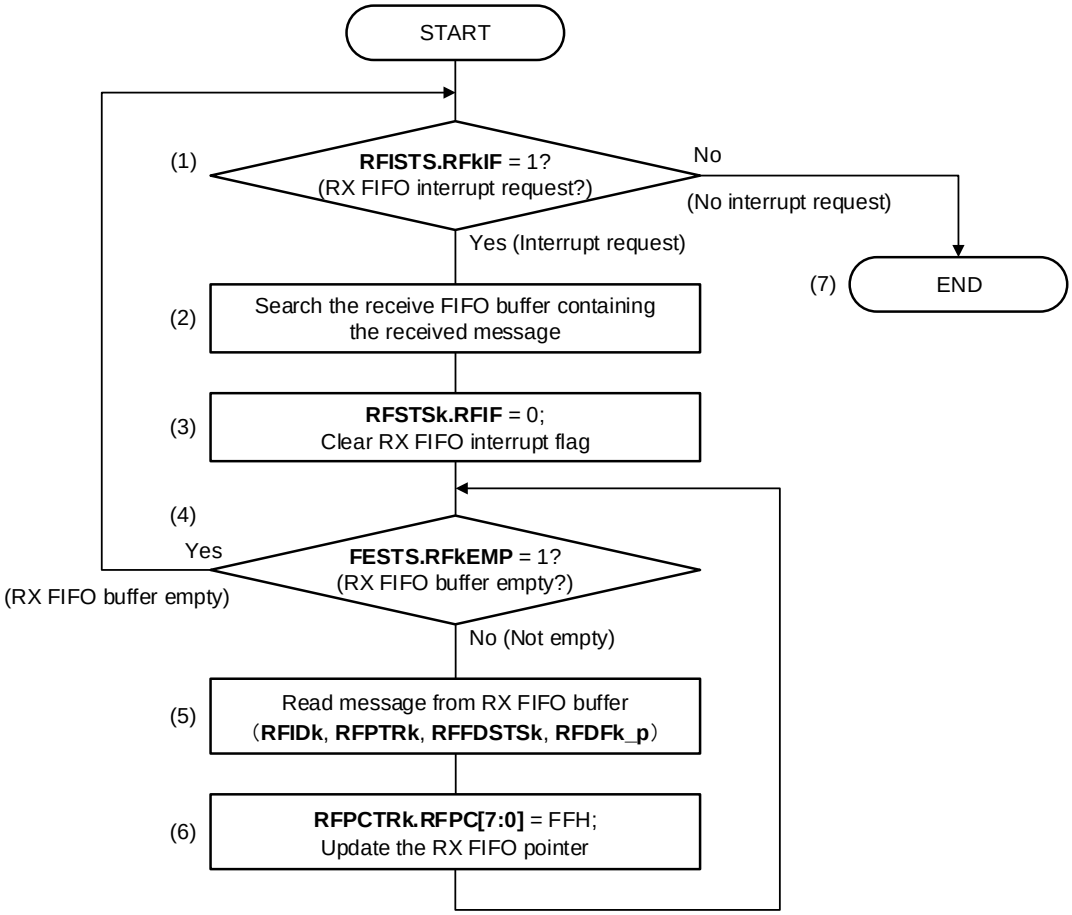
All interrupt signals are not cleared automatically. To clear an interrupt flag in the SFR during the interrupt routine.

A global interrupt can only occur if the RS-CANFD lite module is in the Global Operation or Global Halt mode. A channel interrupt can only occur when the related CAN channel is in Channel Operation mode.

As shown in **Figure 19-46**, clear the interrupts with the software and then check that all interrupt request flags corresponding to the CAN module interrupt source are 0 before completing the interrupt processing flow.

For other CAN interrupt processing, as with this flow, make sure that all CAN interrupt sources are cleared before the interrupt processing ends.

Figure 19-46. Interruption Processing Flow (ex. RXFIFO Interrupt)



(1) Check whether the interrupt request flag is first set by the interrupt processing routine.
If the flag is not set by return from (4), it escapes from an interrupt routine by (7).

(2) RXFIFO with a request is searched.

(3) The interrupt factor flag of RXFIFO of relevance is cleared.

(4) RXFIFO of relevance confirms whether to be empty.
If it is empty, an interrupt request flag will be again returned to a check (1).

(5) Data is read from a RXFIFO buffer.

(6) The increment of the RXFIFO buffer pointer is carried out, and it returns to (4).

(7) SW escapes from an interrupt routine.

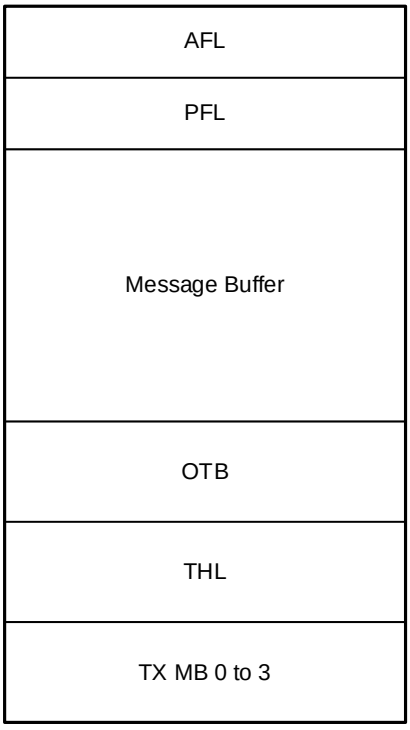
19.13 RAM Area Configuration

The RAM area used in RS-CANFD lite can be split into the following groups as shown below in **Figure 19-47**.

- AFL Rule Table area
- PFL Rule Table area
- Message Buffer area (RX MB + FIFO Buffer + Common FIFO)
- OTB area (Temporary area)
- THL area
- TX MB area

Physically the RAM is the Message Buffer RAM composed by AFL Rule Table, PFL Rule Table, Message Buffer (RX MB, RX FIFO, Common FIFO), OTB, THL, and TX MB.

Figure 19-47. RAM Area Grouping



Message Buffer RAM

The MRAM area starts with the TX-MB area at address 0000H. The TX-MB area is followed immediately by the THL area which is then followed immediately by the OTB area. The size of the TX-MB, THL and OTB area is fixed.

The OTB area is followed by the message buffer area. The message buffer area size depends on the configuration of the flat RX MBs, RXFIFOs and CFIFO. When all are configured the RX-MB area is followed by the RX FIFO area which is followed by the CFIFO area.

The configured MRAM area can be calculated then as follows:

- MRAM_cfg = AFL_MRAM_cfg
  + PFL_MRAM_cfg
  + RXMB_MRAM_cfg + RXFIFO_MRAM_cfg + CFIFO_MRAM_cfg
  + THL_MRAM_cfg
  + OTB_MRAM_cfg
  + TXMB_MRAM_cfg

  - AFL_MRAM_cfg = 256 Bytes
  - PFL_MRAM_cfg = 72 Bytes
  - RXMB_MRAM_cfg = (12 Bytes + **RMND.RMPLS**) × **RMNB.NRXMB**
  - RXFIFO_MRAM_cfg = SUM ((12 Bytes + **RFCCk.RFPLS**) × **RFCCk.RFDC**)
  - CFIFO_MRAM_cfg = (12 Bytes + **CFCC.CFPLS**) × **CFCC.CFDC**
  - THL_MRAM_cfg = 64 Bytes
  - OTB_MRAM_cfg = 160 Bytes
  - TXMB_MRAM_cfg = 304 Bytes

  “k” means RX FIFO index = [0…“number of RXFIFOs – 1”]

Caution For RFCCk.RFDC, CFCC.CFDC, RMNB.RMPLS, RMNB.NRXMB, RFCCk.RFPLS, and CFCC.CFPLS the related number of bytes must be used.

The **Table 19-23** shows the calculation of the different RAM areas used for the AFL entries, PFL entries, RX Message Buffers, RX/Common FIFOs, THL, OTB buffers, TX Message Buffers.

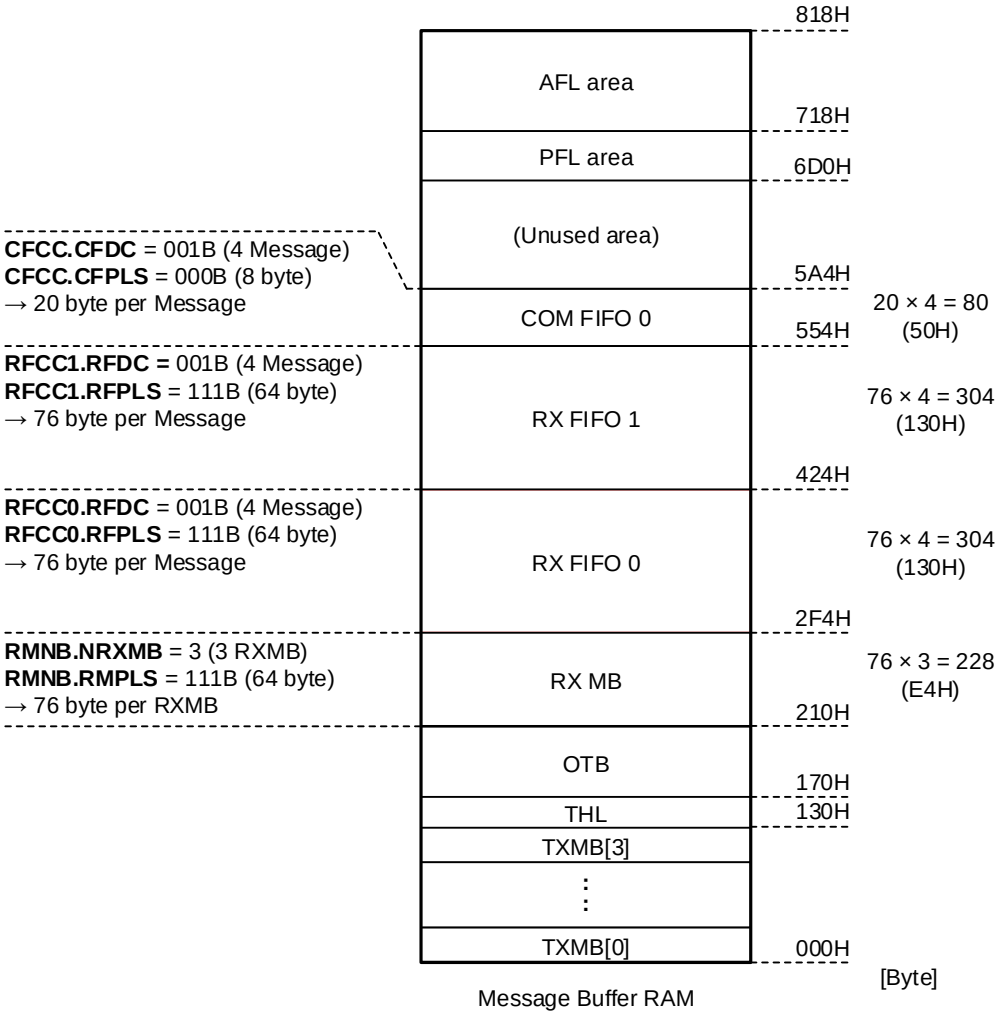
Table 19-23. MRAM Area Calculation

RAM Name	RAM Property	RAM Area Calculation Method	RAM Values
AFL	Number of rule entries	–	16
	Number of Bytes in a rule entry	Fixed	16
	Number of Bytes in AFL area	Number of rule entries × Number of Bytes in a rule entry.	256
PFL	Number of rule entries	–	2
	Number of Bytes in a rule entry	Fixed	36
	Number of Bytes in PFL area	Number of rule entries × Number of Bytes in a rule entry.	72
Message Buffer	Number of RX-MB	Fixed	16
	Number of RX FIFO	Fixed	2
	Number of Common FIFO	Fixed	1
	Average number of messages for RX-MB and FIFO buffers	–	16
	Number of Bytes for each stored message	Fixed	–
	Average size of a Message Buffer in Bytes	–	76
	Number of Bytes in Message Pool area	Average number of messages for RX-MB and FIFO buffers × Average size of a Message Buffer in Bytes.	1,216
THL	Number of entries in 1 THL buffer	Fixed	8
	Number of Bytes needed for each THL entry	Fixed	8
	Number of Bytes in THL area	Number of entries in 1 THL buffer × Number of Bytes needed for each THL entry.	64
OTB	Number of Byte OTB buffers	Fixed	160
TX-MB	Number of TX-MBs	Fixed	4
	Number of Bytes needed for each TX-MB	Fixed	76
	Number of Bytes in TX-MB area	Number of TX MBs × Number of Bytes needed for each TX-MB.	304
Total Number of Bytes Message RAM (Byte)		Number of Bytes in AFL area (256) + Number of Bytes in PFL area (72) + Number of Bytes in Message Pool area (1216) + Number of Bytes in THL area (64) + Number of Bytes in OTB area (160) + Number of Bytes in TX-MB area (304)	2,072

19.13.1 Message Buffer Area Configuration Examples

The **Figure 19-48** below shows one possible configuration.

Figure 19-48. Message Buffer Area Configuration Examples



19.13.2 RAM Initialization

The number of RAM initialization cycles and the RAM number of pages are shown below.

Table 19-24. Information for RAM Initialization

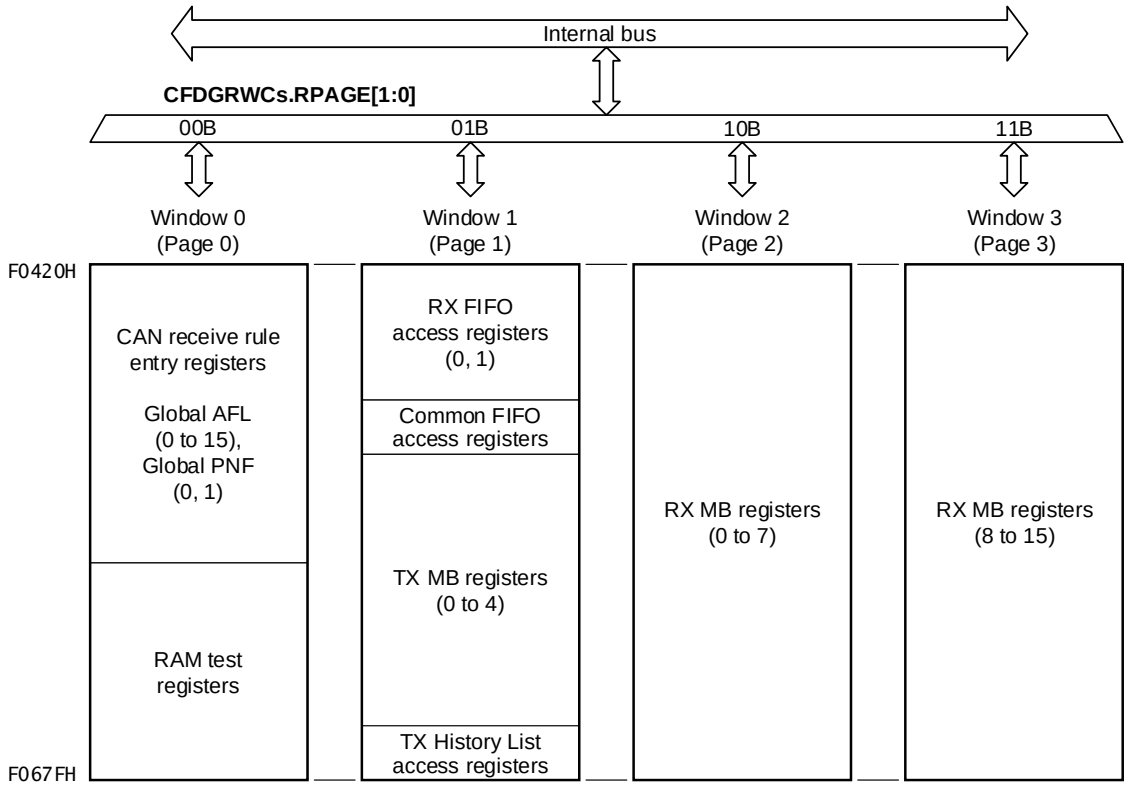
MRAM Area Size [byte]	RAM Initialization Cycles [PCLK cycle]	RAM Test RTMPS Range Note
2072	520	0H to 8H

Note Other values are prohibited.

19.14 RAM Window

The CAN RAM area from F0420H to F067FH consists of four windows. The **CFDGRWCs.RPAGE** bits are used to switch the allocation of registers.

Figure 19-49. RAM Window



19.15 Initial Settings

Before joining CAN communications the following shall be configured:

- Clock setting
- Bit timing setting (nominal and data rate)
- Baud Rate setting (nominal and data rate)
- CAN-FD setting
- Acceptance Filter setting (configuration of Global Acceptance Filter List)
- Reception-FIFO, Transmission-FIFO setting
- CAN Operation Mode setting

19.15.1 Initialization of CAN Clock, Bit Timing and Baud Rate

19.15.1.1 Bit Timing Conditions

The following lines describe the composition of each segment and the limitations that apply to the segment setting.

(1) Each segment setting

- SS = Fixed to 1 Tq
- TSEG1 = Set by **CsNCFG.NTSEG1** (nominal bit), **CsDCFG.DTSEG1** (data bit)
- TSEG2 = Set by **CsNCFG.NTSEG2** (nominal bit), **CsDCFG.DTSEG2** (data bit)
- SJW = Set by **CsNCFG.NSJW** (nominal bit) and **CsDCFG.DSJW** (data bit)
- SS + TSEG1 + TSEG2 = 5 to 49 Tq for data bit and 8 to 385 Tq for nominal bit.

(2) Limitations on TSEG1, TSEG2 and SJW

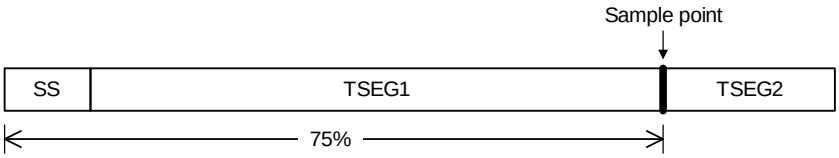
- TSEG1(N) > TSEG2(N) ≥ SJW(N)
- TSEG1(D) ≥ TSEG2(D) ≥ SJW(D)
- When only classical frames will be used, the bit fields TSEG1 (**CsNCFG.NTSEG1**) and TSEG2 (**CsNCFG.NTSEG2**) must be set to valid values.

Table 19-25 shows an example of how to set the bit timing to achieve required Sample Point settings.

Table 19-25. Bit Timing Examples

1 bit	Setting Value (Tq)				Sample Point Note [%]
	SS	TSEG1	TSEG2	SJW	
5 Tq	1	2	2	1	60.00
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
12 Tq	1	8	3	1	75.00
	1	9	2	1	83.33
15 Tq	1	10	4	1	73.33
	1	11	3	1	80.00
16 Tq	1	10	5	1	68.75
	1	11	4	1	75.00
20 Tq	1	12	7	1	65.00
	1	13	6	1	70.00
24 Tq	1	15	8	1	66.66
	1	16	7	1	70.83
50 Tq	1	39	10	4	80.00

Note Sample point (in case of 75%)



19.15.1.2 CAN Bit Timing

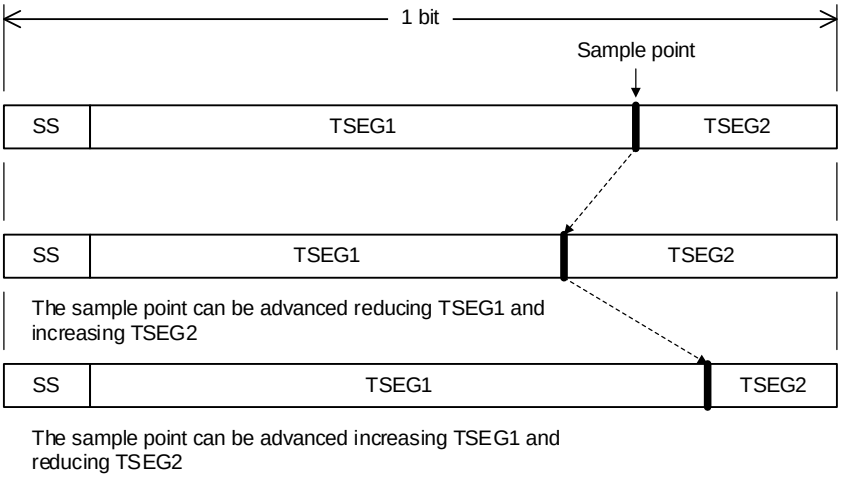
In the CAN protocol, each bit in a communication frame is composed of three segments, which can be configured individually for each channel via the related **CsNCFG** and **CsDCFG** registers.

Figure 19-50 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN Bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (Tq), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the Baud Rate Prescaler (nominal and data rate).

Figure 19-50. Segment Composition of a Bit and the Sample Point



- SS (Synchronization Segment):

  This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the Interframe Space (comprised of Intermission, Suspend Transmission, and Bus Idle; during Bus Idle, all nodes can start transmission)

- TSEG1 (Time Segment 1):

  This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.

- TSEG2 (Time Segment 2):

  This segment is used to correct a phase error by performing resynchronisation. It can be shortened by SJW (While sending or receiving a message, communication frames between some nodes may get out of sync due to a drift in the oscillator frequency or a delay in the transmission path. This is referred to as a phase error).

- SJW (Re-synchronization Jump Width):

  This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

The above figure reports only one symbolic sample point.

19.15.1.3 Baud Rate

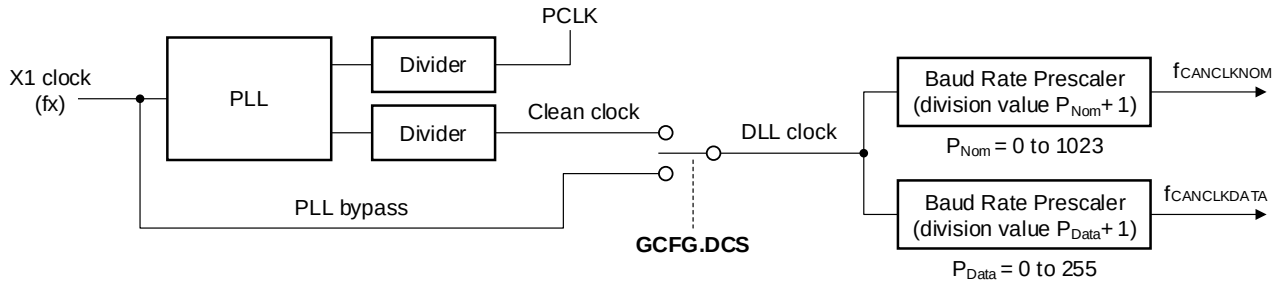
Either the CAN channel system clock (Clean clock) or the External oscillator clock can be selected globally for all CAN channels as CAN communication clock.

The transfer speed is determined by the DLL clock, the divide-by-N value of the BRP (Baud Rate Prescaler), and the number of Tq par bit, .

$$\text{Baud rate} = \frac{\text{DLL clock}}{(\text{BRP} + 1) \times (\text{number of Tq par bit})}$$

Figure 19-51 shows a block diagram of the circuit that generates the CAN channel system clock and **Table 19-26** shows a Baud Rate examples.

Figure 19-51. Block Diagram of CAN Channel Communication Clock Generation Circuit



X1: External oscillator clock
Clean clock: Peripheral clock with small clock jitter
DLL clock: Data Link Layer clock, DLL clock ≤ Peripheral clock
PCLK: CAN module system clock
P_{Nom}: Nominal baud rate prescaler division ratio set by **CsNCFG.NBRP[9:0]**
P_{Data}: Data baud rate prescaler division ratio set by **CsDCFG.DBRP[7:0]**
$f_{CANCLKNOM}$: CAN nominal communication clock, $f_{CANCLKNOM}$ = (DLL clock)/(P_{Nom} + 1)
$f_{CANCLKDATA}$: CAN data communication clock, $f_{CANCLKDATA}$ = (DLL clock)/(P_{Data} + 1)

Table 19-26. Nominal Baud Rate Calculation Formula and Example CAN Communication Configurations

Baud Rate Calculation Formula	DLL clock / (**CsNCFG.NBRP** + 1) × (Number of Tq per bit)							
	40 MHz	32 MHz	30 MHz	24 MHz	20 MHz	16 MHz	10 MHz	8 MHz Note3
1 Mbps	8 Tq (5) Note2 20 Tq (2)	8 Tq (4) 16 Tq (2)	10 Tq (3) 15 Tq (2)	8 Tq (3) 12 Tq (2) 24 Tq (1)	10 Tq (2) 20 Tq (1)	8 Tq (2) 16 Tq (1)	10 Tq (1)	8 Tq (1)
500 Kbps	8 Tq (10) 20 Tq (4)	8 Tq (8) 16 Tq (4)	10 Tq (6) 15 Tq (4) 20 Tq (3)	8 Tq (6) 12 Tq (4) 24 Tq (2)	10 Tq (4) 20 Tq (2)	8 Tq (4) 16 Tq (2)	10 Tq (2) 20 Tq (1)	8 Tq (2) 16 Tq (1)
250 Kbps	8 Tq (20) 20 Tq (8)	8 Tq (16) 16 Tq (8)	10 Tq (12) 15 Tq (8) 20 Tq (6)	8 Tq (12) 12 Tq (8) 24 Tq (4)	10 Tq (8) 20 Tq (4)	8 Tq (8) 16 Tq (4)	10 Tq (4) 20 Tq (2)	8 Tq (4) 16 Tq (2)
125 Kbps	8 Tq (40) 20 Tq (16)	8 Tq (32) 16 Tq (16)	10 Tq (24) 15 Tq (16) 20 Tq (12)	8 Tq (24) 12 Tq (16) 24 Tq (8)	10 Tq (16) 20 Tq (8)	8 Tq (16) 16 Tq (8)	10 Tq (8) 20 Tq (4)	8 Tq (8) 16 Tq (4)
83.3 Kbps	8 Tq (60) 12 Tq (40) 16 Tq (30) 24 Tq (20)	8 Tq (48) 12 Tq (32) 16 Tq (24) 24 Tq (16)	8 Tq (45) 10 Tq (36) 12 Tq (30) 15 Tq (24) 20 Tq (18) 24 Tq (15)	8 Tq (36) 12 Tq (24) 16 Tq (18) 24 Tq (12)	8 Tq (30) 10 Tq (24) 12 Tq (20) 15 Tq (16) 16 Tq (15) 20 Tq (12) 24 Tq (10)	8 Tq (24) 12 Tq (16) 16 Tq (12) 24 Tq (8)	8 Tq (15) 10 Tq (12) 12 Tq (10) 15 Tq (8) 20 Tq (6) 24 Tq (5)	8 Tq (12)
33.3 Kbps	8 Tq (150) 12 Tq (100) 16 Tq (75) 20 Tq (60) 24 Tq (50)	8 Tq (120) 10 Tq (96) 12 Tq (80) 15 Tq (64) 16 Tq (60) 20 Tq (48) 24 Tq (40)	10 Tq (90) 12 Tq (75) 15 Tq (60) 20 Tq (45)	8 Tq (90) 10 Tq (72) 12 Tq (60) 15 Tq (48) 16 Tq (45) 20 Tq (36) 24 Tq (30)	8 Tq (75) 10 Tq (60) 12 Tq (50) 15 Tq (40) 20 Tq (30) 24 Tq (25)	8 Tq (60) 10 Tq (48) 12 Tq (40) 15 Tq (32) 16 Tq (30) 20 Tq (24) 24 Tq (20)	10 Tq (30) 12 Tq (25) 15 Tq (20) 20 Tq (15)	8 Tq (30)

Table 19-27. Baud Rate Calculation Example for Nominal and Data Bitrate CAN Communication Configurations

Baud Rate Calculation Formula	$\frac{\text{(DLL clock)}}{\text{(CsDCFG.DBRP + 1) × (Number of Tq per bit)}}$	
	40 MHz	20 MHz
Nominal: 1 Mbps Data: 5 Mbps	40 Tq (1)	20 Tq (1)
	8 Tq (1)	Not possible
Nominal: 500 Kbps Data: 2 Mbps	80 Tq (1)	40 Tq (1)
	20 Tq (1)	10 Tq (1)

Caution The values in "( )" in Table 19-26 and Table 19-27 are the baud rate prescaler division values. They indicate the value of CsNCFG.NBRP setting + 1 for nominal bits and the value of CsDCFG.DBRP setting + 1 for data bits.

For optimum clock tolerance in networks using the FD frame format, the length of the time quantum should be the same in the nominal bit time and in the data bit time; means **CsNCFG.NBRP = CsDCFG.DBRP**.

Further if Transceiver Delay Compensation is used then the **CsDCFG.DBRP** shall not be programmed greater than 1; 1 means divide by 2.

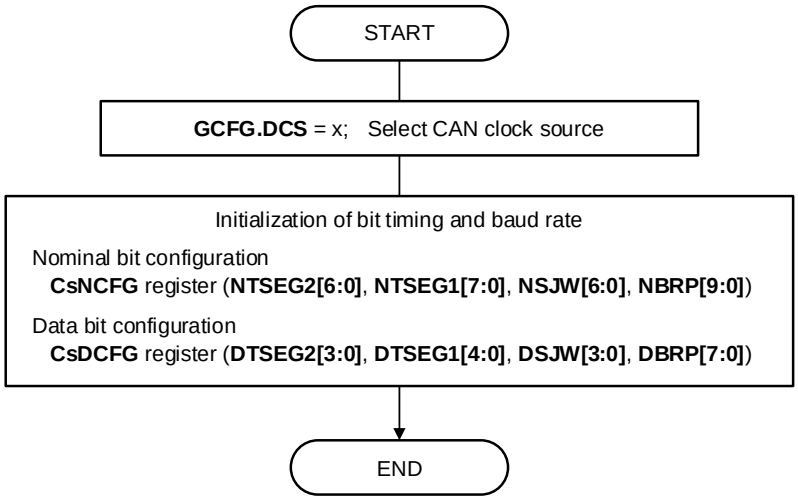
19.15.1.4 Setting of CAN Clock, Bit Timing and Baud Rate

Figure 19-52 shows the procedure for setting the CAN clock and the Baud Rate for each channel.

These settings should be performed during Channel Reset Mode (Configuration Mode) for the CAN channels.

Before going to channel communication state the Baud Rate must be configured, otherwise the mode will not switch correctly.

Figure 19-52. Procedure for Setting the CAN Bit Timing and Baud Rate



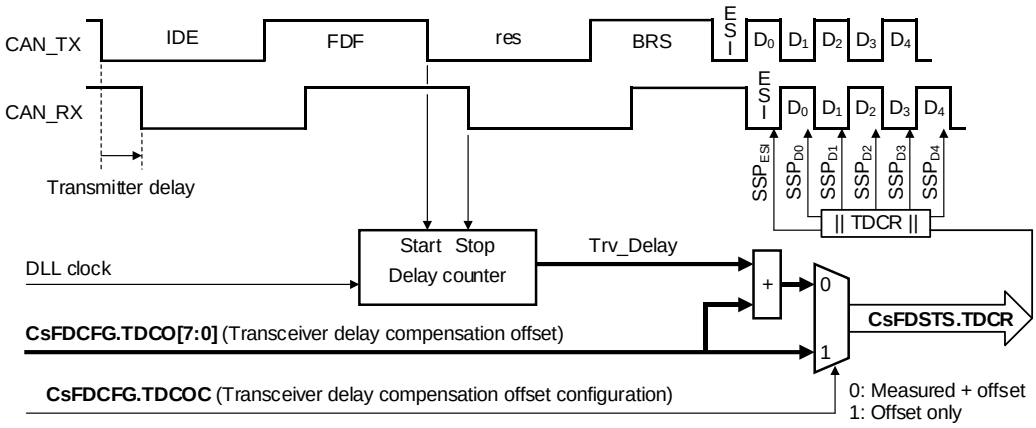
19.15.1.5 CAN Transmitter Delay Compensation

In case a high Baud Rate is used such as 5 Mbps for the Data phase, the Transmitter delay could become greater than TSEG1. In this case the transmitter would always detect a bit-error in the data-phase of the CAN-FD frame. The TDC (Transceiver Delay Compensation) compensates the transmitter's inability to receive its own transmitted bit at the sample point of that bit.

There is another symbolic sample point known as the SSP (Secondary Sample Point) that is used only during the Data phase of CAN-FD frames. This is derived from the Transceiver Delay Compensation Result (**CsFDSTS.TDCR**) as shown in **Figure 19-53**.

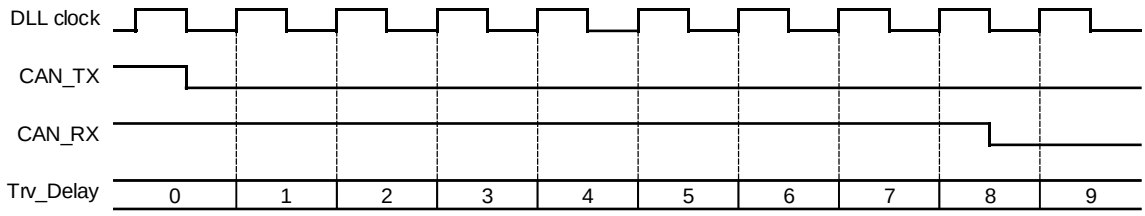
The resolution of the configuration, measured and offset values is based on the CAN channel DLL clock.

Figure 19-53. Transmitter Delay Compensation



The measured Trv_Delay is based on integer number of DLL clock cycles. The delay is counted up by one for each started clock until the dominant value is seen on CAN_RX. The figure below illustrates the measured result. Trv_Delay counted to maximum 127 with a DLL clock.

Figure 19-54. Trv_Delay Measurement Example



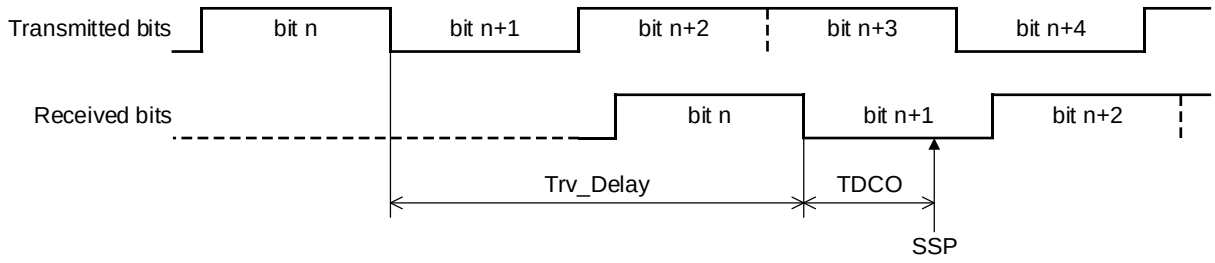
The SSP (Secondary Sample Point) is calculated by taking the result from **CsFDSTS.TDCR** and rounding the value down to the nearest integer number of data time quanta.

The **Figure 19-55** illustrates the positioning of the secondary sample point. In case the **CsFDCFG.TDCOC** is equal to 0 the SSP is equal to the Trv_Delay (measured delay) + **CsFDCFG.TDCO**, rounded down to the nearest integer number of time quanta. Usually the **TDCO** value should have the size of (SyncSegment $_{data}$ + TSEG1$_{data}$) to position the SSP to the theoretical location of a sample point.

If the **CsFDCFG.TDCOC** is 1 then the SSP is just defined by the **CsFDCFG.TDCO**.

If the **CsDCFG.DBRP** > zero then the value will also be rounded down to the nearest integer number of time quanta.

Figure 19-55. Position of the Secondary Sample Point



The maximum delay (Trv_Delay + TDCO) which can be compensated by the RS-CANFD lite module is “6 data bits – 2 × DLL clock”.

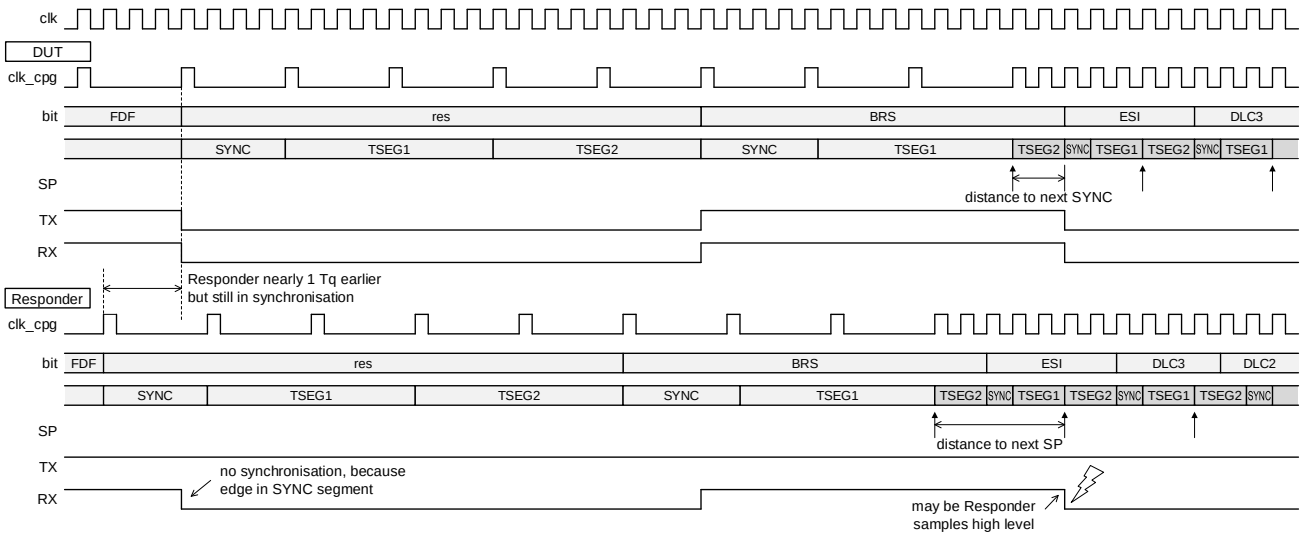
The ISO 11898-1 allows BRP_data and BRP_nom to be set to different values.

If different values are used for **CsNCFG.NBRP** and **CsDCFG.DBRP**, then 2 CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after Sample point of the BRS bit. This condition is shown in **Figure 19-56** below.

The length of the time quantum should be the same in the nominal bit time and in the data bit time; means **CsNCFG.NBRP** = **CsDCFG.DBRP**.

Different Bitrates can be achieved by selecting different configuration values for the Time Segments. Note the Nominal Bitrate can be configured from 8 to 385 Tq and the Data Bit rate from 5 to 49 Tq.

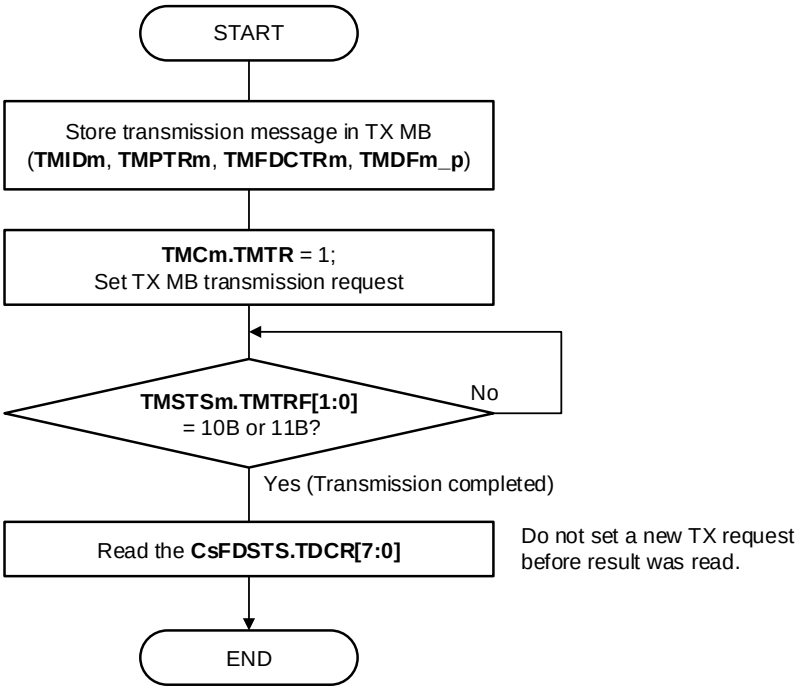
Figure 19-56. Loss of Synchronization Between 2 CAN Nodes



The transmitter delay compensation measurement result is updated at the falling edge from FDF bit to res bit when configured accordingly (**CsFDCFG.TDCE** = 1, **CsFDCFG.TDCOC** = 0)

Figure 19-57 shows the read flow to get the measured transmitter delay compensation result.

Figure 19-57. TDC Result Read Flow



19.15.2 CAN Module Configuration After Reset

After CAN reset (**PER2.CANsEN** bit is set to 1 from 0) or after setting and clearing a **GRSTC.SRST** bit, the RS-CANFD lite module enters Global Sleep Mode automatically.

To enable configuration of the RS-CANFD lite module, the Sleep Mode has to be left by clearing the Global Sleep Request bit **GCTR.GSLPR** to 0.

After CAN reset the module starts the RAM initialization, the **GSTS.GRAMINIT** bit is set automatically indicating that the RS-CANFD lite logic is initializing the RAM.

After the RAM initialization is completed, this bit is cleared automatically.

The RAM initialization is necessary to avoid setting of false ECC error flag after CAN reset due to random data present in the RAM.

The registers of RS-CANFD lite should not be accessed (in either read or write) until the RAM initialization is complete and the **GSTS.GRAMINIT** bit is cleared.

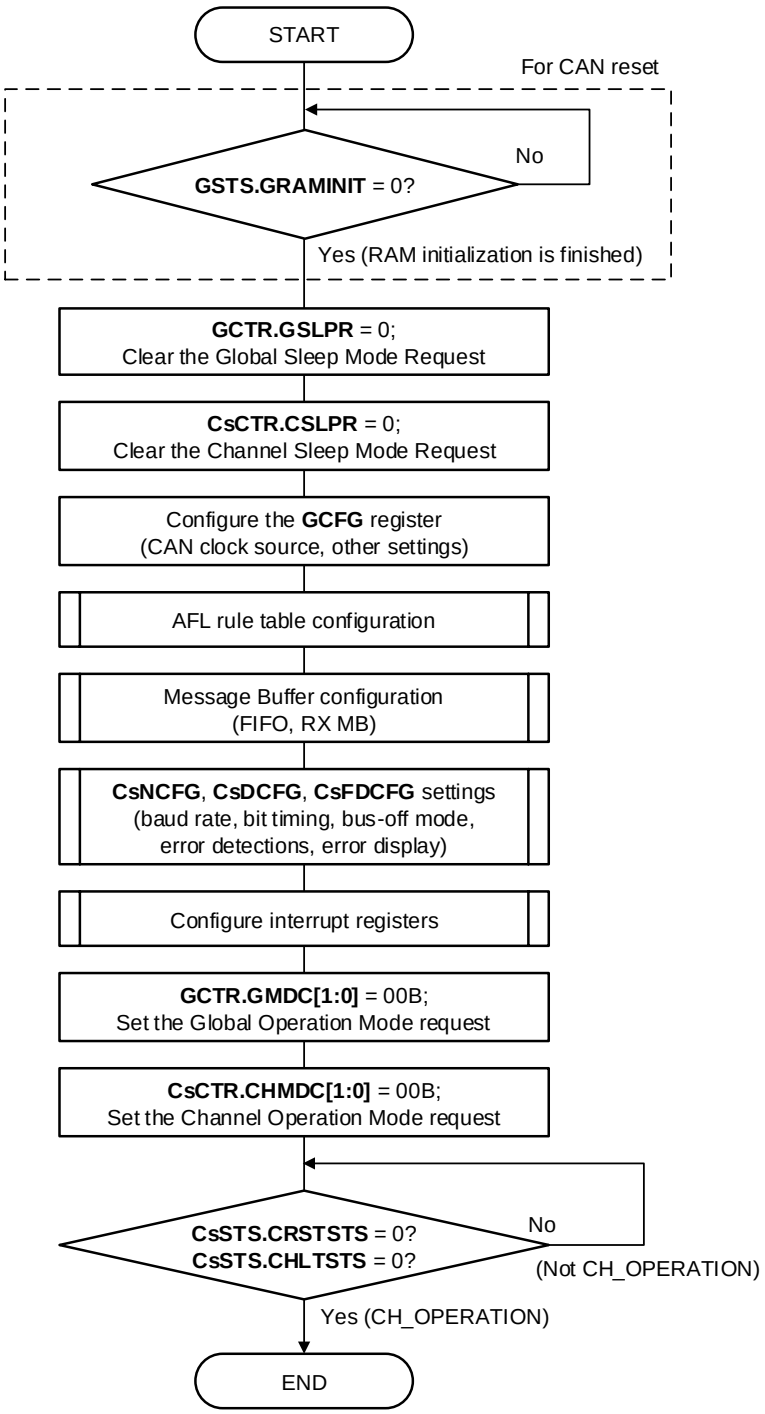
Before going to communication mode the Global Acceptance Filter List and message FIFO buffers must be configured. Furthermore each required CAN channel has to be configured (e.g. CAN bit timing)

For this all required CAN channels have to be released from channel Sleep Mode and have to be configured for communication in channel Reset Mode (Configuration Mode).

Figure 19-58 shows the configuration procedure. For details about each step, refer to **19.7 Acceptance Filtering Function using Global Acceptance Filter List (AFL)**, **19.8 FIFO Buffers and Normal MB Configuration**, **19.12 Interrupt**, and **19.15.1.3 Baud Rate**.

The RS-CANFD lite module does not perform the RAM initialization sequence after executing SW Reset by setting **GRSTC.SRST.**

Figure 19-58. Initial Procedure After Reset



19.16 Pretended Network Filter (PNF)

The purpose of this function is to reduce the current consumption.

If without the Pretended Network (PN) mode, CPU check all frames which passed ID filter and decides application processing.

Therefore, CPU is always active.

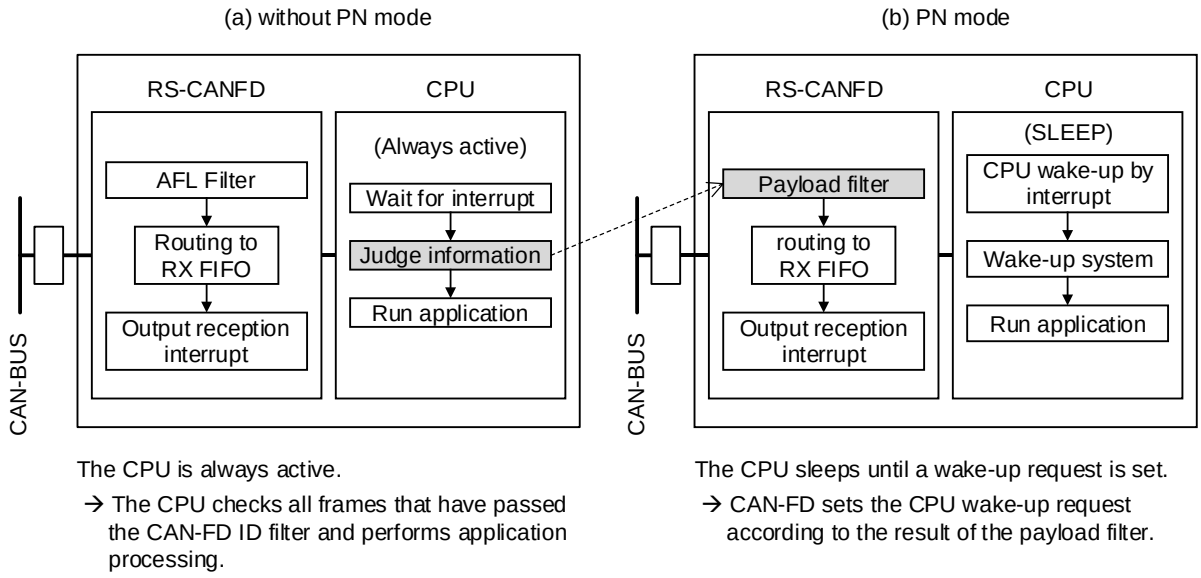
In PN mode, RS-CANFD lite module generate wake-up request for CPU by result of payload filter.

In this case, the RS-CANFD lite module process a part of CPU processing.

Therefore, CPU can sleep and reduce its current consumption.

RSCAN-FD controller compare a payload with the payload filter, and if successful, CPU wake-up by CAN reception Interrupt.

Figure 19-59. Overview of Processing in PN Mode



The CPU is always active.

→ The CPU checks all frames that have passed the CAN-FD ID filter and performs application processing.

The CPU sleeps until a wake-up request is set.

→ CAN-FD sets the CPU wake-up request according to the result of the payload filter.

19.16.1 Explanation in Each Mode

PNF compares ID and payload value of a receiving frame with a filter value.

The PNF operates independently from the AFL.

There are four kinds of combination of filtering of PNF and AFL.

The mode can be selected by **CsFDCTR.PNMDC** bit.

The current mode can be confirmed by **CsFDSTS.PNSTS** bit.

Table 19-28. Comparison of PN Mode and Normal Mode

Mode / Filter	Pretended Network Filter Mode (PN Mode)	Normal Mode		
		Pretended Network Filter ID only and Acceptance Filter Mode	Acceptance Filter Mode	Pretended Network Filter and Acceptance Filter Mode
AFL Filter	OFF	ON	ON	ON
PN ID Filter	ON	ON	OFF	ON
PN Payload Filter	ON	OFF	OFF	ON
CsFDCTR.PNMDC	11B	01B	00B	10B
CsFDSTS.PNSTS				

19.16.1.1 Pretended Network Filter Mode

This mode is for CPU wake-up from CPU HALT mode.

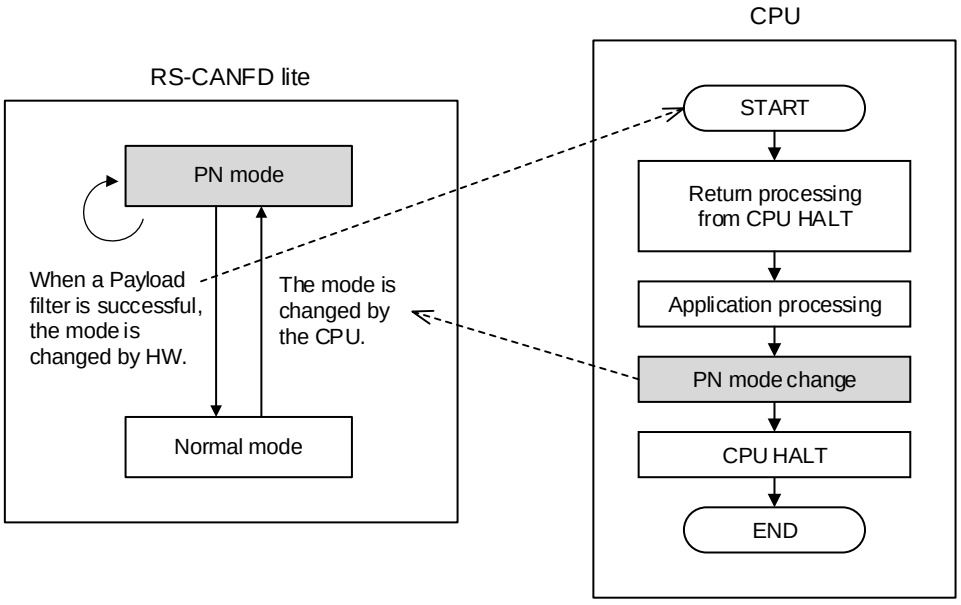
In PN mode, received ID and received payload are compared with PN filter.

In this mode AFL is not performed.

If ID and payload matches PN filter, this mode changes to the mode automatically set up in the **CsFDCFG.RPNMD** bit.

Priority is given to mode transition of RS-CANFD lite when the mode transition from RS-CANFD lite and the mode transition from CPU compete.

Figure 19-60. Processing Flow in PN Mode



19.16.1.2 Pretended Network Filter ID only and Acceptance Filter Mode

In this mode, ID of a receiving frame is filtered using AFL and PNF.

Comparison of a payload is not performed.

When a filter matches by both AFL and PNF, routing is carried out based on the information on PNF, and routing of AFL is not performed.

Table 19-29. Match/Unmatch Action of Filter (Pretended Network Filter ID only and Acceptance Filter Mode)

AFL	PNF		Routing
ID Filter	ID Filter	Payload Filter	
unmatch	unmatch	not perform	Receive message is rejected
not perform	match		Use PNF information
match	unmatch		Use AFL information

19.16.1.3 Acceptance Filter Mode

In this mode, ID of a receiving frame is filtered only by AFL.

The PNF is not performed.

Table 19-30. Match / Unmatch Action of Filter (Acceptance Filter Mode)

AFL	PNF		Routing
ID Filter	ID Filter	Payload Filter	
unmatch	not perform	not perform	Receive message is rejected
match			Use AFL information

19.16.1.4 Pretended Network Filter and Acceptance Filter Mode

In this mode, ID of a receiving frame is filtered using AFL and PNF.

PNF compares the payload of a receiving frame.

When a filter matches by both AFL and PNF, priority is given to PNF and then a payload is compared.

Routing will be performed if a payload filter is matches.

Received messages is rejected when a filter is unmatching.

Table 19-31. Match / Unmatch Action of Filter (Pretended Network Filter and Acceptance Filter Mode)

AFL	PNF		Routing
ID Filter	ID Filter	Payload Filter	
unmatch	unmatch	not perform	Receive message is rejected
not perform	match	unmatch	Receive message is rejected
not perform	match	match	Use PNF information
match	unmatch	not perform	Use AFL information

19.16.2 Details of a Filtering Function

19.16.2.1 PN ID Filter

PN ID filter is the same operation as ID filter of AFL.

However, PFL (Pretended Network Filter List) does not correspond to an AFL rewriting function (updating AFL entry during communication).

Table 19-32 shows the behavior of the Pretended network filter unit depending on the setting of the related input signals.

Table 19-32. Pretended Network Filter Behavior Based on Loopback Configuration Setting in PNF Entry

Mirror Mode Enable (**GCFG.MME**)	Channel Test Mode Select (**CsCTR.CTMS**)	Channel Mode	Global Pretended Network Filter List Entry Loopback Configuration (**GPFLIDj.GPFLLB**)	PNF
0 (Mirror mode disabled)	≠ 10B, 11B	Receive	0	valid
			1	invalid
		Transmission	0	invalid
			1	invalid
	= 10B, 11B (Loopback mode)	Receive	0	valid
			1	invalid
		Transmission	0	valid
			1	valid
1 (Mirror mode enabled)	≠ 10B, 11B	Receive	0	valid
			1	invalid
		Transmission	0	invalid
			1	valid
	= 10B, 11B (Loopback mode)	Receive	0	valid
			1	invalid
		Transmission	0	valid
			1	valid

19.16.2.2 PN Payload Filter

Payload filter compares 32-bit data at two places.
A comparison position is chosen by the preset value of offset. (Offset position is 4 bytes boundaries.)
Receiving frame is rejected when DLC is less than four.
Moreover, which also rejects a remote frame.
When the range specified by offset exceeds payload length, payload filter rejects a frame.
There are two filter in order to compare to two 32-bit data.
There are the two comparison methods in each filter.
A match filter compares a payload per bit.

An upper and lower limit filter confirms that the value of a payload is within range of upper and lower limit.
Or an upper and lower limit filter confirms that the value of a payload is out of range of upper and lower limit.
The filter 0 and the filter 1 can select the different comparison method.

- The case considered as a pass by match of both the filter 0 and the filter 1.
- The case considered as a pass by match of either one of the filter 0 or the filter 1.

A filter can be selected from these two cases.

The conditions within the range of an upper and lower limit filter are shown below.

Lower limit ≤ payload value ≤ upper limit

The conditions out the range of an upper and lower limit filter are shown below.

Payload value < lower limit or upper limit < payload value

19.16.2.3 Setup of the Number of Rules of PNF

PNF mounts RAM for exclusive use aside from AFL and stores a rule.
The number of rules becomes 2 rules.
The number of rules can be assigned in the 0 to 2 ranges.
9 registers per one rule are needed.

Figure 19-61. Configuration of PNF for Each Channel

1 entry = 9 longword = 36 (24H) byte

1	GPFLIDj	ID filter	for ID filter (same as AFL)
2	GPFLMj	ID mask	
3	GPFLP0j	Label, DLC, RXMB	
4	GPFLP1j	FIFO	
5	GPFLPTj	Filter setting	for Payload filter
6	GPFLPD0j	Filter 0	
7	GPFLPM0j		
8	GPFLPD1j	Filter 1	
9	GPFLPM1j		

RAM for PN filter

entry 0
entry 1

[j = 0, 1] : Global PNF register number

19.16.2.4 Software Flow of a Payload Filter

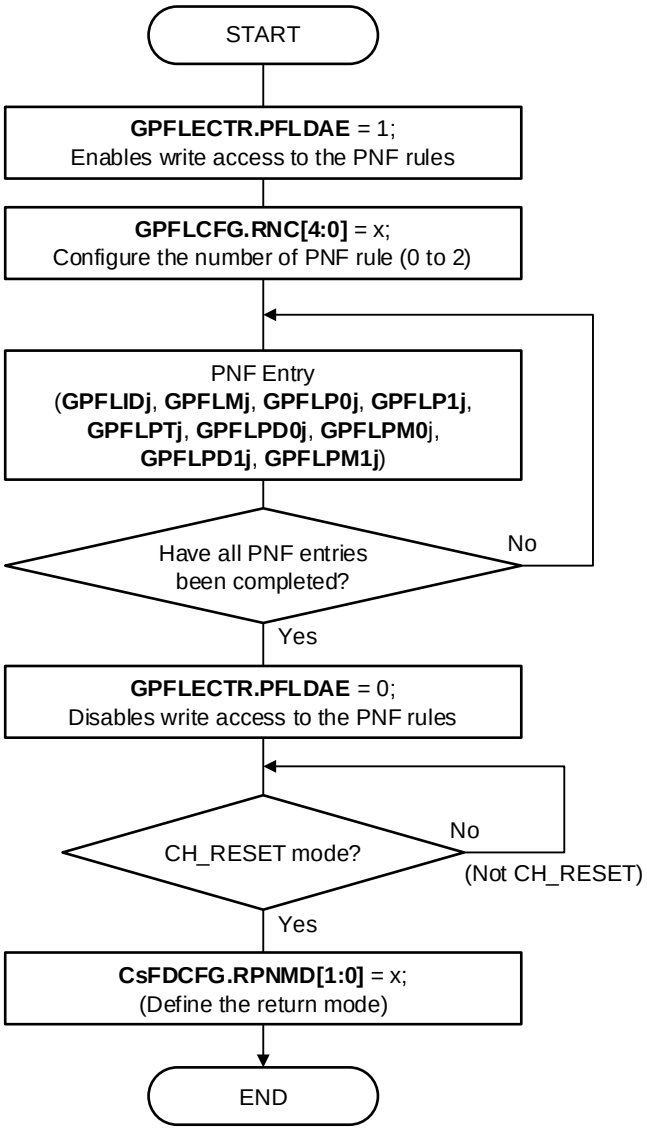
The configuration flow shown in **Figure 19-62** should be followed to program the PFL.

After entering all entries in Configuration Mode, locking of the PNF access should be performed to protect unwanted write access to the PFL.

Write protection is active during all global modes if the lock bit is set (GL_RESET, GL_HALT, and GL_OPERATION).

Read access to PFL is still possible during all global modes even if PFL data access is disabled (consistency check of PFL contents possible during runtime).

Figure 19-62. PNF Configuration Flow



Next, the software flow to operation in PN mode is shown below.

The interrupt return from PN mode becomes reception interrupt or Global error interrupt.

When the received payload filter frame matches ID filter and a payload filter, it is stored in RXFIFO etc.

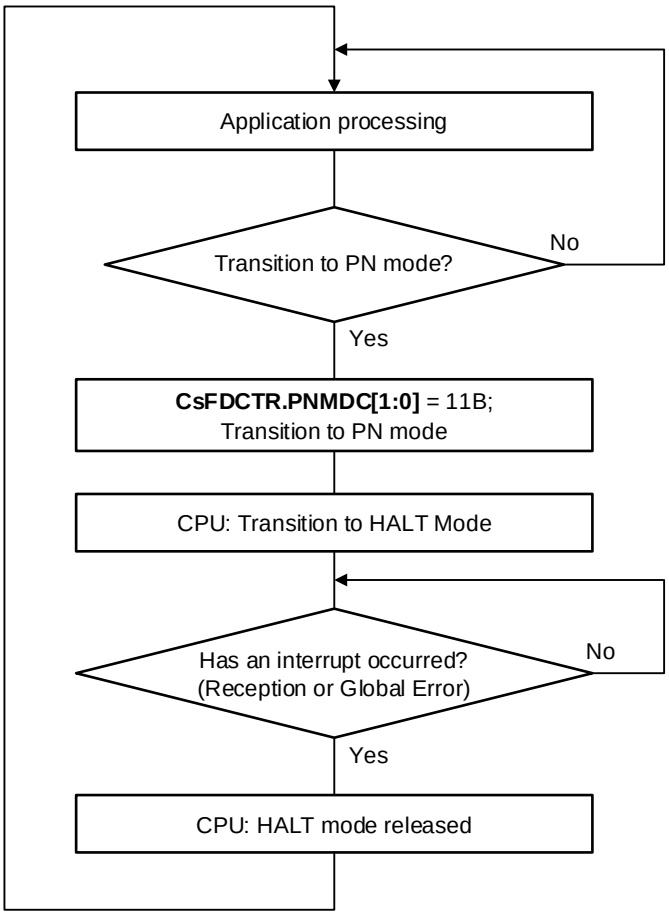
Reception interrupt will be generated if a receiving frame is stored in FIFO.

When the error of a DLC check or payload overflow occurs after matching ID filter and a payload filter, receiving frame is not stored in FIFO.

When these errors occur, Global error interrupt is generated.

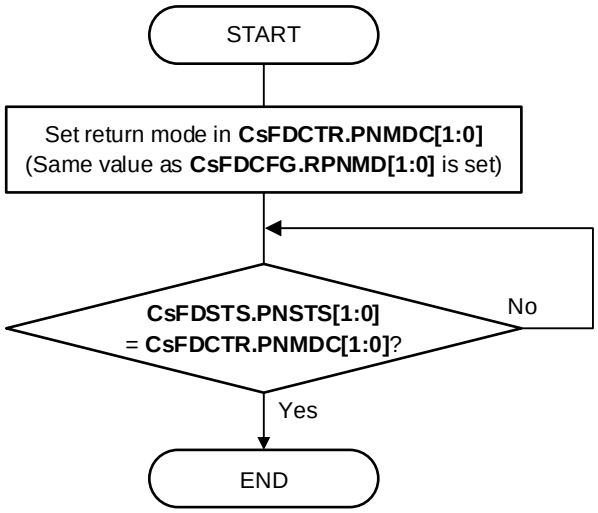
CPU transitions to normal status from CPU HALT status by these two interrupts.

Figure 19-63. PNF Operation Flow



The transition to a normal mode from PN mode is shown below.
This is used when it cannot return from PN mode.
Mode transition will be performed if it writes in **CsFDCTR.PNMDC[1:0].**
CsFDCTR.PNMDC[1:0] and **CsFDCFG.RPNMD[1:0]** bits must write the same value.

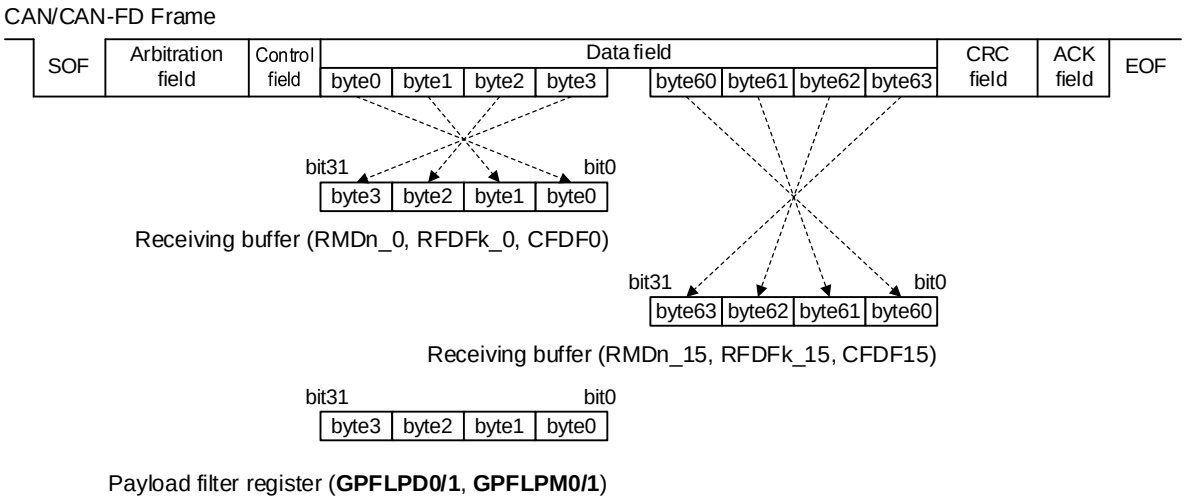
Figure 19-64. Normal Mode Transition



19.16.2.5 Byte Alignment of Payload Filter

Byte alignment of the filter register (**GPFLPD0/1**, **GPFLPM0/1**) of an upper and lower limit filter is explained.
The CAN communications system consists of little endians. Moreover, RS-CANFD lite module is little endian structure, endian is matching. Therefore, the endian of a register is as follows.

Figure 19-65. Relationship Between CAN/CANFD Frame Data and Registers for Byte Endianness



An upper and lower limit filter compares the data of this receiving buffer and the contents of the payload filter register.

19.17 Notes on the CAN Module

- When linking transmit buffers to Common FIFO buffers, set the control register (**TMCm**) of the corresponding transmit buffer to 00H. The status register (**TMSTSm**) 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 Common FIFO buffers remain unchanged. Set the enable bit in the corresponding interrupt enable register (the **TMIEC** register) to 0 (transmit buffer interrupt is disabled).

- For the clock source setting for CAN Module (except Global Sleep Mode), refer to the following:
  - f_{MP} and f_{CLK} for CAN module are should be supplied with same clock source. Also, select $f_{MP}/2$ in the f_{MP} Clock Division Register (MDIV) so that $f_{CLK} = f_{MP}/2$
  - The high-speed on-chip oscillator can be used for f_{MP} and f_{CLK}, but it's prohibited for f_{CAN} (communication clock).
  - When using X1 clock for f_{CAN} (communication clock) with setting of **GCFG.DCS** bit = 1, the frequency must be equal or less than f_{CLK}.

- Interrupt processing of RS-CANFD lite

  The interrupt flags in RS-CANFD lite (such as the RFIF bit in the RFSTS0s register) are not cleared automatically when an interrupt occurs, so all these interrupt flags must be cleared within the interrupt processing. See **19.12.2 Interrupt Processing Flow** for details.

- Some interrupts of channel 1 (CAN1) share the vector table with interrupts from other peripheral functions. To determine the interrupt of CAN1 function, use the interrupt determination flag of the INTFLG3 register. For details, see **22.4.1 Determination of Interrupt Source**.

CHAPTER 20 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.

20.1 Overview

Table 20-1 lists the DTC specifications.

Table 20-1. DTC Specifications

Item		Specification
Activation sources		RL78/F25: 49 sources max, RL78/F22: 37 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 or HDTCCTm registers to change from 1 to 0, the repeat area address is initialized, the value of DTRLDj is reloaded to the DTCCTj register, or the value of HDTRLDm is reloaded to the HDTCCTm register, and the DTC continues the transfer.
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 20-5. 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). • 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). • 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

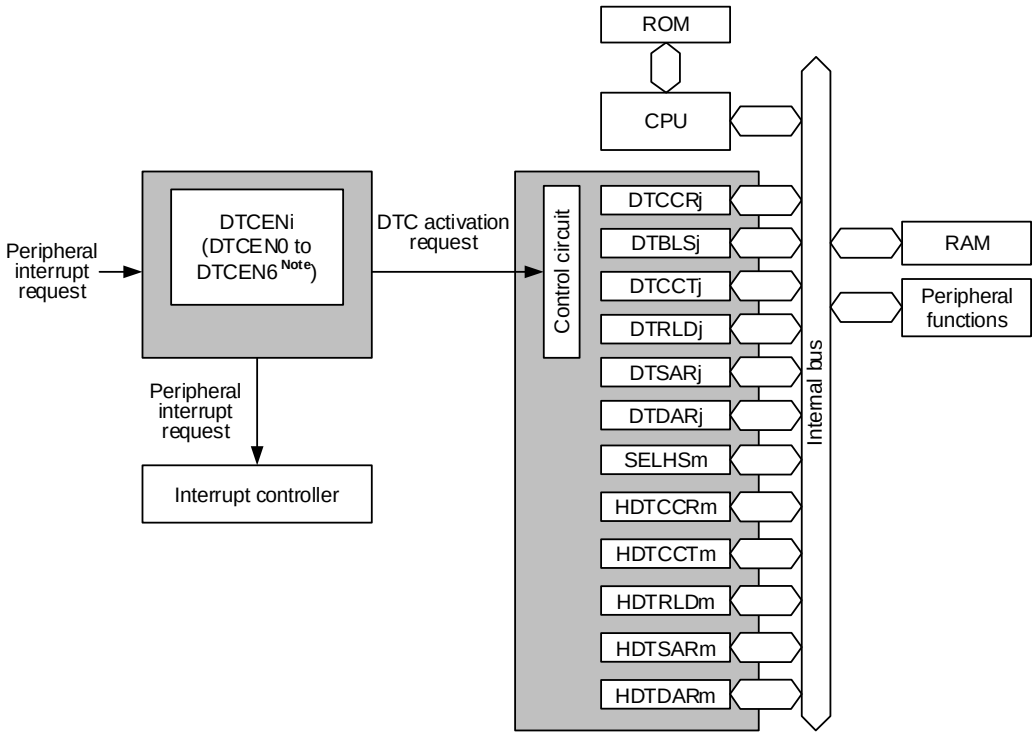
(**Notes** and **Remark** are listed on the next page.)

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 RL78/F22: i = 0 to 4, j = 0 to 23, m = 0, 1
Products of RL78/F25: i = 0 to 6, j = 0 to 23, m = 0, 1

Figure 20-1. DTC Block Diagram



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 DTCEN6 Note: DTC activation enable registers 0 to 6 Note
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

Note Products of RL78/F22 do not have the DTCEN5 and DTCEN6 registers.
Remark RL78/F22: i = 0 to 4, j = 0 to 23, m = 0, 1
RL78/F25: i = 0 to 6, j = 0 to 23, m = 0, 1

20.2 Registers

Tables 20-2 and **20-4** list the DTC register configuration.

Table 20-2. DTC Register Configuration (1)

Address	Register Name	Symbol	After Reset	Access Size
F02C1H	Peripheral enable register 1	PER1	00H	1, 8
F02E0H	DTC base address register	DTCBAR	FDH	8
F02E8H	DTC activation enable register 0	DTCEN0	00H	1, 8
F02E9H	DTC activation enable register 1	DTCEN1	00H	1, 8
F02EAH	DTC activation enable register 2	DTCEN2	00H	1, 8
F02EBH	DTC activation enable register 3	DTCEN3	00H	1, 8
F02ECH	DTC activation enable register 4	DTCEN4	00H	1, 8
F02EDH	DTC activation enable register 5 Note	DTCEN5	00H	1, 8
F02EEH	DTC activation enable register 6 Note	DTCEN6	00H	1, 8

Note Only in products of RL78/F25.

Table 20-3 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 20-3. 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

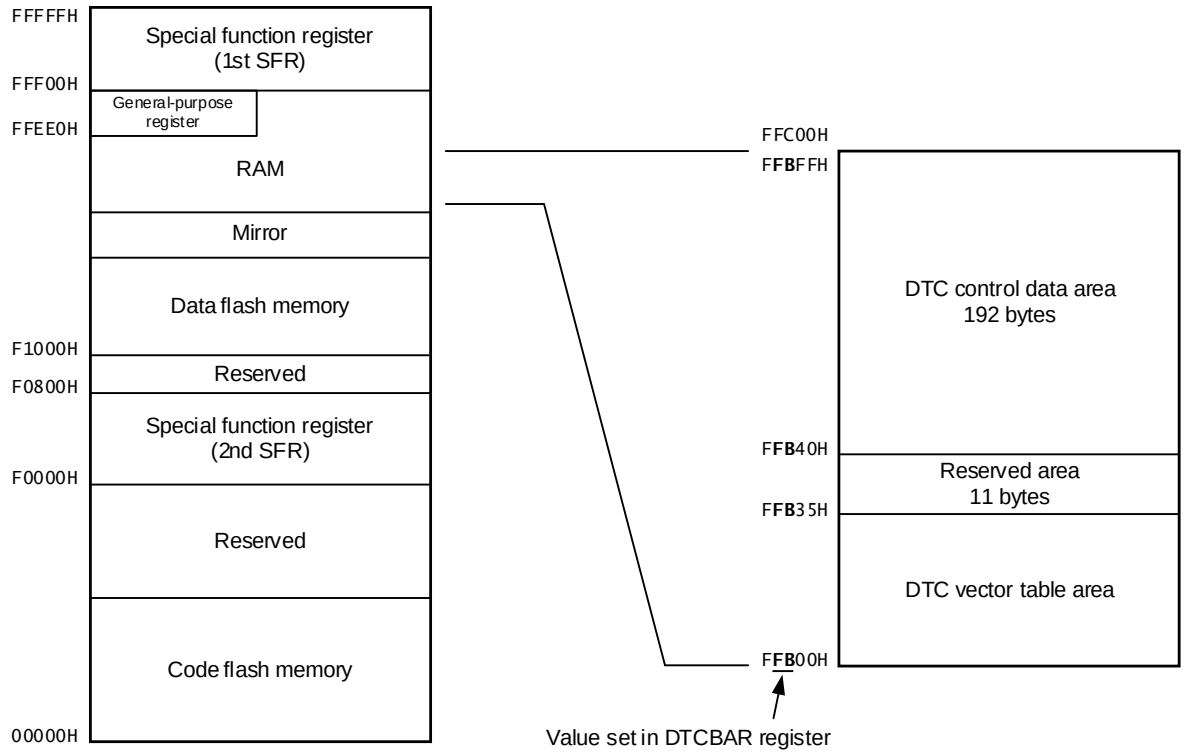
Table 20-4. DTC Register Configuration (2)

Address	Register Name	Symbol	After Reset	Access Size
F02D0H	High-speed DTC control register 0	HDTCCR0	00H	1, 8
F02D2H	High-speed DTC transfer count register 0	HDTCCT0	00H	1, 8
F02D3H	High-speed DTC transfer count reload register 0	HDTRLD0	00H	1, 8
F02D4H	High-speed DTC source address register 0	HDTSAR0	0000H	16
F02D6H	High-speed DTC destination address register 0	HDTDAR0	0000H	16
F02D8H	High-speed DTC control register 1	HDTCCR1	00H	1, 8
F02DAH	High-speed DTC transfer count register 1	HDTCCT1	00H	1, 8
F02DBH	High-speed DTC transfer count reload register 1	HDTRLD1	00H	1, 8
F02DCH	High-speed DTC source address register 1	HDTSAR1	0000H	16
F02DEH	High-speed DTC destination address register 1	HDTDAR1	0000H	16
F02E1H	High-speed DTC channel select register 0	SELHS0	3FH	1, 8
F02E2H	High-speed DTC channel select register 1	SELHS1	3FH	1, 8

20.2.1 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 20-2 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 20-2. 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 (FFEE0H to FFEFFH) space as the DTC control data area or DTC vector table area.

2. The 11-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 used by the Hot plug-in, RRM (real-time RAM monitor)/DMM (dynamic memory modification), or On-chip trace functions cannot be used as the DTC control data area or DTC vector table area when those functions are used. For details, see Table 34-1. RAM Area Used and Number of Branches Retained by On-chip Trace in CHAPTER 34 ON-CHIP DEBUG FUNCTION.

20.2.2 DTC Control Data Allocation

Control data is allocated beginning with each start address in the order: Registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj.

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.

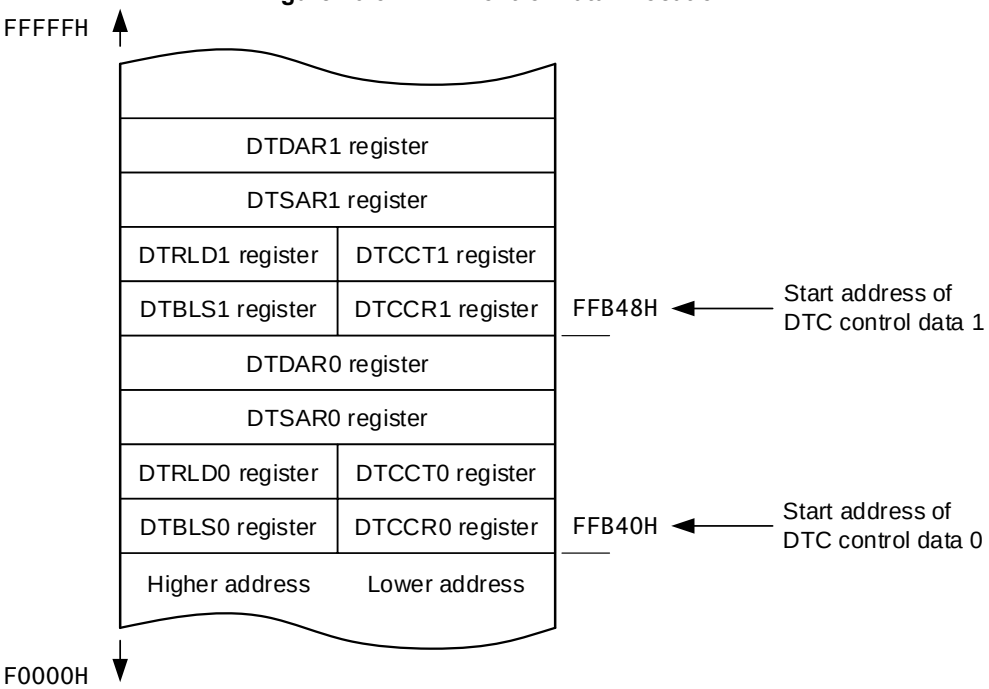
Figure 20-3 shows an example of DTC control data allocation when the DTCBAR register is set to FBH.

Cautions 1. Change the data in registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi register is set to 0 (DTC activation disabled).

2. Do not access DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj using a DTC transfer.

Remark Products of RL78/F22: i = 0 to 4, j = 0 to 23
Products of RL78/F25: i = 0 to 6, j = 0 to 23

Figure 20-3. DTC Control Data Allocation



20.2.3 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 20-5 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 34H 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 in the DTCENi register is set to 0 (DTC activation disabled).

Remark Products of RL78/F22: i = 0 to 4
Products of RL78/F25: i = 0 to 6

Table 20-5. 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 Note 1	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	6	Address set in DTCBAR register +06H	
INTP6 Note 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 Note 3	12	Address set in DTCBAR register +0CH	
UART1 transmission transfer end/CSI10 transfer end or buffer empty/IIC10 transfer end	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	
Configuration register write request for each CTSU channel	16	Address set in DTCBAR register +10H	
CTSU measurement data transfer request	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 1. Not provided in the products with 24-pins.

2. Not provided in the products with 24-, 32-pins.

3. RL78/F22 24-, 32-pins products and RL78/F25 32-pins products do not have CSI11 transfer end or buffer empty interrupt/IIC11 transfer end interrupt.

Table 20-5. 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	
Reserved	35	Address set in DTCBAR register +23H	
Reserved	36	Address set in DTCBAR register +24H	
End of channel 1 of timer array unit 1 count or capture	37	Address set in DTCBAR register +25H	
End of channel 2 of timer array unit 1 count or capture	38	Address set in DTCBAR register +26H	
End of channel 3 of timer array unit 1 count or capture	39	Address set in DTCBAR register +27H	
Comparator detection 0 Note	40	Address set in DTCBAR register +28H	
End of channel 0 of timer array unit 1 count or capture Note	41	Address set in DTCBAR register +29H	
End of channel 4 of timer array unit 1 count or capture Note	42	Address set in DTCBAR register +2AH	
End of channel 5 of timer array unit 1 count or capture Note	43	Address set in DTCBAR register +2BH	
End of channel 6 of timer array unit 1 count or capture Note	44	Address set in DTCBAR register +2CH	
End of channel 7 of timer array unit 1 count or capture Note	45	Address set in DTCBAR register +2DH	
LIN1 reception end Note	46	Address set in DTCBAR register +2EH	
LIN1 transmission start/end Note	47	Address set in DTCBAR register +2FH	
LIN2 reception end Note	48	Address set in DTCBAR register +30H	
LIN2 transmission start/end Note	49	Address set in DTCBAR register +31H	
CAN0 reception end Note	50	Address set in DTCBAR register +32H	
CAN1 reception end Note	51	Address set in DTCBAR register +33H	
Reserved	52	Address set in DTCBAR register +34H	Lowest

Note Only in RL78/F25 products.

20.2.4 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 20-4. Format of Peripheral Enable Register 1 (PER1)

Address: F02C1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER1	DACEN Note 1	CTSUEN	CMPEN Note 1	TRD0EN Note 2	DTCEN	PWMOPEN	0	TRJ0EN

DTCEN	Control of DTC input clock supply
0	Stops input clock supply. • DTC cannot run.
1	Enables input clock supply. • DTC can run.

Notes 1. Only in RL78/F25.

2. When FRQSEL4 = 1 in the user option byte (000C2H/040C2H), 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/F22: bits 1, 5, and 7
RL78/F25: bit 1

20.2.5 DTC Activation Enable Register i (DTCENi)

This is an 8-bit register which enables or disables DTC activation by interrupt sources. **Table 20-6** 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.

Cautions 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.

Remark Products of RL78/F22: i = 0 to 4
Products of RL78/F25: i = 0 to 6

Figure 20-5. DTC Activation Enable Register i (DTCENi)

Address: F02E8H (DTCEN0), F02E9H (DTCEN1), F02EAH (DTCEN2), F02EBH (DTCEN3), F02ECH (DTCEN4), F02EDH (DTCEN5) Note, F02EEH (DTCEN6) Note After reset: 00H

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
DTCENi	DTCENi7	DTCENi6	DTCENi5	DTCENi4	DTCENi3	DTCENi2	DTCENi1	DTCENi0

Note Only in RL78/F25 products.

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.		

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.		

Table 20-6. 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	Configuration register write request for each CTSU channel	CTSU measurement data transfer request	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 0 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 RDe compare match A0	Timer RDe compare match B0	Timer RDe compare match C0	Timer RDe compare match D0	Timer RDe compare match A1	Timer RDe compare match B1
DTCEN4	Timer RDe compare match C1	Timer RDe compare match D1	Timer RJ0	Reserved	Reserved	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	Comparator detection 0	End of channel 0 of timer array unit 1 count or capture	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	LIN1 reception end	LIN1 transmission start/end
DTCEN6 Note	LIN2 reception end	LIN2 transmission start/end	CAN0 reception end	CAN1 reception end	Reserved	Reserved	Reserved	Reserved

Note Only in RL78/F25 products.

Remark RL78/F22 products: i = 0 to 4
RL78/F25 products: i = 0 to 6

Caution Be sure to set the reserved bit to 0. Reserved bit always returns 0 when read.

20.2.6 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 20.2.1 Allocation of DTC Control Data Area and DTC Vector Table Area.

Figure 20-6. 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

20.2.7 DTC Control Register j (DTCCRj) (j = 0 to 23)

The DTCCRj register is used to control the DTC operating mode.

Figure 20-7. Format of DTC Control Register j (DTCCRj)

Address: Refer to **20.2.2 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.

20.2.8 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 20-8. Format of DTC Block Size Register j (DTBLSj)

Address: Refer to **20.2.2 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.

20.2.9 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 20-9. Format of DTC Transfer Count Register j (DTCCTj)

Address: Refer to **20.2.2 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.

20.2.10 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 20-10. Format of DTC Transfer Count Reload Register j (DTRLDj)

Address: Refer to **20.2.2 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.

20.2.11 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 20-11. Format of DTC Source Address Register j (DTSARj)

Address: Refer to **20.2.2 DTC Control Data Allocation**. After reset: Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTSARj	DTSARj15	DTSARj14	DTSARj13	DTSARj12	DTSARj11	DTSARj10	DTSARj9	DTSARj8	DTSARj7	DTSARj6	DTSARj5	DTSARj4	DTSARj3	DTSARj2	DTSARj1	DTSARj0

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.

20.2.12 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 20-12. Format of DTC Destination Address Register j (DTDARj)

Address: Refer to **20.2.2 DTC Control Data Allocation**. After reset: Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTDARj	DTDARj15	DTDARj14	DTDARj13	DTDARj12	DTDARj11	DTDARj10	DTDARj9	DTDARj8	DTDARj7	DTDARj6	DTDARj5	DTDARj4	DTDARj3	DTDARj2	DTDARj1	DTDARj0

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.

20.2.13 High-speed DTC Channel Select Register 0 (SELHS0)

The SELHS0 register is an 8-bit register that is used to select tye high-speed DTC channel.

Figure 20-13. Format of High-speed DTC Channel Select Register 0 (SELHS0)

Address: F02E1H After reset: 3FH

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.

The correspondence between the activation sources and SELHS05 to SELHS00 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	0	Activation source number 2 is selected as high-speed channel 0.
:						:
:						:
1	1	0	0	0	1	Activation source number 49 is selected as high-speed channel 0.
1	1	0	0	1	0	Activation source number 50 is selected as high-speed channel 0.
1	1	0	0	1	1	Activation source number 51 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 register is 0 (DTC activation disabled).

2. Do not access the SELHS0 register using a DTC transfer.

Remark Products of RL78/F22: i = 0 to 4
Products of RL78/F25: i = 0 to 6

20.2.14 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 20-14. Format of High-speed DTC Channel Select Register 1 (SELHS1)

Address: F02E2H After reset: 3FH

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.

The correspondence between activation sources and SELHS15 to SELHS10 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	0	Activation source number 2 is selected as high-speed channel 1.
:						:
:						:
1	1	0	0	0	1	Activation source number 49 is selected as high-speed channel 1.
1	1	0	0	1	0	Activation source number 50 is selected as high-speed channel 1.
1	1	0	0	1	1	Activation source number 51 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 register is 0 (DTC activation disabled).

2. Do not access the SELHS1 register using a DTC transfer.

Remark Products of RL78/F22: i = 0 to 4
Products of RL78/F25: i = 0 to 6

20.2.15 High-speed DTC Control Register m (HDTCCRm) (m = 0, 1)

The HDTCCRm register is used to control the high-speed DTC transfer operating mode.

Figure 20-15. 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). 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.

20.2.16 High-speed DTC Transfer Count Register m (HDTCCTm) (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 20-16. 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.

20.2.17 DTC Transfer Count Reload Register m (HDTRLDm) (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 20-17. 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.

20.2.18 High-speed DTC Source Address Register m (HDTSARm) (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 20-18. 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
HDTSARm	0	0	0	0	HDTSARm11	HDTSARm10	HDTSARm9	HDTSARm8	HDTSARm7	HDTSARm6	HDTSARm5	HDTSARm4	HDTSARm3	HDTSARm2	HDTSARm1	HDTSARm0

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.

20.2.19 High-speed DTC Destination Address Register m (HDTDARm) (m = 0, 1)

This register is used to specify the transfer destination address for data transfer.

Figure 20-19. 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
HDTDARm	HDTDARm15	HDTDARm14	HDTDARm13	HDTDARm12	HDTDARm11	HDTDARm10	HDTDARm9	HDTDARm8	HDTDARm7	HDTDARm6	HDTDARm5	HDTDARm4	HDTDARm3	HDTDARm2	HDTDARm1	HDTDARm0

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.

20.3 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 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.

Remark j = 0 to 23

20.3.1 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 [Note] register.

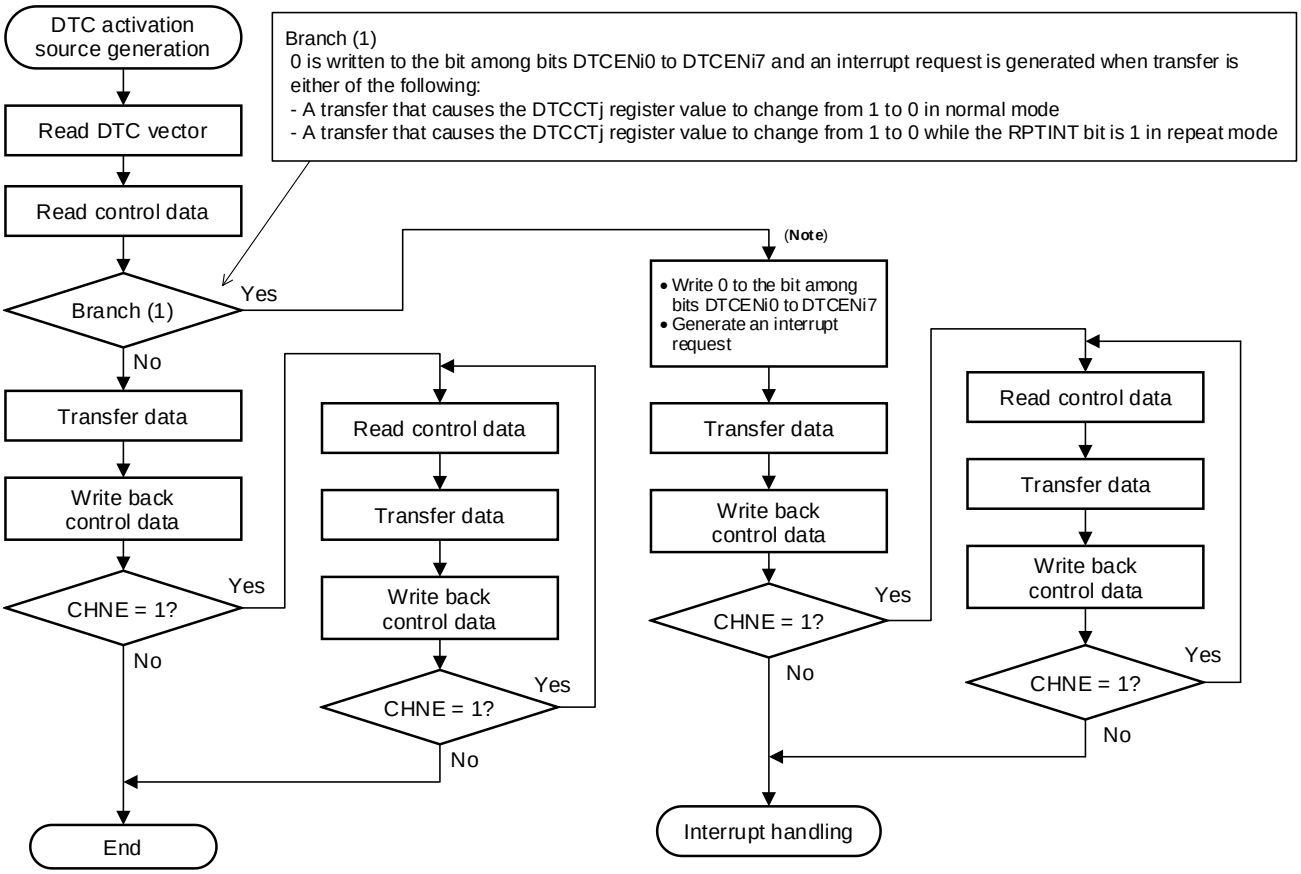
The DTC sets the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi register to 0 (activation disabled) during operation when the setting of data transfer (the first transfer in chain transfers) is either of the following:

- 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

Remark Products of RL78/F22: i = 0 to 4
Products of RL78/F25: i = 0 to 6

Figure 20-20 shows the DTC internal operation flowchart.

Figure 20-20. DTC Internal Operation Flowchart



Note 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.

Remarks 1. DTCENi0 to DTCENi7: Bits in DTCENi register
2. RPTINT, CHNE: Bits in DTCCRj register
3. RL78/F25: i = 0 to 6, j = 0 to 23
RL78/F22: i = 0 to 4, j = 0 to 23

20.3.2 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 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 in the DTCENi register to 0 (activation disabled).

Table 20-7 shows register functions in normal mode. **Figure 20-21** shows data transfers in normal mode.

Remark Products of RL78/F22: i = 0 to 4, j = 0 to 23
Products of RL78/F25: i = 0 to 6, j = 0 to 23

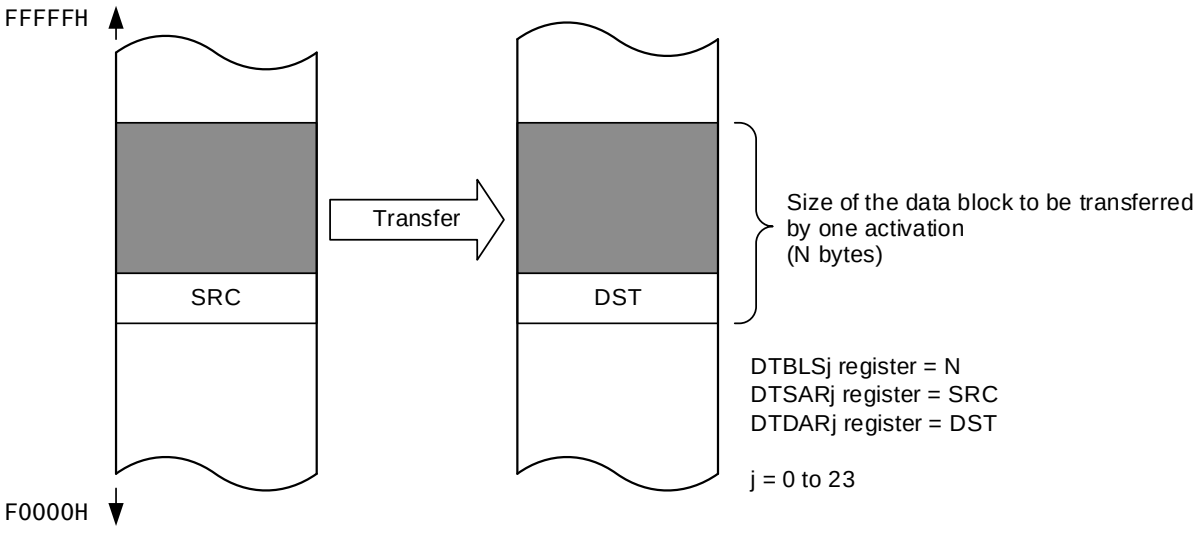
Table 20-7. Register Functions in Normal Mode

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 20-21. 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

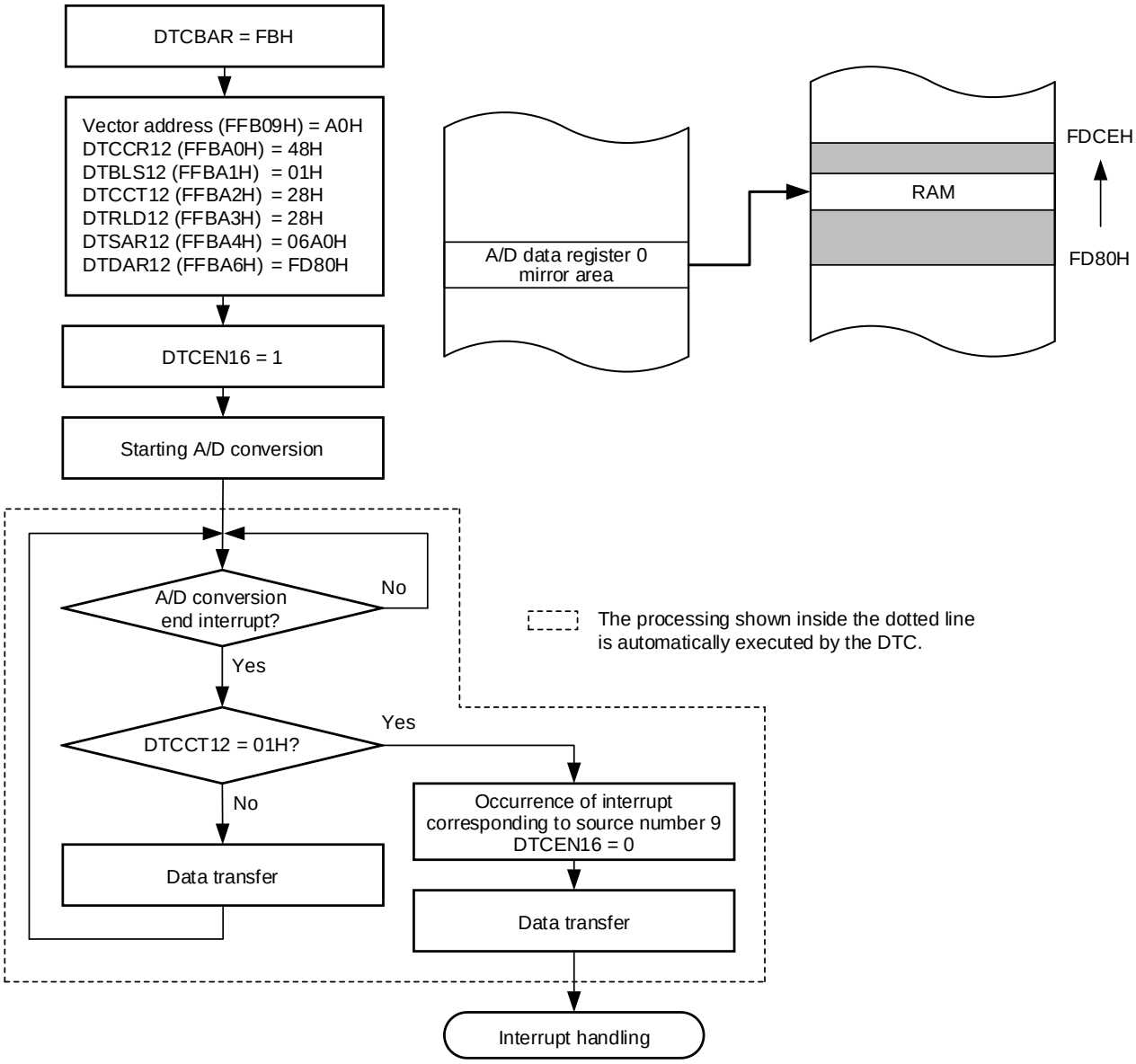
X: 0 or 1

- **Example 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 data register 0 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 data register 0 mirror area (F06A1H, F06A0H) to 80 bytes of FFD80H to FFDCFH of RAM 40 times.

Figure 20-22. Example Using Normal Mode: Consecutively Capturing A/D Conversion Results



The data of DTRLD12 does not affect DTC transfer operation because of normal mode.

20.3.3 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 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 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 20-8 lists register functions in repeat mode. **Figure 20-23** shows data transfers in repeat mode.

Remark Products of RL78/F22: i = 0 to 4, j = 0 to 23
Products of RL78/F25: i = 0 to 6, j = 0 to 23

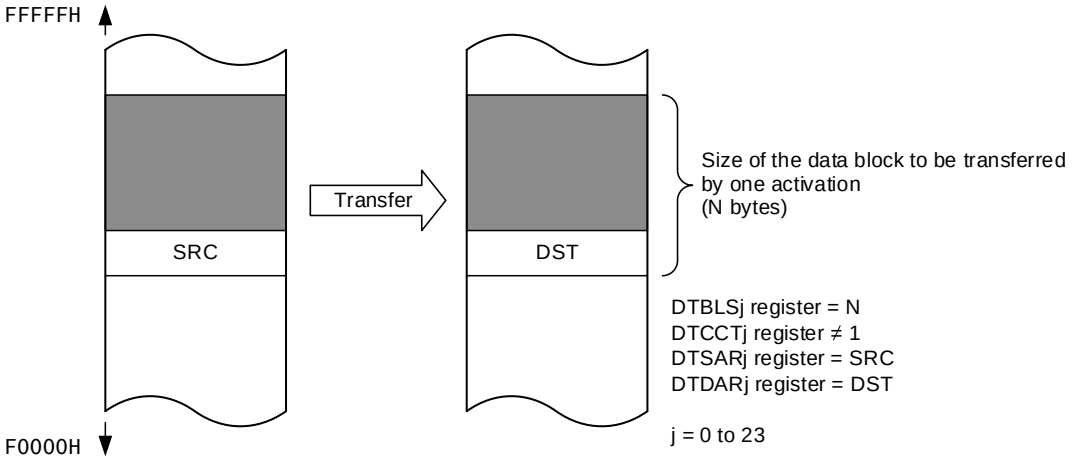
Table 20-8. Register Functions in Repeat Mode

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 20-23. Data Transfers in Repeat Mode

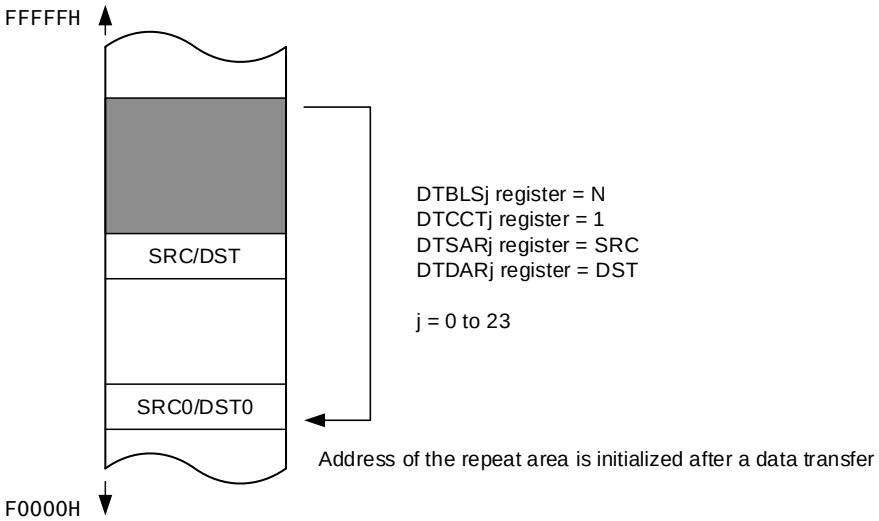
DTCCTj register ≠ 1



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.

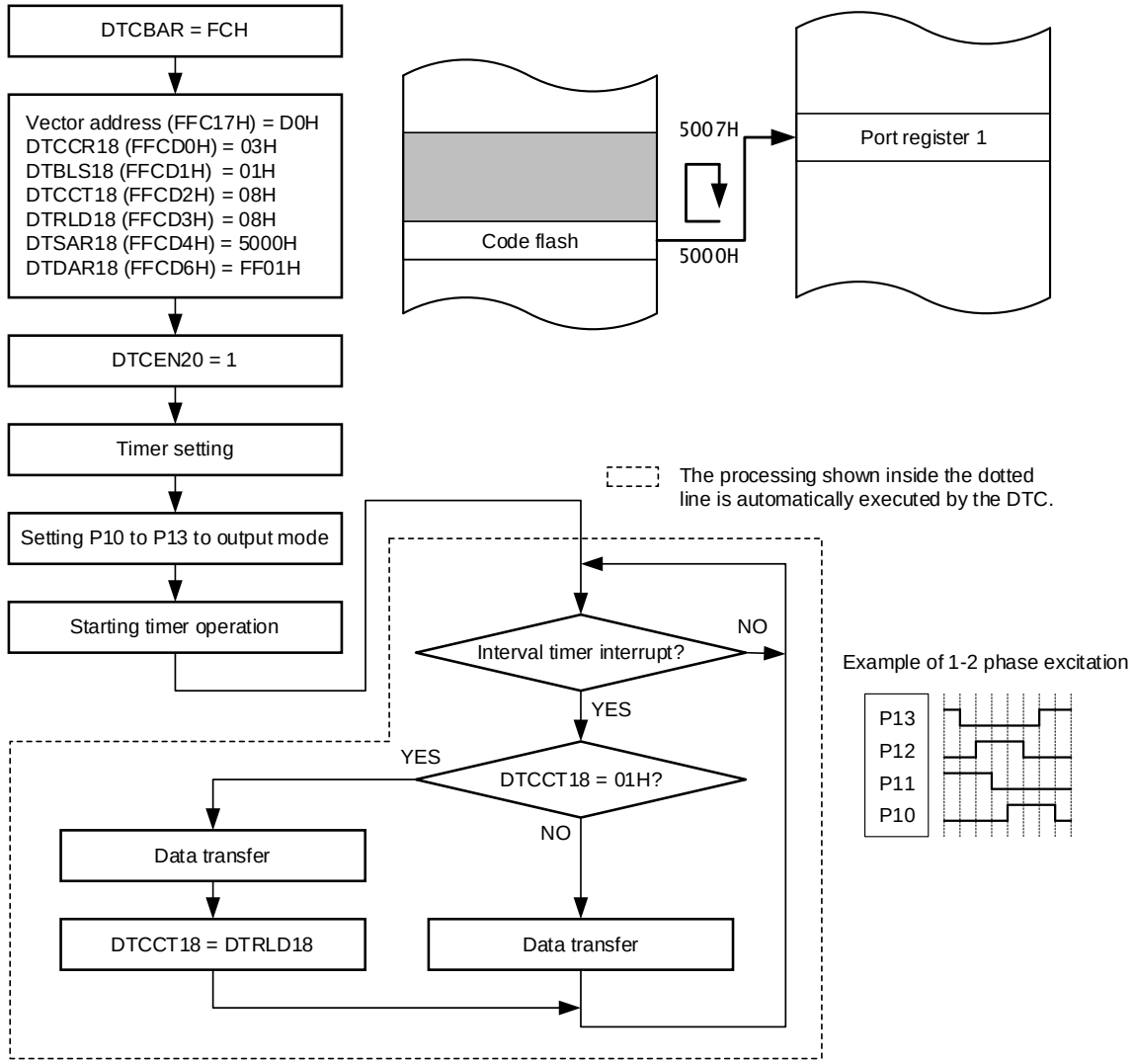
2. When repeat mode is used, the data size of the repeat area must be set to 255 bytes or less.

(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 05000H to 05007H of the code flash memory from the mirror space (F5000H to F5007H) to port register 1 (FFF01H).
- A repeat mode interrupt is disabled.

Figure 20-24. 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.

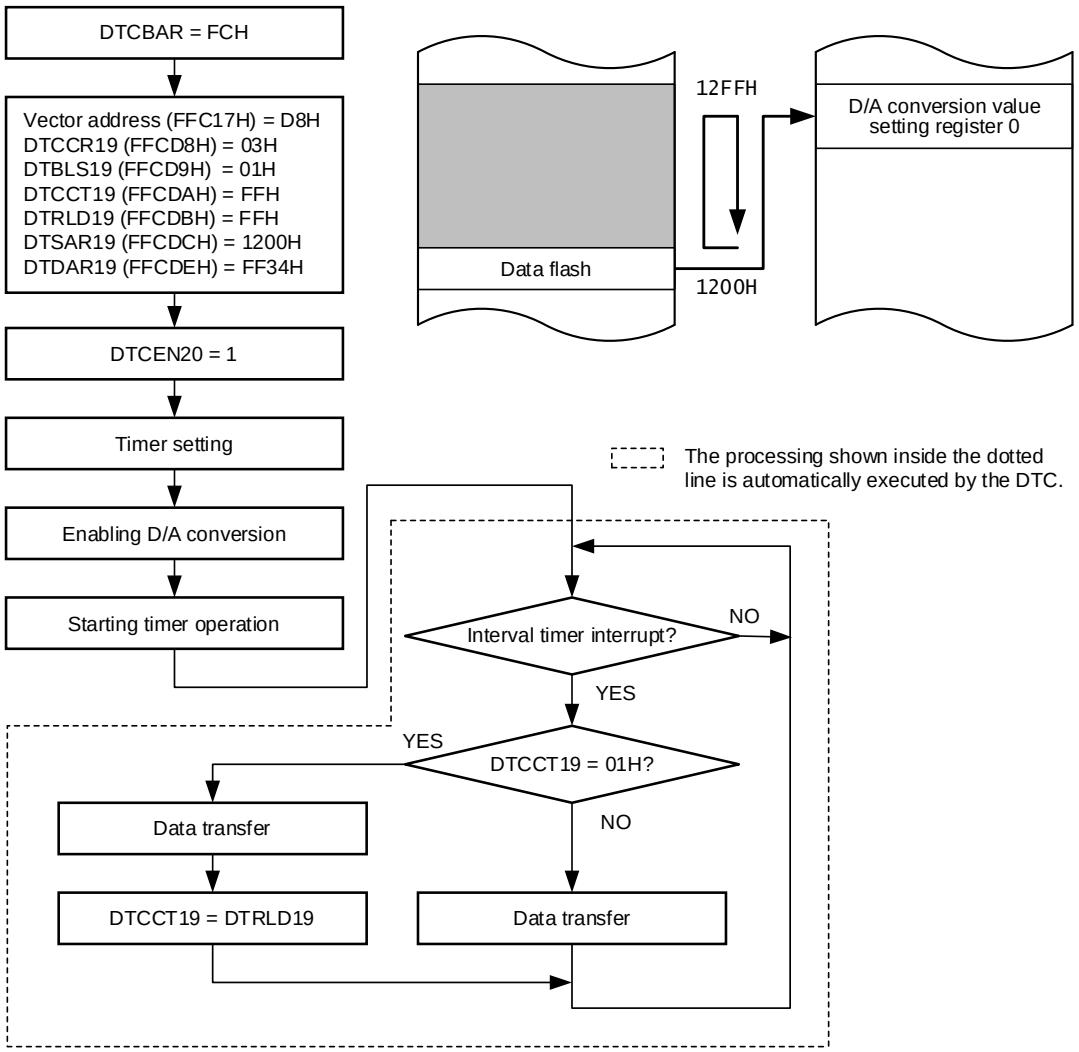
(2) Example 2 of using repeat mode: Outputting a sine wave using the 8-bit D/A converter

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 0.

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 0 (FFF34H).
- A repeat mode interrupt is disabled.

Figure 20-25. 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 RL78/F25 products.

20.3.4 Chain Transfers

When the CHNE bit in the DTCCRj (j = 0 to 22 Note) 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.

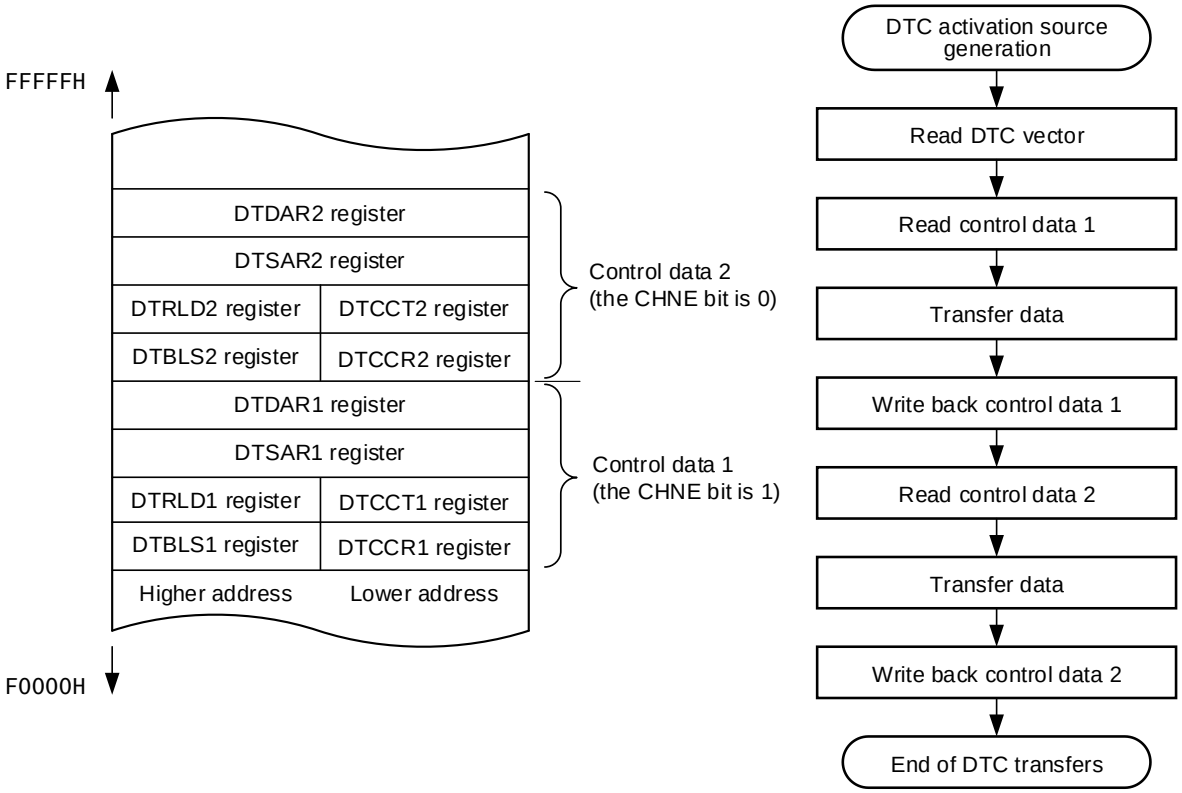
In chain transfers with the use of multiple control data sets, if the normal mode is selected by the second or subsequent control data, the number of transfers set in the first of the control data is valid and settings for the number of transfers in the second or subsequent control data are invalid.

On the other hand, if the repeat mode is selected by the second or subsequent control data, the second or subsequent control data are written back, so the number of transfers needs to be set for each of the control data.

Note Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).

Figure 20-26 shows data transfers in chain transfers.

Figure 20-26. Data Transfers during Chain Transfers



20.3.5 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 (m = 0, 1). 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 20-9 shows the register functions in high-speed transfer operation.

Table 20-9. Register Functions in High-speed Transfer Mode (m = 0, 1)

Register Name	Symbol	Function
High-speed DTC channel select register m	SELHSm	Channel selection
High-speed DTC control register m	HDTCCRm	Operating mode control
High-speed DTC transfer count register m	HDTCCTm	Number of data transfers
High-speed DTC transfer count reload register m	HDTRLDm	Initial value setting
High-speed DTC source address register m	HDTSARm	Data transfer source address
High-speed DTC destination address register m	HDTDARm	Data transfer destination address

20.4 Notes on DTC

20.4.1 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 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 register is 0 (DTC activation disabled).
- In repeat mode, the lower 8 bits of the initial value for the repeat area address must be 00H, and the number of transfers and reload transfers varies depending on the transfer data size.

  8-bit transfer: 01H to FFH (1 to 255 times)

  16-bit transfer: 01H to 7FH (1 to 127 times)

Remark Products of RL78/F22: i = 0 to 4, j = 0 to 23
Products of RL78/F25: i = 0 to 6, j = 0 to 23

20.4.2 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 (FFEE0H to FFEFFH) space as the DTC control data area or DTC vector table area.
- The 11-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.

20.4.3 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
- 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.

20.4.4 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 Registers located in the area from F0100H to F01FFH and from F0300H to F06FFH.

20.4.5 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
```

20.4.6 Number of DTC Execution Clock Cycles

Table 20-10 lists the operations following DTC activation and required number of clock cycles for each operation.

Table 20-10. 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 20-11 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 20-12 Number of Clock Cycles Required for Data Read/Write Operation**.

Table 20-11. 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 20-12. 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.

20.4.7 Number of High-speed DTC Execution Clock Cycles

Table 20-13 lists the operations following high-speed DTC activation and required number of clock cycles for each operation.

Table 20-13. 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 20-14 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 20-15 Number of Clock Cycles Required for Data Read/Write Operation**.

Table 20-14. 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 20-15. 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.

20.4.8 DTC Response Time

Table 20-16 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 20-16. 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 **20.4.3 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 second SFR **Note** that a wait occurs
  Maximum response time: Maximum response time for each condition + 1 clock

Note Registers located in the area from F0100H to F01FFH and from F0300H to F06FFH.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU/peripheral hardware clock)

20.4.9 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.
- When the timer RJ0 is selected as a DTC activation source and DTC transfer is to proceed again following the completion of a previous DTC transfer activated by the same source, set the corresponding DTCEN45 bit in the DTCEN4 register to 1 (enabling activation) after one cycle of the operating clock for the timer RJ0.
- 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 **20.2.3 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

20.4.10 Operation in Standby Mode Status

Table 20-17. Standby Mode Status and DTC Operation

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 1
SNOOZE mode	Operable Note 2

Notes **1**. 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.

2. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected as f_{CLK}.

20.4.11 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.

20.4.12 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 21 EVENT LINK CONTROLLER (ELC) (RL78/F25 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.

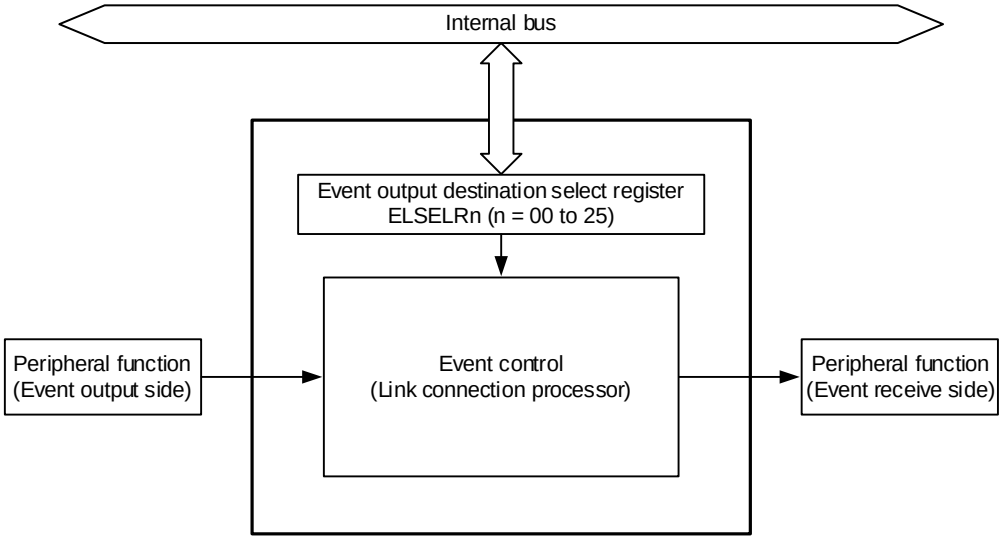
21.1 Overview

The ELC has the following functions.

- Capable of directly linking event signals from 26 types of peripheral functions to specified peripheral functions
- Event signals can be used as activation sources for operating any one of 10 types of peripheral functions

Figure 21-1 shows the Event Link Controller Block Diagram.

Figure 21-1. Event Link Controller Block Diagram



21.2 Registers

Table 21-1 lists the ELC register configuration.

Table 21-1. ELC Register Configuration

Address	Register Name	Symbol	After Reset	Access Size
F0780H	Event output destination select register 00	ELSELR00	00H	1, 8
F0781H	Event output destination select register 01	ELSELR01	00H	1, 8
F0782H	Event output destination select register 02	ELSELR02	00H	1, 8
F0783H	Event output destination select register 03	ELSELR03	00H	1, 8
F0784H	Event output destination select register 04	ELSELR04	00H	1, 8
F0785H	Event output destination select register 05	ELSELR05	00H	1, 8
F0786H	Event output destination select register 06	ELSELR06	00H	1, 8
F0787H	Event output destination select register 07	ELSELR07	00H	1, 8
F0788H	Event output destination select register 08	ELSELR08	00H	1, 8
F0789H	Event output destination select register 09	ELSELR09	00H	1, 8
F078AH	Event output destination select register 10	ELSELR10	00H	1, 8
F078BH	Event output destination select register 11	ELSELR11	00H	1, 8
F078CH	Event output destination select register 12	ELSELR12	00H	1, 8
F078DH	Event output destination select register 13	ELSELR13	00H	1, 8
F078EH	Event output destination select register 14	ELSELR14	00H	1, 8
F078FH	Event output destination select register 15	ELSELR15	00H	1, 8
F0790H	Event output destination select register 16	ELSELR16	00H	1, 8
F0791H	Event output destination select register 17	ELSELR17	00H	1, 8
F0792H	Event output destination select register 18	ELSELR18	00H	1, 8
F0793H	Event output destination select register 19	ELSELR19	00H	1, 8
F0794H	Event output destination select register 20	ELSELR20	00H	1, 8
F0795H	Event output destination select register 21	ELSELR21	00H	1, 8
F0796H	Event output destination select register 22	ELSELR22	00H	1, 8
F0797H	Event output destination select register 23	ELSELR23	00H	1, 8
F0798H	Event output destination select register 24	ELSELR24	00H	1, 8
F0799H	Event output destination select register 25	ELSELR25	00H	1, 8

21.2.1 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 and the function of the event output destination (event receive side) is stopped.

Table 21-2 lists the correspondence between ELSELRn (n = 00 to 25) registers and peripheral functions, and **Table 21-3** lists the correspondence between values set to ELSELRn (n = 00 to 25) registers and operation of link destination peripheral functions at reception.

Figure 21-2. 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	0	0	0	0	ELSELRn3	ELSELRn2	ELSELRn1	ELSELRn0

Bits 7 to 4	Reserved
–	The read value is 0.

ELSELRn3	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
0	0	1	0	Select operation of peripheral function to link Note
0	0	1	1	Select operation of peripheral function to link Note
0	1	0	0	Select operation of peripheral function to link Note
0	1	0	1	Select operation of peripheral function to link Note
0	1	1	0	Select operation of peripheral function to link Note
0	1	1	1	Select operation of peripheral function to link Note
1	0	0	0	Select operation of peripheral function to link Note
1	0	0	1	Select operation of peripheral function to link Note
1	0	1	0	Select operation of peripheral function to link Note
Other than above				Setting prohibited

Note See **Table 21-3 Correspondence Between Values Set to ELSELRn (n = 00 to 25) Registers and Operation of Link Destination Peripheral Functions at Reception.**

Table 21-2. 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	TRD0 input capture A/Compare match A	INTTRD0
ELSELR09	TRD0 input capture B/Compare match B	INTTRD0
ELSELR10	TRD1 input capture A/Compare match A	INTTRD1
ELSELR11	TRD1 input capture B/Compare match B	INTTRD1
ELSELR12	TRD1 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	TAU0 channel 5 count end/Capture end	INTTM05
ELSELR21	TAU0 channel 6 count end/Capture end	INTTM06
ELSELR22	TAU0 channel 7 count end/Capture end	INTTM07
ELSELR23	TAU1 channel 0 count end/Capture end	INTTM10
ELSELR24	TAU1 channel 1 count end/Capture end	INTTM11
ELSELR25	TAU1 channel 2 count end/Capture end	INTTM12

Table 21-3. 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 , 2	Delay counter, input pulse interval measurement, external event counter
0011B	Timer input of timer array unit 0 channel 1 Notes 1 , 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 , 2	Delay counter, input pulse interval measurement, external event counter
1001B	Timer input of timer array unit 0 channel 3 Notes 1 , 2	
1010B	PWMOPA	Pulse output forced cutoff

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 mode while the real-time output event mode for D/A conversion is enabled, disable linking of ELC events before entering STOP mode.

Remark m = 0 to 3

21.2.2 Timer Input Select Register 0 (TIS0)

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)**.

21.2.3 A/D Conversion Start Trigger Select Register (ADSTRGR)

This register has the function of specifying the A/D conversion trigger. The 12-bit A/D converter can use the event input from ELC as a start trigger. For details, see **12.2.10 A/D Conversion Start Trigger Select Register (ADSTRGR)**.

21.2.4 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 **13.3.2 D/A Converter Mode Register (DAM)**.

21.3 Operation

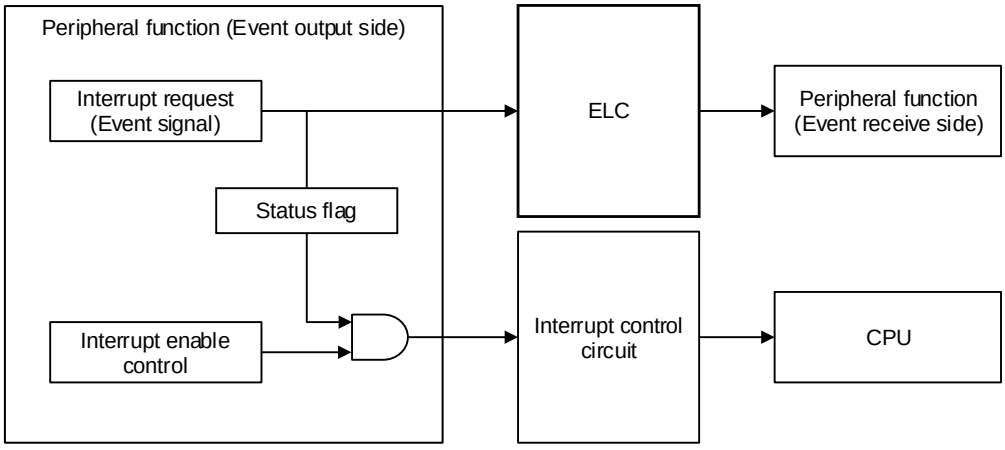
The path for using an event signal generated by a peripheral function as an interrupt request to the interrupt control circuit is independent from the path for using it as an ELC event. Therefore, each event signal can be used as an event signal for operation of an event-receiving peripheral function, 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 21-3 shows the relationship between interrupt handling and ELC. The figure shows 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 21-3 Correspondence Between Values Set to ELSELRn (n = 00 to 25) Registers and Operation of Link Destination Peripheral Functions at Reception**).

Figure 21-3. Relationship Between Interrupt Handling and ELC



CHAPTER 22 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/F25				RL78/F22		
		100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
Maskable interrupts	External	17	17	16	15	12	8	7
	Internal	63	63	63	63	39	39	39

22.1 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, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H).

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 22-1**.

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.

22.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Table 22-1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt request are two bytes each, so interrupts jump to a 64 KB address of 00000H to 0FFFFH.

Table 22-1. Interrupt Source List (1/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	RL78/F25				RL78/F22		
		Name	Trigger				100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
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	Pin input edge detection 4		0010H		√	√	√	√	√	√	√
		INTSPM	Stack pointer overflow/underflow	Internal		(A)	√	√	√	√	√	√	√
	7	INTP5	Pin input edge detection 5	External	0012H	(B)	√	√	√	√	√	√	√
		INTCMP0	Comparator detection 0	Internal		(A)	√	√	√	√	–	–	–
	8	INTP13	Pin input edge detection 13	External	0014H	(B)	√	√	–	–	–	–	–
		INTCLM	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	Internal	0016H	(A)	√	√	√	√	√	√	√
	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 underflow, capture		001EH		√	√	√	√	√	√	√
	14	INTRAM	RAM 1-bit correction/2-bit error detection		0020H		√	√	√	√	√	√	√
	15	INTCTSUWR	Configuration register write request for each CTSU channel		0022H		√	√	√	√	√	√	√
	16	INTCTSURD	CTSU measurement data transfer request		0024H		√	√	√	√	√	√	√
	17	INTCTSUFN	CTSU measurement completed		0026H		√	√	√	√	√	√	√

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

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 60 indicates the lowest priority.

2. Basic configuration types (A) to (F) correspond to (A) to (F) in **Figure 22-1**.

3. When bit 7 (WDTINT) of the user option byte (000C0H/040C0H) is set to 1.

4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

Remark For details on how to determine multiple interrupt sources that share the same vector table address, refer to **22.4.1 Determination of Interrupt Source**.

Table 22-1. Interrupt Source List (2/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	RL78/F25				RL78/F22		
		Name	Trigger				100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
Maskable	18	INTIICA0	IICA0 transfer end	Internal	0028H	(A)	√	√	√	√	√	√	√
	19	INTP8	Pin input edge detection 8	External	002AH	(B)	√	√	√	√	√	–	–
		INTRTC	RTC pretimed signal or alarm match detection	Internal		(A)	√	√	√	√	√	√	√
	20	INTTM00	End of TAU0 channel 0 count/capture		002CH		√	√	√	√	√	√	√
	21	INTTM01	End of TAU0 channel 1 count/capture		002EH		√	√	√	√	√	√	√
		INTLIN2TRM	LIN2 transmission				√	√	√	√	–	–	–
	22	INTTM02	End of TAU0 channel 2 count/capture		0030H		√	√	√	√	√	√	√
		INTLIN2RVC	LIN2 reception end				√	√	√	√	–	–	–
	23	INTTM03	End of TAU0 channel 3 count/capture		0032H		√	√	√	√	√	√	√
		INTLIN2STA /INTLIN2	LIN2 reception status/ LIN2 interrupt				√	√	√	√	–	–	–
	24	INTAD	End of A/D conversion		0034H		√	√	√	√	√	√	√
	25	INTP6 Note 3, 4	Pin input edge detection 6	External	0036H	(B)	√	√	√	√	√	–	–
		INTLIN2WUP Note 3, 4	LIN2 reception pin input			(E)	√	√	√	√	–	–	–
		INTTM11H	Upper 8-bit interval timer interrupt of TAU1 channel 1 (when 8-bit timer function is selected)	Internal		(A)	√	√	√	√	√	√	√
	26	INTP7 Note 4	Pin input edge detection 7	External	0038H	(B)	√	√	√	√	√	–	–
		INTTM13H Note 4	Upper 8-bit interval timer interrupt of TAU1 channel 3 (when 8-bit timer function is selected)	Internal		(A)	√	√	√	√	√	√	√
	27	INTP9 Note 4	Pin input edge detection 9	External	003AH	(B)	√	√	√	√	√	–	–
		INTTM01H Note 4	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)	√	√	√	–	–	–	–
		INTTM03H Note 4	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 5

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

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 60 indicates the lowest priority.

2. Basic configuration types (A) to (F) correspond to (A) to (F) in **Figure 22-1**.

3. Select INTP6 and INTLIN2WUP by the ISC4 bit in the ISC register.

4. If these functions that share the same vector table address are used at the same time, the cause of the interrupt cannot be determined.

5. Only INTST1 and INTCSI10 are provided.

Remark For details on how to determine multiple interrupt sources that share the same vector table address, refer to **22.4.1 Determination of Interrupt Source**.

Table 22-1. Interrupt Source List (3/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	RL78/F25				RL78/F22		
		Name	Trigger				100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
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
	31	INTTM04	End of TAU0 channel 4 count/capture		0042H		√	√	√	√	√	√	√
	32	INTTM05	End of TAU0 channel 5 count/capture		0044H		√	√	√	√	√	√	√
		INTLIN0TRM	LIN0 transmission				√	√	√	√	√	√	√
	33	INTTM06	End of TAU0 channel 6 count/capture		0046H		√	√	√	√	√	√	√
		INTLIN0RVC	LIN0 reception end				√	√	√	√	√	√	√
	34	INTTM07	End of TAU0 channel 7 count/capture		0048H		√	√	√	√	√	√	√
		INTLIN0STA/ INTLIN0	LIN0 reception status/ LIN0 interrupt				√	√	√	√	√	√	√
	35	INTP11 Note 4	Pin input edge detection 11	External	004AH	(B)	√	√	√	–	–	–	–
		INTLIN0WUP Note 4	LIN0 reception pin input			(E)	√	√	√	√	√	√	√
	36	INTTM10	End of TAU1 channel 0 count/capture	Internal	004CH	(A)	√	√	√	√	√	√	√
	37	INTTM11	End of TAU1 channel 1 count/capture		004EH		√	√	√	√	√	√	√
		INTLIN1TRM	LIN1 transmission				√	√	√	√	–	–	–
	38	INTTM12	End of TAU1 channel 2 count/capture		0050H		√	√	√	√	√	√	√
		INTLIN1RVC	LIN1 reception end				√	√	√	√	–	–	–
	39	INTTM13	End of TAU1 channel 3 count/capture		0052H		√	√	√	√	√	√	√
		INTLIN1STA/ INTLIN1	LIN1 reception status/ LIN1 interrupt				√	√	√	√	–	–	–
	40	Reserved	Reserved		0054H		√	√	√	√	√	√	√
	41	Reserved	Reserved		0056H		√	√	√	√	√	√	√
	42	INTADGB	End of A/D conversion (Group B)		0058H		√	√	√	√	√	√	√
	43	INTROM	Code Flash ROM ECC Error Detection (DED/SED)		005AH		√	√	√	√	√	√	√
	44	INTKR Note 5	Key interrupt detection	External	005CH	(C)	√	√	√	√	√	√	√
		INTRCAN0GRVC Note 5	CAN0 global receive message buffer reception	Internal		(A)	√	√	√	√	–	–	–
	45	INTRCAN0ERR	CAN0 channel error detection		005EH		√	√	√	√	–	–	–
	46	INTRCAN0WUP	CAN0 wakeup reception pin input	External	0060H	(D)	√	√	√	√	–	–	–
	47	Reserved	Reserved	Internal	0062H	(A)	–	–	–	–	–	–	–

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

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 60 indicates the lowest priority.

2. Basic configuration types (A) to (F) correspond to (A) to (F) in **Figure 22-1**.

3. Only INTSR1 is provided.

4. Select INTP11 and INTLIN0WUP by the ISC2 bit in the ISC register.

5. If these functions are used at the same time, the cause of the interrupt cannot be determined.

Remark For details on how to determine multiple interrupt sources that share the same vector table address, refer to **22.4.1 Determination of Interrupt Source**.

Table 22-1. Interrupt Source List (4/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	RL78/F25				RL78/F22		
		Name	Trigger				100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
Maskable	48	INTRCAN0CFR	CAN0 transmit/receive FIFO message reception	Internal	0064H	(A)	√	√	√	√	–	–	–
	49	INTRCAN0TRM	CAN0 channel transmit		0066H		√	√	√	√	–	–	–
	50	INTRCAN0GRFR	CAN0 global receive FIFO message reception		0068H		√	√	√	√	–	–	–
	51	INTRCAN0GERR	CAN0 global error detection		006AH		√	√	√	√	–	–	–
	52	INTP12 Note 3	Pin input edge detection 12	External	006CH	(B)	√	√	√	–	–	–	–
		INTLIN1WUP Note 3	LIN1 reception pin input			(E)	√	√	√	√	–	–	–
	53	INTRCAN1GRFR	CAN1 global receive FIFO message reception	Internal	006EH	(A)	√	√	√	√	–	–	–
	54	INTRCAN1GERR	CAN1 global error detection		0070H		√	√	√	√	–	–	–
	55	INTRCAN1WUP	CAN1 wakeup reception pin input	External	0072H	(D)	√	√	√	√	–	–	–
	56	INTTM14	End of TAU1 channel 4 count/capture	Internal	0074H	(A)	√	√	√	√	–	–	–
		INTRCAN1GRVC	CAN1 global receive message buffer reception				√	√	√	√	–	–	–
	57	INTTM15	End of TAU1 channel 5 count/capture		0076H		√	√	√	√	–	–	–
		INTRCAN1ERR	CAN1 channel error detection				√	√	√	√	–	–	–
	58	INTTM16	End of TAU1 channel 6 count/capture		0078H		√	√	√	√	–	–	–
		INTRCAN1CFR	CAN1 transmit/receive FIFO message reception				√	√	√	√	–	–	–
	59	INTTM17	End of TAU1 channel 7 count/capture		007AH		√	√	√	√	–	–	–
		INTRCAN1TRM	CAN1 channel transmit				√	√	√	√	–	–	–
	60	INTCRAM Note 4	CAN-RAM ECC 1-bit correction/2-bit error detection		007CH		√	√	√	√	–	–	–
Software	–	BRK	Execution of BRK instruction	–	007EH	(F)	√	√	√	√	√	√	√
Reset	–	RESET	RESET pin input	–	0000H	–	√	√	√	√	√	√	√
		POR	Power-on-reset				√	√	√	√	√	√	√
		LVD	Voltage detection Note 5				√	√	√	√	√	√	√
		WDT	Overflow of watchdog timer				√	√	√	√	√	√	√
		TRAP	Execution of illegal instruction Note 6				√	√	√	√	√	√	√
		IAW	Illegal-memory access				√	√	√	√	√	√	√
		CLM	Main clock oscillation stop				√	√	√	√	√	√	√

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

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 60 indicates the lowest priority.

2. Basic configuration types (A) to (F) correspond to (A) to (F) in **Figure 22-1**.

3. Select INTP12 and INTLIN1WUP by the ISC3 bit in the ISC register.

4. The ECC detection interrupts from CANRAM0 and CANRAM1 are input as a single interrupt through the logical OR circuit. Determine which CANRAM is the source of the interrupt by reading the CFDECCTL register.

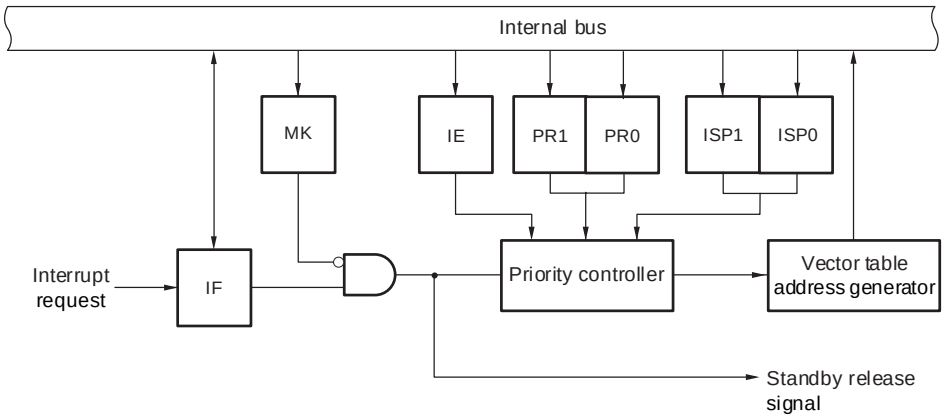
5. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

6. 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.

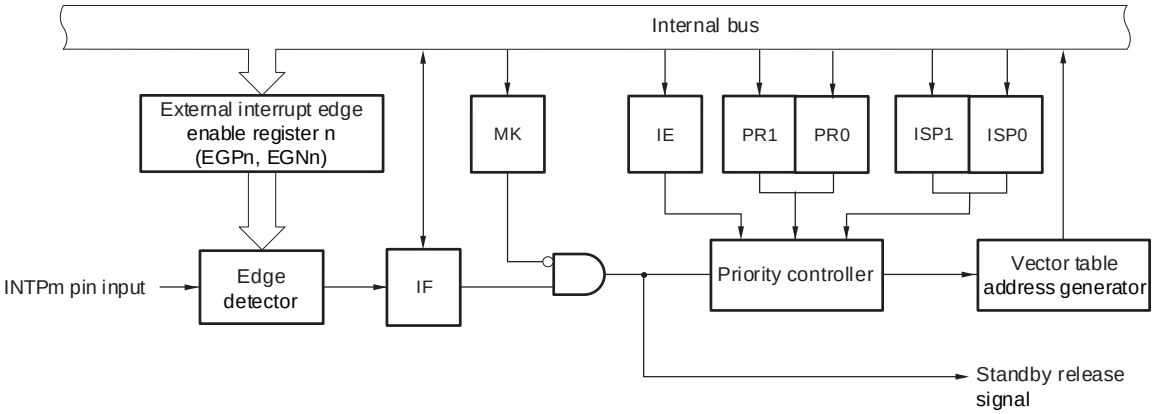
Remark For details on how to determine multiple interrupt sources that share the same vector table address, refer to **22.4.1 Determination of Interrupt Source**.

Figure 22-1. Basic Configuration of Interrupt Function (1/3)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPm)

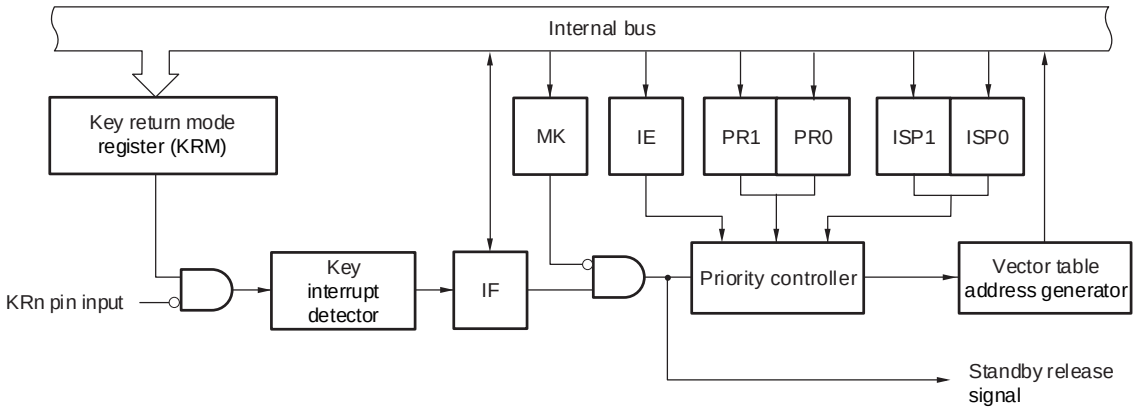


IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

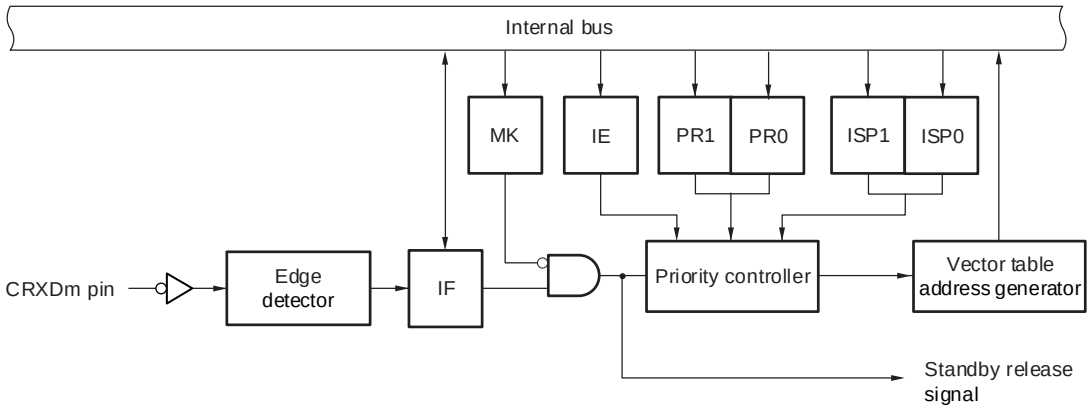
Remark n = 0, 1
24-pin: m = 0, 2 to 5
32-pin: m = 0 to 5
48-pin: m = 0 to 9
64-pin: m = 0 to 12
80-, 100-pin: m = 0 to 13

Figure 22-1. Basic Configuration of Interrupt Function (2/3)

(C) External maskable interrupt (INTKR)



(D) External maskable interrupt (CAN wake-up) (RL78/F25 only)

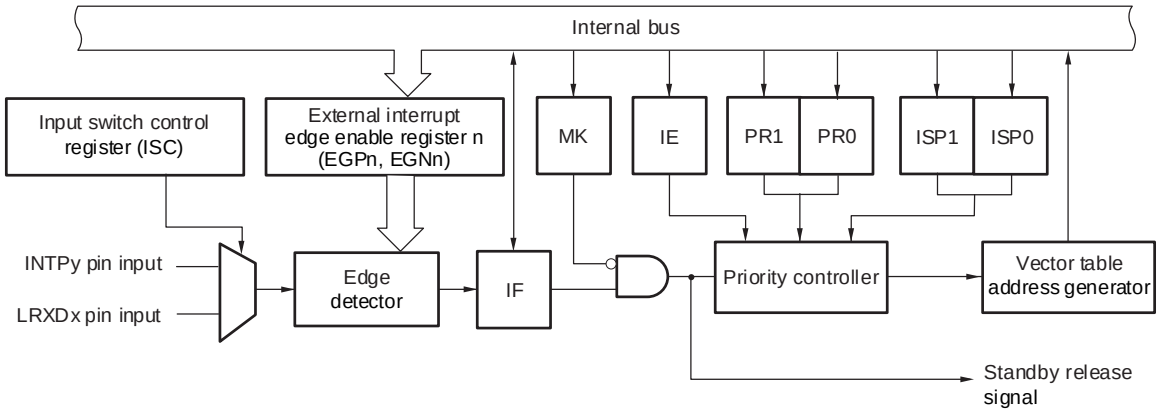


IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

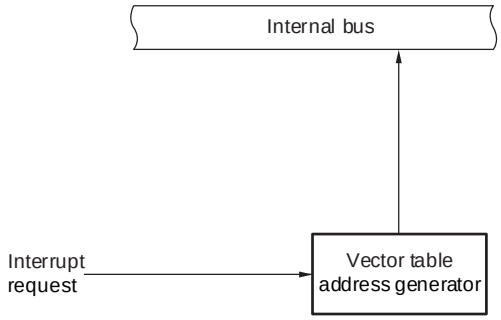
Remark 24-pin: n = 0 to 2
32-pin: n = 0 to 5
48-, 64-, 80-, 100-pin: n = 0 to 7
m = 0, 1

Figure 22-1. 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
RL78/F22 products: x = 0, y = 11
RL78/F25 48-, 64-, 80-pin: x = 0, 1, y = 11, 12
RL78/F25 100-pin: x = 0 to 2, y = 6, 11, 12

22.3 Registers Controlling Interrupt Functions

Table 22-2. Interrupt Function Register Configuration

Address	Register Name	Symbol	After Reset	Access Size
FFF38H	External interrupt rising edge enable register 0	EGP0	00H	1, 8
FFF39H	External interrupt falling edge enable register 0	EGN0	00H	1, 8
FFF3AH	External interrupt rising edge enable register 1	EGP1	00H	1, 8
FFF3BH	External interrupt falling edge enable register 1	EGN1	00H	1, 8
FFFD0H	Interrupt request flag register 2L	IF2L	00H	1, 8
FFFD1H	Interrupt request flag register 2H	IF2H	0000x000B	1, 8
FFFD2H	Interrupt request flag register 3L	IF3L	00H	1, 8
FFFD3H	Interrupt request flag register 3H	IF3H	00H	1, 8
FFFD4H	Interrupt mask flag register 2L	MK2L	FFH	1, 8
FFFD5H	Interrupt mask flag register 2H	MK2H	FFH	1, 8
FFFD6H	Interrupt mask flag register 3L	MK3L	FFH	1, 8
FFFD7H	Interrupt mask flag register 3H	MK3H	FFH	1, 8
FFFD8H	Interrupt priority specification flag 0 register 2L	PR02L	FFH	1, 8
FFFD9H	Interrupt priority specification flag 0 register 2H	PR02H	FFH	1, 8
FFFDAH	Interrupt priority specification flag 0 register 3L	PR03L	FFH	1, 8
FFFDBH	Interrupt priority specification flag 0 register 3H	PR03H	FFH	1, 8
FFFDCH	Interrupt priority specification flag 1 register 2L	PR12L	FFH	1, 8
FFFDDH	Interrupt priority specification flag 1 register 2H	PR12H	FFH	1, 8
FFFDEH	Interrupt priority specification flag 1 register 3L	PR13L	FFH	1, 8
FFFDFH	Interrupt priority specification flag 1 register 3H	PR13H	FFH	1, 8
FFFE0H	Interrupt request flag register 0L	IF0L	00H	1, 8
FFFE1H	Interrupt request flag register 0H	IF0H	00H	1, 8
FFFE2H	Interrupt request flag register 1L	IF1L	00H	1, 8
FFFE3H	Interrupt request flag register 1H	IF1H	00H	1, 8
FFFE4H	Interrupt mask flag register 0L	MK0L	FFH	1, 8
FFFE5H	Interrupt mask flag register 0H	MK0H	FFH	1, 8
FFFE6H	Interrupt mask flag register 1L	MK1L	FFH	1, 8
FFFE7H	Interrupt mask flag register 1H	MK1H	FFH	1, 8
FFFE8H	Interrupt priority specification flag 0 register 0L	PR00L	FFH	1, 8
FFFE9H	Interrupt priority specification flag 0 register 0H	PR00H	FFH	1, 8
FFFEAH	Interrupt priority specification flag 0 register 1L	PR01L	FFH	1, 8
FFFEBH	Interrupt priority specification flag 0 register 1H	PR01H	FFH	1, 8
FFFECH	Interrupt priority specification flag 1 register 0L	PR10L	FFH	1, 8
FFFEDH	Interrupt priority specification flag 1 register 0H	PR10H	FFH	1, 8
FFFEEH	Interrupt priority specification flag 1 register 1L	PR11L	FFH	1, 8
FFFEFH	Interrupt priority specification flag 1 register 1H	PR11H	FFH	1, 8
F0073H	Interrupt switch control register	ISC	00H	1, 8
F0079H	Interrupt source determination flag register 0	INTFLG0	00H	8
F007DH	Interrupt source determination flag register 1	INTFLG1	00H	8
F01F2H	Interrupt source determination flag register 2	INTFLG2	00H	8
F01F3H	Interrupt source determination flag register 3	INTFLG3	00H	8
F007CH	Interrupt mask register	INTMSK	FFH	8

Table 22-3 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 22-3. Flags Corresponding to Interrupt Request Sources (1/7)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	RL78/F25				RL78/F22		
							100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
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		√	√	√	√	–	–	–

Table 22-3. Flags Corresponding to Interrupt Request Sources (2/7)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		RL78/F25				RL78/F22		
		Register		Register		Register	100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
INTP13	PIF13	IF0H	PMK13	MK0H	PPR013, PPR113	PR00H, PR10H	√	√	–	–	–	–	–
INTCLM	CLMIF		CLMMK		CLMPR0, CLMPR1		√	√	√	√	√	√	√
INTST0	STIF0		STMK0		STPR00, STPR10		√	√	√	√	√	√	√
INTCSI00	CSIIF00		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		RAMMK		RAMPR0, RAMPR1		√	√	√	√	√	√	√
INTCTSUWR	CTSUWRIF		CTSUWRMK		CTSUWRPR0, CTSUWRPR1		√	√	√	√	√	√	√

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.

Table 22-3. Flags Corresponding to Interrupt Request Sources (3/7)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		RL78/F25				RL78/F22		
		Register		Register		Register	100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
INTCTSURD	CTSURDIF	IF1L	CTSURDMK	MK1L	CTSURDPR0, CTSURDPR1	PR01L, PR11L	√	√	√	√	√	√	√
INTCTSUFN	CTSUFNIF		CTSUFNMK		CTSUFNPR0, CTSUFNPR1		√	√	√	√	√	√	√
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		√	√	√	√	√	√	√
INTP8	PIF8		PMK8		PPR08, PPR18		√	√	√	√	√	–	–
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1		√	√	√	√	√	√	√
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		√	√	√	√	√	√	√
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		√	√	√	√	√	√	√
INTLIN2TRM	LIN2TRMIF		LIN2TRMMK		LIN2TRMPR0, LIN2TRMPR1		√	√	√	√	–	–	–
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		√	√	√	√	√	√	√
INTLIN2RVC	LIN2RVCIF		LIN2RVCMK		LIN2RVCPR0, LIN2RVCPR1		√	√	√	√	–	–	–
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		√	√	√	√	√	√	√
INTLIN2STA	LIN2STAIF		LIN2STAMK		LIN2STAPR0, LIN2STAPR1		√	√	√	√	–	–	–
INTLIN2	LIN2IF		LIN2MK		LIN2PR0, LIN2PR1		√	√	√	√	–	–	–

Table 22-3. Flags Corresponding to Interrupt Request Sources (4/7)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	RL78/F25				RL78/F22		
							100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H	√	√	√	√	√	√	√
INTP6	PIF6		PMK6		PPR06, PPR16		√	√	√	√	√	–	–
INTLIN2WUP	LIN2WUPIF		LIN2WUPMK		LIN2WUPPR0 LIN2WUPPR1		√	√	√	√	–	–	–
INTTM11H	TMIF11H		TMMK11H		TMPR011H, TMPR111H		√	√	√	√	√	√	√
INTP7	PIF7		PMK7		PPR07, PPR17		√	√	√	√	√	–	–
INTTM13H	TMIF13H		TMMK13H		TMPR013H, TMPR113H		√	√	√	√	√	√	√
INTP9	PIF9		PMK9		PPR09, PPR19		√	√	√	√	√	–	–
INTTM01H	TMIF01H		TMMK01H		TMPR001H, TMPR101H		√	√	√	√	√	√	√
INTP10	PIF10		PMK10		PPR010, PPR110		√	√	√	–	–	–	–
INTTM03H	TMIF03H		TMMK03H		TMPR003H, TMPR103H		√	√	√	√	√	√	√
INTST1	STIF1		STMK1		STPR01, STPR11		√	√	√	√	√	√	√
INTCSI10	CSIIF10		CSIMK10		CSIPR010, CSIPR110		√	√	√	√	√	√	√
INTIIC10	IICIF10		IICMK10		IICPR010, IICPR110		√	√	√	√	√	√	√
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11		√	√	√	√	√	√	√
INTCSI11	CSIIF11		CSIMK11		CSIPR011, CSIPR111		√	√	√	√	√	–	–
INTIIC11	IICIF11		IICMK11		IICPR011, IICPR111		√	√	√	√	√	–	–
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104		√	√	√	√	√	√	√

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 INTP9 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 INTP10 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.**

Table 22-3. Flags Corresponding to Interrupt Request Sources (5/7)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		RL78/F25				RL78/F22		
		Register		Register		Register	100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L	√	√	√	√	√	√	√
INTLIN0TRM	LIN0TRMIF		LIN0TRMMK		LIN0TRMPR0, LIN0TRMPR1		√	√	√	√	√	√	√
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106		√	√	√	√	√	√	√
INTLIN0RVC	LIN0RVCIF		LIN0RVCMK		LIN0RVCPR0, LIN0RVCPR1		√	√	√	√	√	√	√
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		√	√	√	√	√	√	√
INTLIN0STA	LIN0STAIF		LIN0STAMK		LIN0STAPR0, LIN0STAPR1		√	√	√	√	√	√	√
INTLIN0	LIN0IF		LIN0MK		LIN0PR0, LIN0PR1		√	√	√	√	√	√	√
INTP11	PIF11		PMK11		PPR011, PPR111		√	√	√	–	–	–	–
INTLIN0WUP	LIN0WUPIF		LIN0WUPMK		LIN0WUPPR0, LIN0WUPPR1		√	√	√	√	√	√	√
INTTM10	TMIF10		TMMK10		TMPR010, TMPR110		√	√	√	√	√	√	√
INTTM11	TMIF11		TMMK11		TMPR011, TMPR111		√	√	√	√	√	√	√
INTLIN1TRM	LIN1TRMIF		LIN1TRMMK		LIN1TRMPR0, LIN1TRMPR1		√	√	√	√	–	–	–
INTTM12	TMIF12		TMMK12		TMPR012, TMPR112		√	√	√	√	√	√	√
INTLIN1RVC	LIN1RVCIF		LIN1RVCMK		LIN1RVCPR0, LIN1RVCPR1		√	√	√	√	–	–	–
INTTM13	TMIF13		TMMK13		TMPR013, TMPR113		√	√	√	√	√	√	√
INTLIN1STA	LIN1STAIF		LIN1STAMK		LIN1STAPR0, LIN1STAPR1		√	√	√	√	–	–	–
INTLIN1	LIN1IF		LIN1MK		LIN1PR0, LIN1PR1		√	√	√	√	–	–	–

Table 22-3. Flags Corresponding to Interrupt Request Sources (6/7)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		RL78/F25				RL78/F22		
		Register		Register		Register	100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
INTADGB	ADGBIF	IF2H	ADGBMK	MK2H	ADGBPR0, ADGBPR1	PR02H, PR12H	√	√	√	√	√	√	√
INTROM	ROMIF		ROMMK		ROMPR0, ROMPR1		√	√	√	√	√	√	√
INTKR	KRIF		KRMK		KRPR0, KRPR1		√	√	√	√	√	√	√
INTRCAN0GRVC	CAN0GRVCIF		CAN0GRVCMK		CAN0GRVCPR0, CAN0GRVCPR1		√	√	√	√	–	–	–
INTRCAN0ERR	CAN0ERRIF		CAN0ERRMK		CAN0ERRPR0, CAN0ERRPR1		√	√	√	√	–	–	–
INTRCAN0WUP	CAN0WUPIF		CAN0WUPMK		CAN0WUPPR0, CAN0WUPPR1		√	√	√	√	–	–	–
INTRCAN0CFR	CAN0CFRIF	IF3L	CAN0CFRMK	MK3L	CAN0CFRPR0, CAN0CFRPR1	PR03L, PR13L	√	√	√	√	–	–	–
INTRCAN0TRM	CAN0TRMIF		CAN0TRMMK		CAN0TRMPR0, CAN0TRMPR1		√	√	√	√	–	–	–
INTRCAN0GRFR	CAN0GRFRIF		CAN0GRFRMK		CAN0GRFRPR0, CAN0GRFRPR1		√	√	√	√	–	–	–
INTRCAN0GERR	CAN0GERRIF		CAN0GERRMK		CAN0GERRPR0, CAN0GERRPR1		√	√	√	√	–	–	–
INTP12	PIF12		PMK12		PPR012, PPR112		√	√	√	–	–	–	–
INTLIN1WUP	LIN1WUPIF		LIN1WUPMK		LIN1WUPPR0, LIN1WUPPR1		√	√	√	√	–	–	–
INTRCAN1GRFR	CAN1GRFRIF		CAN1GRFRMK		CAN1GRFRPR0, CAN1GRFRPR1		√	√	√	√	–	–	–
INTRCAN1GERR	CAN1GERRIF		CAN1GERRMK		CAN1GERRPR0, CAN1GERRPR1		√	√	√	√	–	–	–
INTRCAN1WUP	CAN1WUPIF		CAN1WUPMK		CAN1WUPPR0, CAN1WUPPR1		√	√	√	√	–	–	–

Table 22-3. Flags Corresponding to Interrupt Request Sources (7/7)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		RL78/F25				RL78/F22		
		Register		Register		Register	100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
INTTM14	TMIF14	IF3H	TMMK14	MK3H	TMPR014, TMPR114	PR03H, PR13H	√	√	√	√	–	–	–
INTRCAN1GRVC	CAN1GRVCIF		CAN1GRVCMK		CAN1GRVCPR0, CAN1GRVCPR1		√	√	√	√	–	–	–
INTTM15	TMIF15		TMMK15		TMPR015, TMPR115		√	√	√	√	–	–	–
INTRCAN1ERR	CAN1ERRIF		CAN1ERRMK		CAN1ERRPR0, CAN1ERRPR1		√	√	√	√	–	–	–
INTTM16	TMIF16		TMMK16		TMPR016, TMPR116		√	√	√	√	–	–	–
INTRCAN1CFR	CAN1CFRIF		CAN1CFRMK		CAN1CFRPR0, CAN1CFRPR1		√	√	√	√	–	–	–
INTTM17	TMIF17		TMMK17		TMPR017, TMPR117		√	√	√	√	–	–	–
INTRCAN1TRM	CAN1TRMIF		CAN1TRMMK		CAN1TRMPR0, CAN1TRMPR1		√	√	√	√	–	–	–
INTCRAM	CRAMIF		CRAMMK		CRAMPR0, CRAMPR1		√	√	√	√	–	–	–

22.3.1 Interrupt Request Flag Registers (IFxL, IFxH)

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, and IF3H 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, and the IF3L and IF3H registers are combined to form 16-bit registers IF0, IF1, IF2, and IF3, 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 22-2. Format of Interrupt Request Flag Registers (IFxL, IFxH) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5 CMPIF0	PIF4 SPMIF	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	CTSUWRIF	RAMIF	TRJIF0	TRDIF1	TRDIF0	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	CLMIF PIF13

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03 LIN2STAIF LIN2IF	TMIF02 LIN2RVCIF	TMIF01 LIN2TRMIF	TMIF00	PIF8 RTCIF	IICAIF0	CTSUFNIF	CTSURDIF

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	TMIF04	SRIF1 CSIIF11 IICIF11	STIF1 CSIIF10 IICIF10	PIF10 TMIF03H	PIF9 TMIF01H	PIF7 TMIF13H	PIF6 LIN2WUPIF TMIF11H	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	TMIF13 LIN1STAIF LIN1IF	TMIF12 LIN1RVCIF	TMIF11 LIN1TRMIF	TMIF10	PIF11 LIN0WUPIF	TMIF07 LIN0STAIF LIN0IF	TMIF06 LIN0RVCIF	TMIF05 LIN0TRMIF

Figure 22-2. Format of Interrupt Request Flag Registers (IFxL, IFxH) (2/2)

Address: FFFD1H After reset: 0000x000B R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	1	0
IF2H	0	CAN0WUPIF	CAN0ERRIF	KRIF CAN0GRVCIF	ROMIF Note	ADGBIF	0	0

Address: FFFD2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF3L	CAN1WUPIF	CAN1GERRIF	CAN1GRFRIF	PIF12 LIN1WUPIF	CAN0GERRIF	CAN0GRFRIF	CAN0TRMIF	CAN0CFRIF

Address: FFFD3H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
IF3H	0	0	0	CRAMIF	TMIF17 CAN1TRMIF	TMIF16 CAN1CFRIF	TMIF15 CAN1ERRIF	TMIF14 CAN1GRVCIF

IFxx	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Note When reset is released, the ROMIF flag indicates an undefined value. Clear this flag to 0 by software before setting the ROMMK flag of the MK2H register to 0 (enabling the code flash ECC error detection (DED/SED) interrupt).

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 22-3. 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 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.

22.3.2 Interrupt Mask Flag Registers (MKxL, MKxH)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

The MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, and MK3H 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, the MK2L and MK2H registers, and the MK3L and MK3H registers are combined to form 16-bit registers MK0, MK1, MK2, and MK3, 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 22-3. Format of Interrupt Mask Flag Registers (MKxL, MKxH) (1/2)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5 CMPMK0	PMK4 SPMMK	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	CTSUWRMK	RAMMK	TRJMK0	TRDMK1	TRDMK0	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	PMK13 CLMMK

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03 LIN2STAMK LIN2MK	TMMK02 LIN2RVCMK	TMMK01 LIN2TRMMK	TMMK00	PMK8 RTCMK	IICAMK0	CTSUFNMK	CTSURDMK

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	TMMK04	SRMK1 CSIMK11 IICMK11	STMK1 CSIMK10 IICMK10	PMK10 TMMK03H	PMK9 TMMK01H	PMK7 TMMK13H	PMK6 LIN2WUPMK TMMK11H	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	TMMK13 LIN1STAMK LIN1MK	TMMK12 LIN1RVCMK	TMMK11 LIN1TRMMK	TMMK10	PMK11 LIN0WUPMK	TMMK07 LIN0STAMK LIN0MK	TMMK06 LIN0RVCMK	TMMK05 LIN0TRMMK

Address: FFFD5H After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	1	0
MK2H	1	CAN0WUPMK	CAN0ERRMK	KRMK CAN0GRVCMK	ROMMK	ADGBMK	1	1

Figure 22-3. Format of Interrupt Mask Flag Registers (MKxL, MKxH) (2/2)

Address: FFFD6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK3L	CAN1WUPMK	CAN1GERRMK	CAN1GRFRMK	PMK12 LIN1WUPMK	CAN0GERRMK	CAN0GRFRMK	CAN0TRMMK	CAN0CFRMK

Address: FFFD7H After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
MK3H	1	1	1	CRAMMK	TMMK17 CAN1TRMMK	TMMK16 CAN1CFRMK	TMMK15 CAN1ERRMK	TMMK14 CAN1GRVCMK

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 22-3. Be sure to set bits that are not available to 1.

22.3.3 Priority Specification Flag Registers (PRxxL, PRxxH)

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, 3L, or 3H).

The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and the PR13H 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 PR03L and PR03H registers, the PR10L and PR10H registers, the PR11L and PR11H registers, the PR12L and PR12H registers, and the PR13L and PR13H registers are combined to form 16-bit registers PR00, PR01, PR02, PR03, PR10, PR11,PR12, and PR13, 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 22-4. Format of Priority Specification Flag Registers (PRxxL, PRxxH) (1/3)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05 CMPPR00	PPR04 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 CMPPR10	PPR14 SPMPR1	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	CTSUWRPR0	RAMPR0	TRJPR00	TRDPR01	TRDPR00	SRPR00 CSIPR001 IICPR001	STPR00 CSIPR000 IICPR000	PPR013 CLMPR0

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	CTSUWRPR1	RAMPR1	TRJPR10	TRDPR11	TRDPR10	SRPR10 CSIPR101 IICPR101	STPR10 CSIPR100 IICPR100	PPR113 CLMPR1

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003 LIN2STAPR0 LIN2PR0	TMPR002 LIN2RVCPR0	TMPR001 LIN2TRMPR0	TMPR000	PPR08 RTCPR0	IICAPR00	CTSUFNPR0	CTSURDPR0

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103 LIN2STAPR1 LIN2PR1	TMPR102 LIN2RVCPR1	TMPR101 LIN2TRMPR1	TMPR100	PPR18 RTCPR1	IICAPR10	CTSUFNPR1	CTSURDPR1

Figure 22-4. Format of Priority Specification Flag Registers (PRxxL, PRxxH) (2/3)

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	TMPR004	SRPR01 CSIPR011 IICPR011	STPR01 CSIPR010 IICPR010	PPR010 TMPR003H	PPR09 TMPR001H	PPR07 TMPR013H	PPR06 LIN2WUPPR0 TMPR011H	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	TMPR104	SRPR11 CSIPR111 IICPR111	STPR11 CSIPR110 IICPR110	PPR110 TMPR103H	PPR19 TMPR101H	PPR17 TMPR113H	PPR16 LIN2WUPPR1 TMPR111H	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	TMPR013 LIN1STAPR0 LIN1PR0	TMPR012 LIN1RVCPR0	TMPR011 LIN1TRMPR0	TMPR010	PPR011 LIN0WUPPR0	TMPR007 LIN0STAPR0 LIN0PR0	TMPR006 LIN0RVCPR0	TMPR005 LIN0TRMPR0

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	TMPR113 LIN1STAPR1 LIN1PR1	TMPR112 LIN1RVCPR1	TMPR111 LIN1TRMPR1	TMPR110	PPR111 LIN0WUPPR1	TMPR107 LIN0STAPR1 LIN0PR1	TMPR106 LIN0RVCPR1	TMPR105 LIN0TRMPR1

Address: FFFD9H After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	1	0
PR02H	1	CAN0WUPPR0	CAN0ERRPR0	KRPR0 CAN0GRVCPR0	ROMPR0	ADGBPR0	1	1

Address: FFFDDH After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	1	0
PR12H	1	CAN0WUPPR1	CAN0ERRPR1	KRPR1 CAN0GRVCPR1	ROMPR1	ADGBPR1	1	1

Address: FFFDAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR03L	CAN1WUPPR0	CAN1GERRPR0	CAN1GRFRPR0	PPR012 LIN1WUPPR0	CAN0GERRPR0	CAN0GRFRPR0	CAN0TRMPR0	CAN0CFRPR0

Address: FFFDEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR13L	CAN1WUPPR1	CAN1GERRPR1	CAN1GRFRPR1	PPR112 LIN1WUPPR1	CAN0GERRPR1	CAN0GRFRPR1	CAN0TRMPR1	CAN0CFRPR1

Figure 22-4. Format of Priority Specification Flag Registers (PRxxL, PRxxH) (3/3)

Address: FFFDBH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR03H	1	1	1	CRAMPR0	TMPR017 CAN1TRMPR0	TMPR016 CAN1CFRPR0	TMPR015 CAN1ERRPR0	TMPR014 CAN1GRVCPR0

Address: FFFDFH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR13H	1	1	1	CRAMPR1	TMPR117 CAN1TRMPR1	TMPR116 CAN1CFRPR1	TMPR115 CAN1ERRPR1	TMPR114 CAN1GRVCPR1

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 22-3. Be sure to set bits that are not available to 1.

22.3.4 External Interrupt Rising/Falling Edge Enable Registers (EGP0, EGP1, 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 22-5. Format of External Interrupt Rising/Falling Edge Enable Registers (EGP0, EGP1, 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 22-4 shows the ports corresponding to the EGPn and EGNn bits.

Table 22-4. Ports Corresponding to EGPn and EGNn bits

Detection Enable Bit		Edge Detection Port	Interrupt Request Signal	RL78/F25				RL78/F22		
				100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
EGP0	EGN0	P137	INTP0	√	√	√	√	√	√	√
EGP1	EGN1	P125	INTP1	√	√	√	√	√	√	–
EGP2	EGN2	P30 (P31)	INTP2	√	√	√	√	√	√ Note 3	√ Note 3
EGP3	EGN3	P17 (P50)	INTP3	√	√	√	√ Note 3	√ Note 3	√ Note 3	√ Note 3
EGP4	EGN4	P120	INTP4	√	√	√	√	√	√	√
EGP5	EGN5	P12	INTP5	√	√	√	√	√	√	√
EGP6	EGN6	P71	INTP6	√	√	√	√	√	–	–
			INTLIN2WUP Note 1	√	√	√	√	–	–	–
EGP7	EGN7	P32	INTP7	√	√	√	√	√	–	–
EGP8	EGN8	P70	INTP8	√	√	√	√	√	–	–
EGP9	EGN9	P00	INTP9	√	√	√	√	√	–	–
EGP10	EGN10	P53	INTP10	√	√	√	–	–	–	–
EGP11	EGN11	P51	INTP11	√	√	√	–	–	–	–
			INTLIN0WUP Note 2	√	√	√	√	√	√	√
EGP12	EGN12	P77	INTP12	√	√	√	–	–	–	–
			INTLIN1WUP Note 2	√	√	√	√	–	–	–
EGP13	EGN13	P47	INTP13	√	√	–	–	–	–	–

Notes 1. Set the EGP0 and EGN0 registers before the INTLIN2WUP interrupt is generated.

2. Set the EGP1 and EGN1 registers before the INTLIN0WUP and INTLIN1WUP interrupts are generated.

3. The pins shown in parentheses are not provided by this product.

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 both bits EGPn and EGNn to 0).

Remarks 1. For edge detection ports, see **2.1 Pin Function List**.

2. n = 0 to 13

22.3.5 Interrupt Source Determination Flag Register 0 (INTFLG0)

This register functions to determine which of the external interrupt sources (INTP4, 5, 8, 13), comparator detect 0 interrupt source, and other sources that also share the same vector table address caused the interrupt.

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.

For details on how to use the interrupt source determination flag, see **22.4.1 Determination of Interrupt Source**.

Figure 22-6. 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 Notes 3, 5	INTFLG06 Notes 2, 4	0	0	0	INTFLG02 Notes 1, 5	INTFLG01 Note 5	INTFLG00 Note 5

INTFLG07 Notes 3, 5	Interrupt source determination flag at vector table address 0014H
0	An INTP13 interrupt has not been generated.
1	An INTP13 interrupt has been generated.

INTFLG06 Notes 2, 4	Interrupt source determination flag at vector table address 0012H
0	A comparator detection 0 interrupt has not been generated.
1	A comparator detection 0 interrupt has been generated.

INTFLG02 Notes 1, 5	Interrupt source determination flag at vector table address 002AH
0	An INTP8 interrupt has not been generated.
1	An INTP8 interrupt has been generated.

INTFLG01 Note 5	Interrupt source determination flag at vector table address 0012H
0	An INTP5 interrupt has not been generated.
1	An INTP5 interrupt has been generated.

INTFLG00 Note 5	Interrupt source determination flag at vector table address 0010H
0	An INTP4 interrupt has not been generated.
1	An INTP4 interrupt has been generated.

Notes 1. Not provided for 24-pin and 32-pin products.

2. Provided only in RL78/F25 products.

3. Provided only in 100-pin or 80-pin products.

4. 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.

5. 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)

22.3.6 Interrupt Source Determination Flag Register 1 (INTFLG1)

This register functions to determine which of the interrupt sources (INTTM01 to INTTM03, INTTM05 to INTTM07) and other interrupt sources that share the same vector table address caused the interrupt.

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.

For details on how to use the interrupt source determination flag, see **22.4.1 Determination of Interrupt Source**.

Figure 22-7. Format of Interrupt Source Determination Flag Register 1 (INTFLG1)

Address: F007DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
INTFLG1	INTFLG17	INTFLG16	INTFLG15	0	INTFLG13	INTFLG12	INTFLG11	0

INTFLG17	Interrupt source determination flag at vector table address 0048H
0	An INTTM07 interrupt has not been generated.
1	An INTTM07 interrupt has been generated.

INTFLG16	Interrupt source determination flag at vector table address 0046H
0	An INTTM06 interrupt has not been generated.
1	An INTTM06 interrupt has been generated.

INTFLG15	Interrupt source determination flag at vector table address 0044H
0	An INTTM05 interrupt has not been generated.
1	An INTTM05 interrupt has been generated.

INTFLG13	Interrupt source determination flag at vector table address 0032H
0	An INTTM03 interrupt has not been generated.
1	An INTTM03 interrupt has been generated.

INTFLG12	Interrupt source determination flag at vector table address 0030H
0	An INTTM02 interrupt has not been generated.
1	An INTTM02 interrupt has been generated.

INTFLG11	Interrupt source determination flag at vector table address 002EH
0	An INTTM01 interrupt has not been generated.
1	An INTTM01 interrupt has been generated.

22.3.7 Interrupt Source Determination Flag Register 2 (INTFLG2)

This register functions to determine which of the interrupt sources (INTTM11 to INTTM17) and other interrupt sources that share the same vector table address caused the interrupt.

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.

For details on how to use the interrupt source determination flag, see **22.4.1 Determination of Interrupt Source**.

Figure 22-8. Format of Interrupt Source Determination Flag Register 2 (INTFLG2)

Address: F01F2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
INTFLG2	INTFLG27	INTFLG26	INTFLG25	INTFLG24	INTFLG23	INTFLG22	INTFLG21	0

INTFLG27	Interrupt source determination flag at vector table address 007AH
0	An INTTM17 interrupt has not been generated.
1	An INTTM17 interrupt has been generated.

INTFLG26	Interrupt source determination flag at vector table address 0078H
0	An INTTM16 interrupt has not been generated.
1	An INTTM16 interrupt has been generated.

INTFLG25	Interrupt source determination flag at vector table address 0076H
0	An INTTM15 interrupt has not been generated.
1	An INTTM15 interrupt has been generated.

INTFLG24	Interrupt source determination flag at vector table address 0074H
0	An INTTM14 interrupt has not been generated.
1	An INTTM14 interrupt has been generated.

INTFLG23	Interrupt source determination flag at vector table address 0052H
0	An INTTM13 interrupt has not been generated.
1	An INTTM13 interrupt has been generated.

INTFLG22	Interrupt source determination flag at vector table address 0050H
0	An INTTM12 interrupt has not been generated.
1	An INTTM12 interrupt has been generated.

INTFLG21	Interrupt source determination flag at vector table address 004EH
0	An INTTM11 interrupt has not been generated.
1	An INTTM11 interrupt has been generated.

Caution This register is not provided in RL78/F22 products.

22.3.8 Interrupt Source Determination Flag Register 3 (INTFLG3)

This register functions to determine which of the interrupt sources (INTRCAN1GRVC, INTRCAN1ERR, INTRCAN1CFR, INTRCAN1TRM) and other interrupt sources that share the same vector table address caused the interrupt.

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.

For details on how to use the interrupt source determination flag, see **22.4.1 Determination of Interrupt Source**.

Figure 22-9. Format of Interrupt Source Determination Flag Register 3 (INTFLG3)

Address: F01F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
INTFLG3	0	0	INTFLG35	INTFLG34	INTFLG33	INTFLG32	0	0

INTFLG35	Interrupt source determination flag at vector table address 007AH
0	An INTRCAN1TRM interrupt has not been generated.
1	An INTRCAN1TRM interrupt has been generated.

INTFLG34	Interrupt source determination flag at vector table address 0078H
0	An INTRCAN1CFR interrupt has not been generated.
1	An INTRCAN1CFR interrupt has been generated.

INTFLG33	Interrupt source determination flag at vector table address 0076H
0	An INTRCAN1ERR interrupt has not been generated.
1	An INTRCAN1ERR interrupt has been generated.

INTFLG32	Interrupt source determination flag at vector table address 0074H
0	An INTRCAN1GRVC interrupt has not been generated.
1	An INTRCAN1GRVC interrupt has been generated.

Caution This register is not provided in RL78/F22 products.

22.3.9 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 22-10. 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 Note 1	INTMSK1	INTMSK0

INTMSK2 Note 1	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	Setting masking for INTP5 interrupt source to the interrupt control circuit Note 2
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 2
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 24-pin and 32-pin products.

2. 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)

22.3.10 Input Switch Control Register (ISC)

The ISC0 bit of the ISC register is used for the LIN-bus communication with UART0. The ISC2, ISC3, and ISC4 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, 3, and 4 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 a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 22-11. 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	ISC4	ISC3	ISC2	0	ISC0

ISC4	Input selection for external interrupt INTP6
0	INTP6 pin input signal is selected as external interrupt input.
1	LRxD2 pin input signal is selected as external interrupt input.

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)

Cautions 1. Bits 7 to 5 and 1 should always be set to 0.

2. Be sure to set the ISC3, ISC4 bits to 0 in RL78/F22 products.

3. The RL78/F22 products do not have the INTP11 function. Therefore, when the ISC2 bit is 0, no external interrupt (INTP11) is generated.

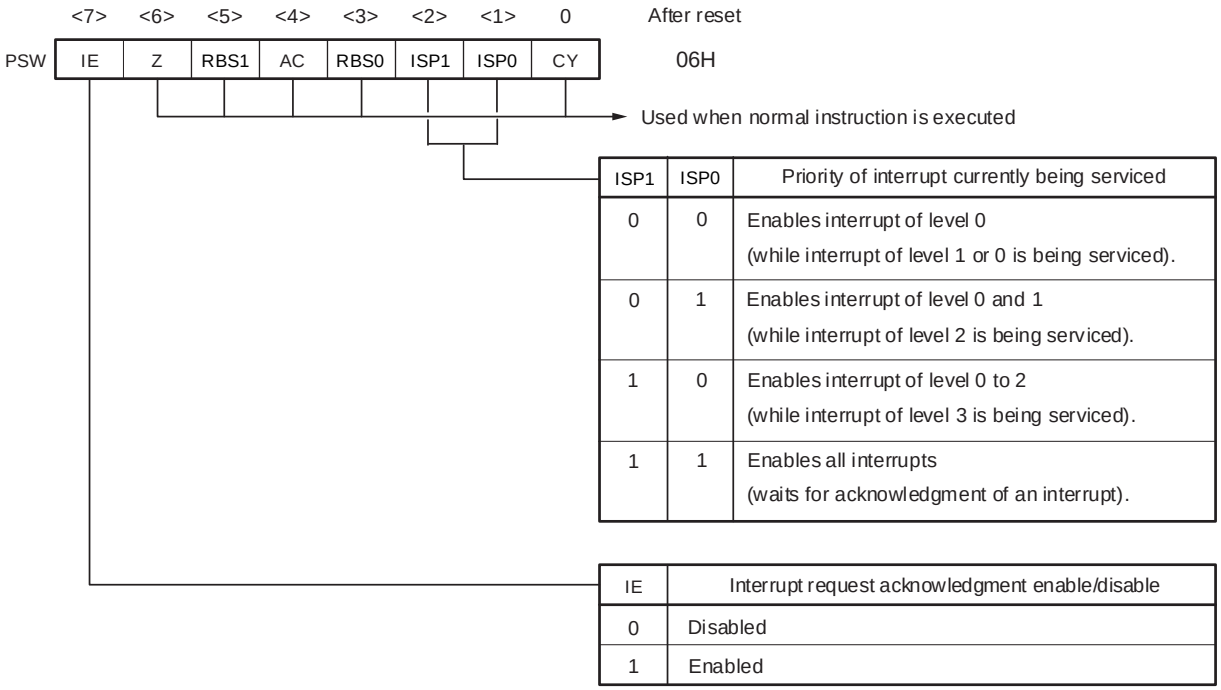
22.3.11 Program Status Word (PSW)

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 22-12. Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

22.4 Interrupt Servicing Operations

22.4.1 Determination of Interrupt Source

Some interrupt sources share the same vector table address. Interrupt sources are determined using the interrupt source determination flag registers (INTFLG0-INTFLG3) and the register or flags of the interrupt request source.

Table 22-5. Determination of Interrupt Source (1/2)

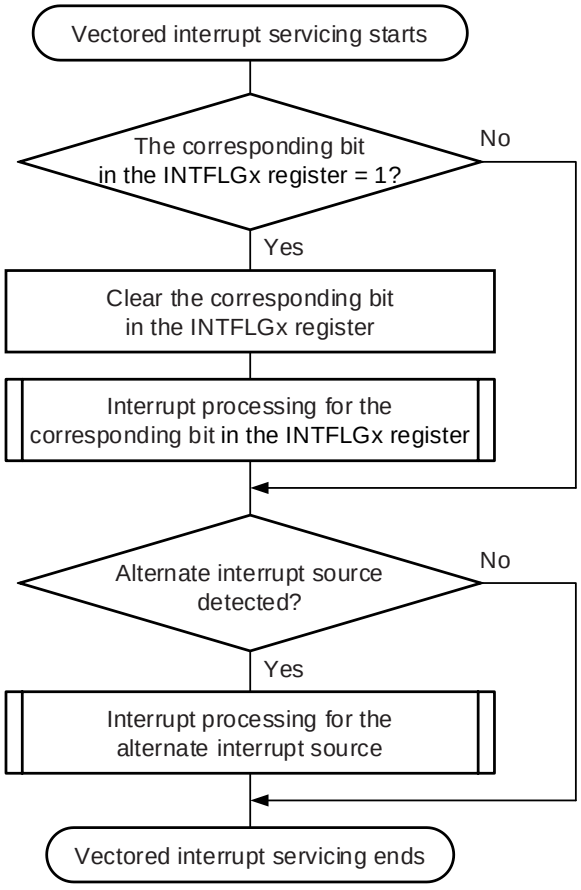
Vector Table Address	Interrupt Sources (Sharing the Same Vector Table Address)		Determination of Interrupt Source
0010H	INTP4	Pin input edge detection 4	INTFLG00 bit of INTFLG0 register
	INTSPM	Stack pointer overflow/underflow	Stack pointer
0012H	INTP5	Pin input edge detection 5	INTFLG01 bit of INTFLG0 register
	INTCMP0	Comparator detection 0	INTFLG06 bit of INTFLG0 register
0014H	INTP13	Pin input edge detection 13	INTFLG07 bit of INTFLG0 register
	INTCLM	PLL clock stop	SELPLLS bit of the PLLSTS
002AH	INTP8	Pin input edge detection 8	INTFLG02 bit of INTFLG0 register
	INTRTC	RTC pretimed signal or alarm match detection	WAFG, RIFG bits of the RTCC1 register
002EH	INTTM01	End of TAU0 channel 1 count/capture	INTFLG11 bit of INTFLG1 register
	INTLIN2TRM	LIN2 transmission	(See section **18.9**)
0030H	INTTM02	End of TAU0 channel 2 count/capture	INTFLG12 bit of INTFLG1 register
	INTLIN2RVC	LIN2 reception end	(See section **18.9**)
0032H	INTTM03	End of TAU0 channel 3 count/capture	INTFLG13 bit of INTFLG1 register
	INTLIN2STA/INTLIN2	LIN2 reception status/LIN2 interrupt	(See section **18.9**)
0036H	INTP6	Pin input edge detection 6	Select either INTP6 or INTLIN2WUP with the ISC4 bit of the ISC register. If the selected interrupt source and INTTM11H are used at the same time, it cannot be determined.
	INTLIN2WUP	LIN2 reception pin input	
	INTTM11H	Upper 8-bit interval timer interrupt of TAU1 channel 1 (when 8-bit timer function is selected)	
0038H	INTP7	Pin input edge detection 7	If these functions are used at the same time, the cause of the interrupt cannot be determined.
	INTTM13H	Upper 8-bit interval timer interrupt of TAU1 channel 3 (when 8-bit timer function is selected)	
003AH	INTP9	Pin input edge detection 9	If these functions are used at the same time, the cause of the interrupt cannot be determined.
	INTTM01H	Upper 8-bit interval timer interrupt of TAU0 channel 1 (when 8-bit timer function is selected)	
003CH	INTP10	Pin input edge detection 10	If these functions are used at the same time, the cause of the interrupt cannot be determined.
	INTTM03H	Upper 8-bit interval timer interrupt of TAU0 channel 3 (when 8-bit timer function is selected)	
0044H	INTTM05	End of TAU0 channel 5 count/capture	INTFLG15 bit of INTFLG1 register
	INTLIN0TRM	LIN0 transmission	(See section **18.9**)
0046H	INTTM06	End of TAU0 channel 6 count/capture	INTFLG16 bit of INTFLG1 register
	INTLIN0RVC	LIN0 reception end	(See section **18.9**)
0048H	INTTM07	End of TAU0 channel 7 count/capture	INTFLG17 bit of INTFLG1 register
	INTLIN0STA/INTLIN0	LIN0 reception status/LIN0 interrupt	(See section **18.9**)
004AH	INTP11	Pin input edge detection 11	Select either INTP11 or INTLIN0WUP with the ISC2 bit of the ISC register.
	INTLIN0WUP	LIN0 reception pin input	
004EH	INTTM11	End of TAU1 channel 1 count/capture	INTFLG21 bit of INTFLG2 register
	INTLIN1TRM	LIN1 transmission	(See section **18.9**)
0050H	INTTM12	End of TAU1 channel 2 count/capture	INTFLG22 bit of INTFLG2 register
	INTLIN1RVC	LIN1 reception end	(See section **18.9**)
0052H	INTTM13	End of TAU1 channel 3 count/capture	INTFLG23 bit of INTFLG2 register
	INTLIN1STA/INTLIN1	LIN1 reception status/LIN1 interrupt	(See section **18.9**)

Table 22-5. Determination of Interrupt Source (2/2)

Vector Table Address	Interrupt Sources (Sharing the Same Vector Table Address)		Determination of Interrupt Source
005CH	INTKR	Key interrupt detection	If these functions are used at the same time, the cause of the interrupt cannot be determined.
	INTRCAN0GRVC	CAN0 global receive message buffer reception	
006CH	INTP12	Pin input edge detection 12	Select either INTP12 or INTLIN1WUP with the ISC3 bit of the ISC register.
	INTLIN1WUP	LIN1 reception pin input	
0074H	INTTM14	End of TAU1 channel 4 count/capture	INTFLG24 bit of INTFLG2 register
	INTRCAN1GRVC	CAN1 global receive message buffer reception	INTFLG32 bit of INTFLG3 register
0076H	INTTM15	End of TAU1 channel 5 count/capture	INTFLG25 bit of INTFLG2 register
	INTRCAN1ERR	CAN1 channel error detection	INTFLG33 bit of INTFLG3 register
0078H	INTTM16	End of TAU1 channel 6 count/capture	INTFLG26 bit of INTFLG2 register
	INTRCAN1CFR	CAN1 transmit/receive FIFO message reception	INTFLG34 bit of INTFLG3 register
007AH	INTTM17	End of TAU1 channel 7 count/capture	INTFLG27 bit of INTFLG2 register
	INTRCAN1TRM	CAN1 channel transmit	INTFLG35 bit of INTFLG3 register

Figure 22-13 shows the flowchart of interrupt processing using the interrupt source determination flag register.

Figure 22-13. Flowchart of Interrupt Processing Using the Interrupt Source Determination Flag Register (INTFLGx)



Remark INTFLGx (x = 0 to 3)

22.4.2 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 22-6** below.

For the interrupt request acknowledgment timing, see **Figure 22-15** and **Figure 22-16**.

Table 22-6. 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.

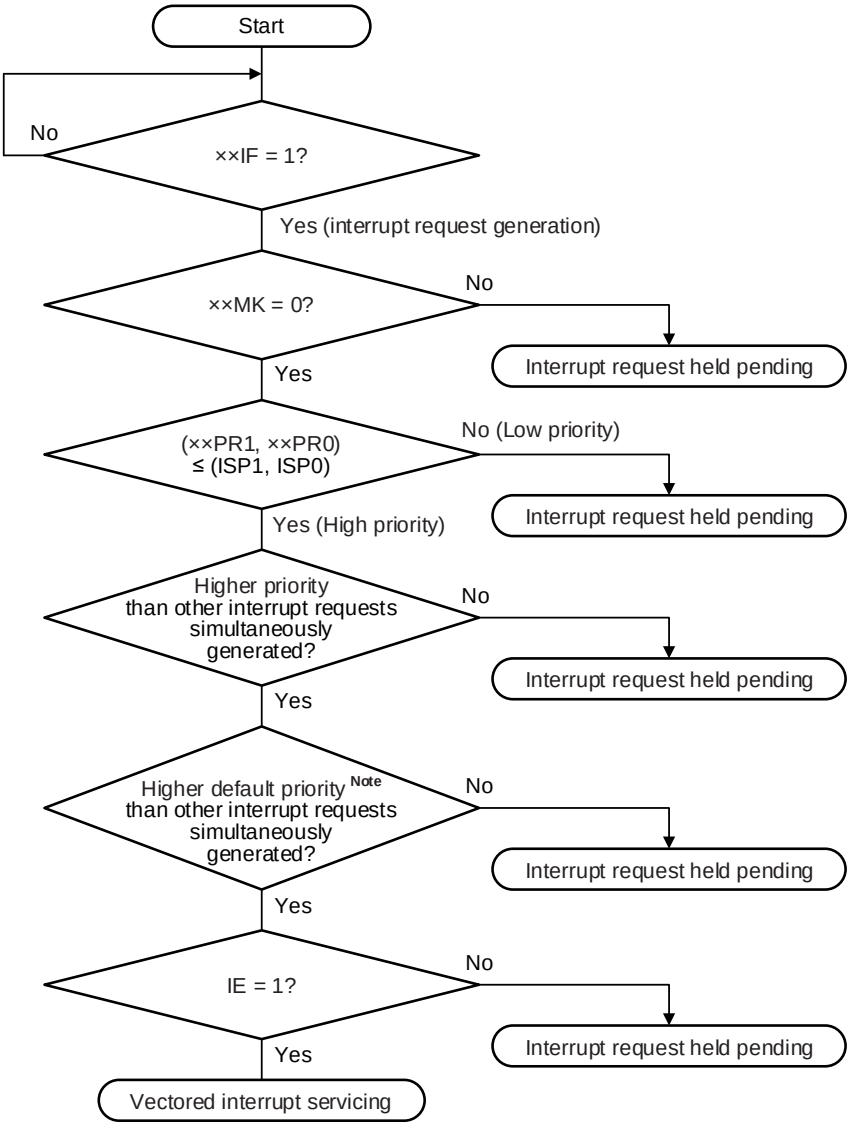
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 22-14 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.

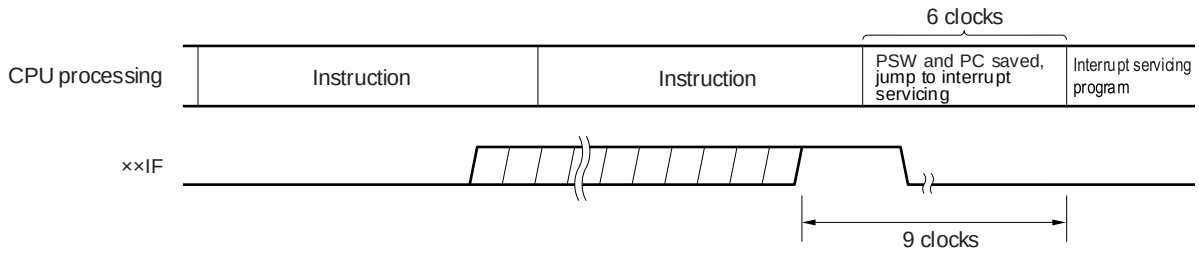
Figure 22-14. Interrupt Request Acknowledgment Processing Algorithm



××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

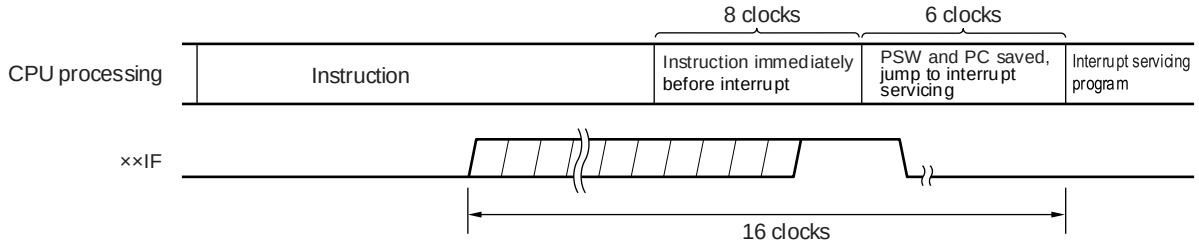
Note For the default priority, see **Table 22-1 Interrupt Source List**.

Figure 22-15. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

Figure 22-16. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

22.4.3 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 (007EH, 007FH) 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.

22.4.4 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 22-7 shows relationship between interrupt requests enabled for multiple interrupt servicing and **Figure 22-17** shows multiple interrupt servicing examples.

Table 22-7. 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, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H registers.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)

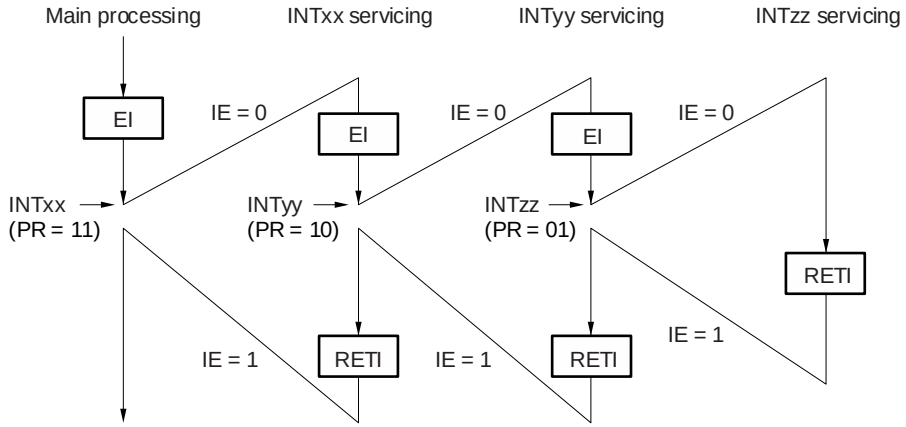
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

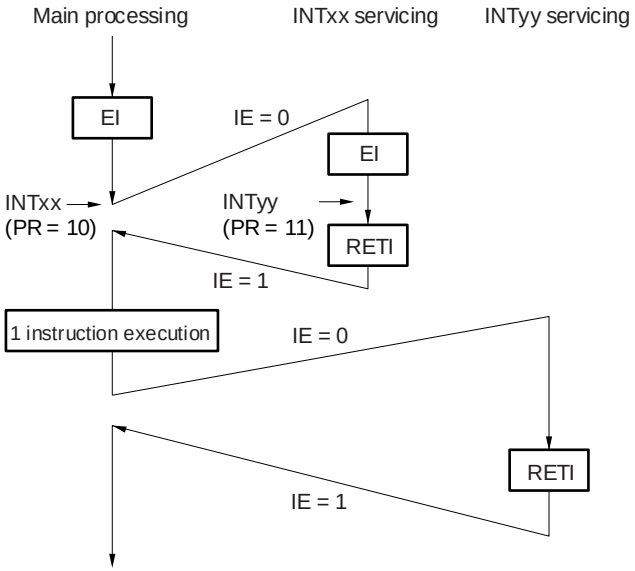
Figure 22-17. Examples of Multiple Interrupt Servicing (1/2)

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

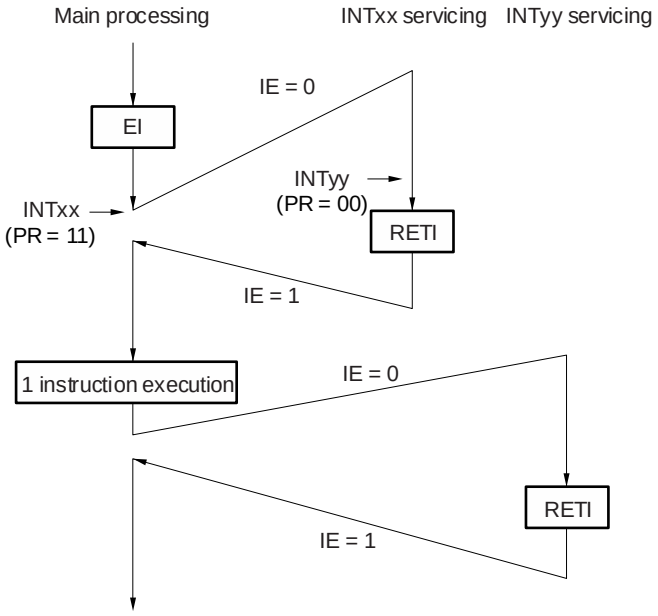


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

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 22-17. 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.

22.4.5 Interrupt Servicing During Division Instruction

The RL78/F22 and RL78/F25 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 22-8. Normal Interrupt Processing and Interrupt Processing while Executing Division Instructions

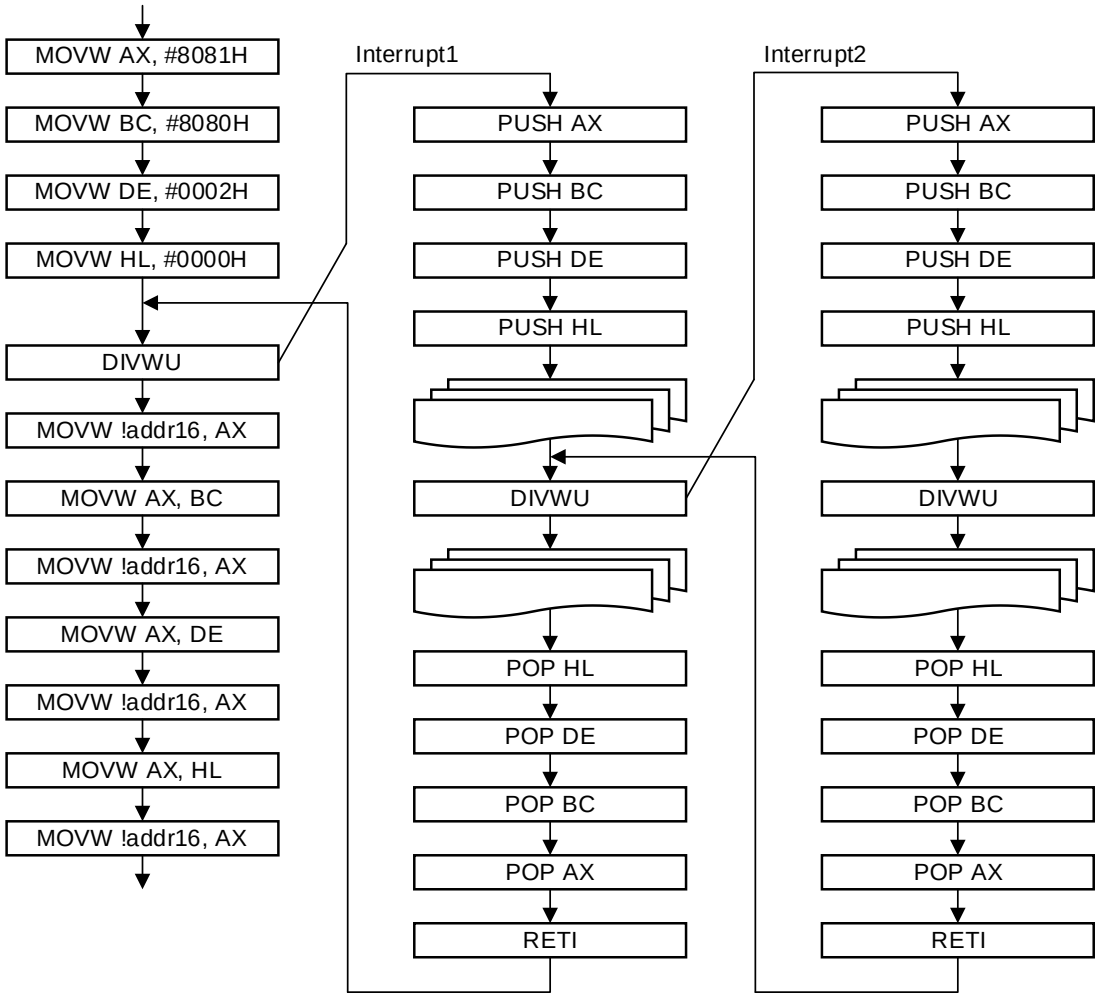
Normal interrupt	Interrupts while Executing DIVHU/DIVWU Instruction
(SP-1) ← PSW	(SP-1) ← PSW
(SP-2) ← $(PC)_S$	(SP-2) ← $(PC-3)_S$
(SP-3) ← $(PC)_H$	(SP-3) ← $(PC-3)_H$
(SP-4) ← $(PC)_L$	(SP-4) ← $(PC-3)_L$
PC_S ← 0000	PC_S ← 0000
PC_H ← (Vector)	PC_H ← (Vector)
PC_L ← (Vector)	PC_L ← (Vector)
SP ← SP-4	SP ← SP-4
IE ← 0	IE ← 0

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 22-18. Example of Interrupt during Division Instruction



22.4.6 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 **Note**. 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
- MULHU
- MULH
- MACHU
- MACH
- Manipulation instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, and PR13H registers

Note If the interrupt request hold instructions are consecutive, the interrupt hold state will also continue.

Example 1) Interrupt requests are held pending.

```
EI ; Enable interrupt instruction        Interrupt request hold instruction
DI ; Disable interrupt instruction       Interrupt request hold instruction
```

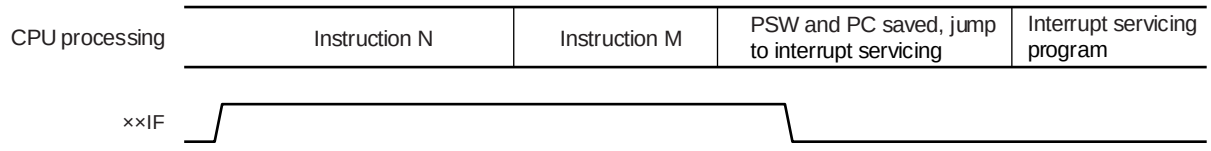
Example 2) Interrupt request is accepted.

```
EI ; Enable interrupt instruction        Interrupt request hold instruction
NOP ;
DI ; Disable interrupt instruction       Interrupt request hold instruction
```

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 22-19 shows the timing at which interrupt requests are held pending.

Figure 22-19. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 23 KEY INTERRUPT FUNCTION

The number of key interrupt input channels differs, depending on the product.

	RL78/F25				RL78/F22		
	100 pin	80 pin	64 pin	48 pin	48 pin	32 pin	24 pin
Key interrupt input channels	8 ch				8 ch	6 ch	3 ch

23.1 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 23-1. 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 assignment of the KRn function to the I/O port is selected by the PIOR50 bit in the PIOR5 register. Among the ports that can be assigned with the PIOR50 bit setting, P90-P96, P80-P87, and P70-P74 are multiplexed with A/D converter and are initially set as analog input ports. When using the key interrupt function, switch to the digital input port with the PIOR50 bit and the PMC7/PMC8/PMC9 registers before using the key interrupt function. For details of the PIOR50 bit and the PMC7/PMC8/PMC9 register, refer to 4.3.14 Peripheral I/O Redirection Register 5 (PIOR5) and 4.3.6 Port Mode Control Registers (PMCm).

Remarks 1. n = 0 to 7

2. The available number of interrupts depends on the setting of the PIOR50. For details, refer to **4.3.14 Peripheral I/O redirection register 5 (PIOR5)**.

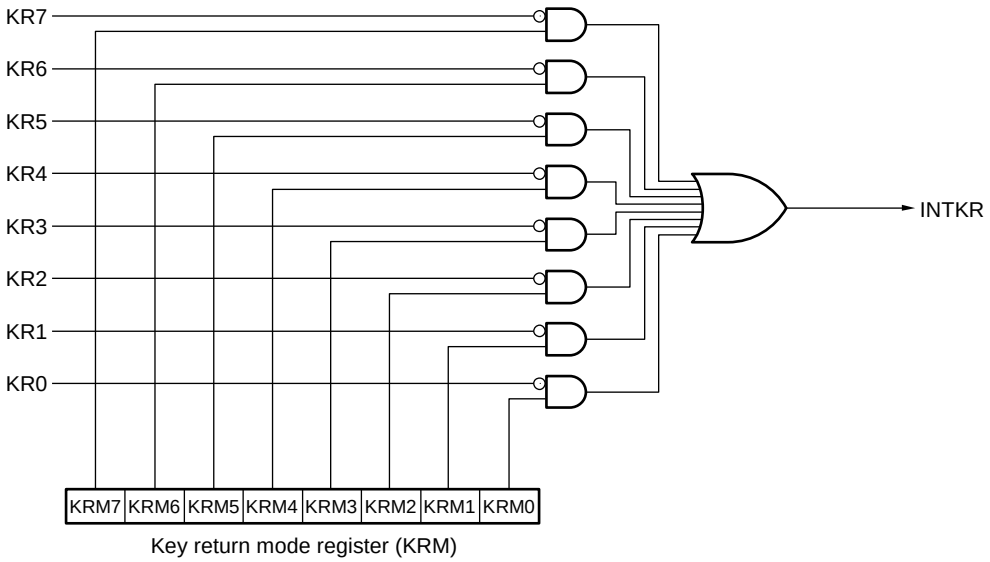
23.2 Configuration of Key Interrupt

Table 23-2 shows the configuration of the key interrupt. **Figure 23-1** is the block diagram of the key interrupt.

Table 23-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return mode register (KRM)

Figure 23-1. Block Diagram of Key Interrupt



23.3 Register Controlling Key Interrupt

The following register is used to control the key interrupt function.

Table 23-3. Key Interrupt Function Register

Address	Register Name	Symbol	After Reset	Access Size
FFF37H	Key return mode register	KRM	00H	1, 8

23.3.1 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 23-2. Format of Key Return Mode Register (KRM)

Addres: 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 24 STANDBY FUNCTION

24.1 Standby Function and Configuration

24.1.1 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 signal for data reception by the LIN/UART module (RLIN3) in the UART mode, a DTC activation signal, or SNOOZE operation request of the capacitive sensing unit (CTSU2SLa), this LSI is released from the STOP mode and 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.

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 LIN/UART module in the SNOOZE mode, set up the UART standby control register (LUSCn) before switching to the STOP mode. For details, see 17.2 Register Descriptions.**
4. **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 32 OPTION BYTE.**

24.2 Registers Controlling Standby Function

The following registers are used to control the standby function.

Table 24-1. Standby Function Registers Configuration

Address	Register Name	Symbol	After Reset	Access Size
F02C6H	STOP status output control register	STPSTC	00H	1, 8
FFFA2H	Oscillation stabilization time counter status register	OSTC	00H	1, 8
FFFA3H	Oscillation stabilization time select register	OSTS	07H	8
F00F9H	Memory power reduction control register	PSMCR	00H	1, 8

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)

Remarks 1. For the registers that start, stop, or select the clock, see **CHAPTER 5 CLOCK GENERATOR**.

2. For the SNOOZE status output control registers (PSNZCNT0, PSNZCNT1, PSNZCNT2, and PSNZCNT3), see **CHAPTER 4 PORT FUNCTIONS**.

24.2.1 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 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 24-1. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H After reset: 00H R

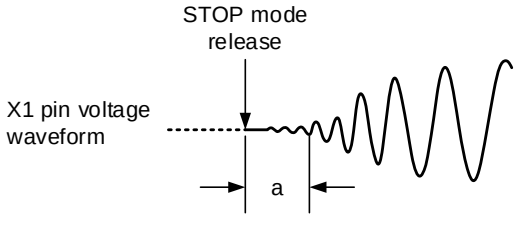
Symbol	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18

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

24.2.2 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 24-2. Format of Oscillation Stabilization Time Select Register (OSTS)

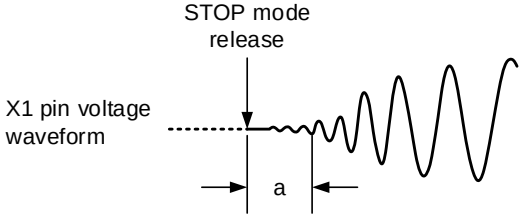
Address: FFFA3H After reset: 07H R/W

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

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. **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

24.2.3 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 24-, 32-pin products do not have the STOPST function, it does 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 24-3. Format of STOP Status Output Control Register (STPSTC)

Address: F02C6H 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

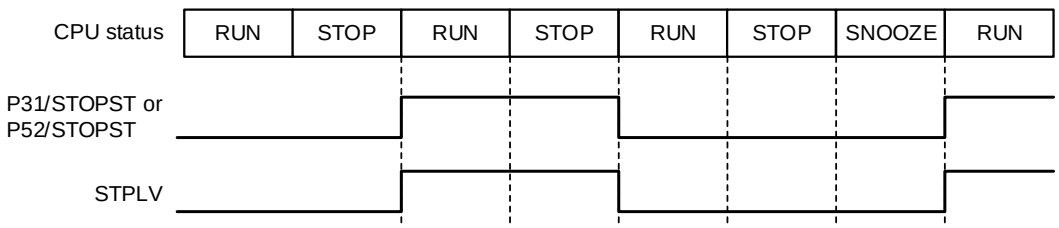
STPSEL Note 2	Control of STOPST pin selection
0	Selects P31
1	Selects P52

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.



24.2.4 Memory Power Reduction Control Register (PSMCR)

The PSMCR register controls the RAM power consumption reduction mode. Leakage current can be reduced by setting the RAM to shutdown mode. In shutdown mode, RAM does not retain data because the power supply is turned off.

Set the PSMCR register by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 24-4. Format of Memory Power Reduction Control Register (PSMCR)

Address: F00F9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PSMCR	0	0	0	0	0	0	RAMSDMD	RAMSDS

RAMSDMD	RAMSDS	Operating mode of the RAM
0	0	Normal mode (continues to operate)
1	0	Standby mode
1	1	Shutdown mode
Other than above		Setting is prohibited

Cautions 1. The following RAM areas are not targets for shutdown mode. RAM areas other than the targets continue to operate and retain data.

RL78/F22: FCF00H to FDFFFH and FF000H to FFEFFH

RL78/F25: F5F00H to F6FFFH and FF000H to FFEFFH

2. Do not access target RAM while it is in the standby mode or shutdown mode.

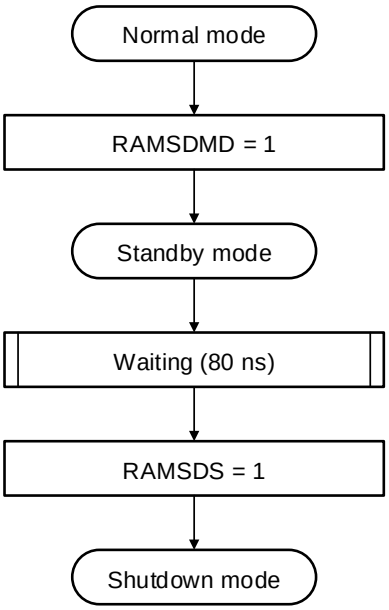
3. When returning from shutdown mode to normal mode, the target RAM area becomes indefinite.

4. During on-chip debugging, the RAM operates in normal mode regardless of the setting of the memory power reduction control register (PSMCR).

5. When the hot plug-in is connected, the RAM operates according to the memory power reduction control register (PSMCR) setting.

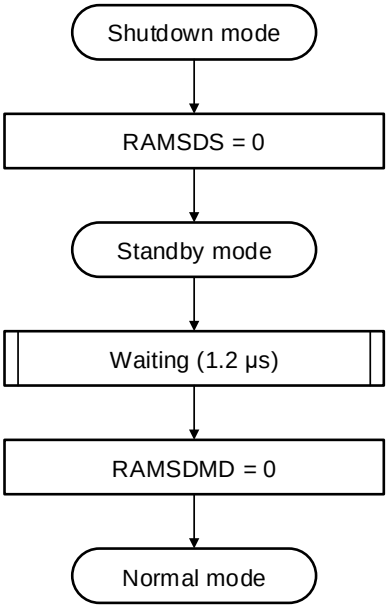
To transition the RAM operation mode from normal mode to shutdown mode, perform the following process.

Figure 24-5. Setting Procedure from Normal Mode to Shutdown Mode



To return the RAM operation mode from the shutdown mode to the normal mode, perform the following process.

Figure 24-6. Setting Procedure from Shutdown Mode to Normal Mode



24.3 Standby Function Operation

24.3.1 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.

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).

The operating statuses in the HALT mode are shown on the next page.

Table 24-2. 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			
		f_{IH}	Operation continues (cannot be stopped)	Operation disabled	Operation disabled	Operation disabled
		f_X	Operation disabled	Operation continues (cannot be stopped)	Cannot operate	Only operation of the PLL clock continues (and cannot be stopped). Clocks other than the PLL clock do not operate.
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)	
		f_{PLL}	Operation disabled	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/040C3H), bit 0 (SELLOSC) of the CKSEL register, and bit 4 (WUTMMCK0) of the OSMC register. • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and SELLOSC = 1: Oscillates • WUTMMCK0 = 0, SELLOSC = 0, and HPIEN = 1: Oscillates • WUTMMCK0 = 0, SELLOSC = 0, and HPIEN = 0: Stops			
f_{WDT}			Set by bits 0 (WDSTBYON) and 4 (WDTON) of user option byte (000C0H/040C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • 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 RDe						
Clock output/buzzer output						
A/D converter						
D/A converter						
Comparator						
AAU						
Security function (AES, TRNG)						
Serial array unit (SAU)						
Serial interface (IICA)						
DTC						
ELC			Linking between operational function blocks is possible.			

(**Remark** is listed on the next page.)

Table 24-2 Operating Statuses in HALT Mode (2/2)

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})
LIN/UART module (RLIN3)		Operable			
CANFD interface (RS-CANFD lite)					
Capacitive sensing unit (CTSU2SLa)					
Power-on-reset function					
Voltage detection function					
External interrupt					
Key interrupt function					
Code Flash ECC function					
CRC operation function	High-speed CRC				
	General-purpose CRC	Operation stopped (operation can continue during DTC transfer)			
Invalid memory access detection function					
Internal RAM-ECC function					
CAN RAM-ECC 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_{XT}: XT1 clock
f_{EX}: External main system clock
f_{EXS}: External subsystem clock
f_{PLL}: PLL clock
f_{WDT}: WDT-dedicated low-speed on-chip oscillator clock

Table 24-3. Operating Statuses in HALT Mode (1/2)

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)	Cannot operate
	f_{IL}		Set by bit 1 (HPIEN) of on-chip debug option byte (000C3H/040C3H), bit 0 (SELLOSC) of the CKSEL register, and bit 4 (WUTMMCK0) of the OSMC register. • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and SELLOSC = 1: Oscillates • WUTMMCK0 = 0, SELLOSC = 0, and HPIEN = 1: Oscillates • WUTMMCK0 = 0, SELLOSC = 0, and HPIEN = 0: Stops		Operation continues (cannot be stopped)
f_{WDT}			Set by bits 0 (WDSTBYON) and 4 (WDTON) of user option byte (000C0H/040C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • 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 RDe					
Clock output/buzzer output					
A/D converter			Operation disabled		
D/A converter					
Comparator					
AAU					
Security function (AES, TRNG)					
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		
CANFD interface (RS-CANFD lite)					
Capacitive sensing unit (CTSU2SLa)					

(**Remark** is listed on the next page.)

Table 24-3. 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})	
Power-on-reset function		Operable		
Voltage detection function				
External interrupt				
Key interrupt function				
CRC operation function	High-speed CRC	Operation disabled		
	General-purpose CRC	Operation stopped (operation can continue during DTC transfer)		
Invalid memory access detection function				
Internal RAM-ECC function				
CAN RAM-ECC function				
Code Flash ECC 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_{XT}: XT1 clock
f_{EX}: External main system clock
f_{EXS}: External subsystem clock
f_{PLL}: PLL clock
f_{WDT}: WDT-dedicated low-speed on-chip oscillator clock

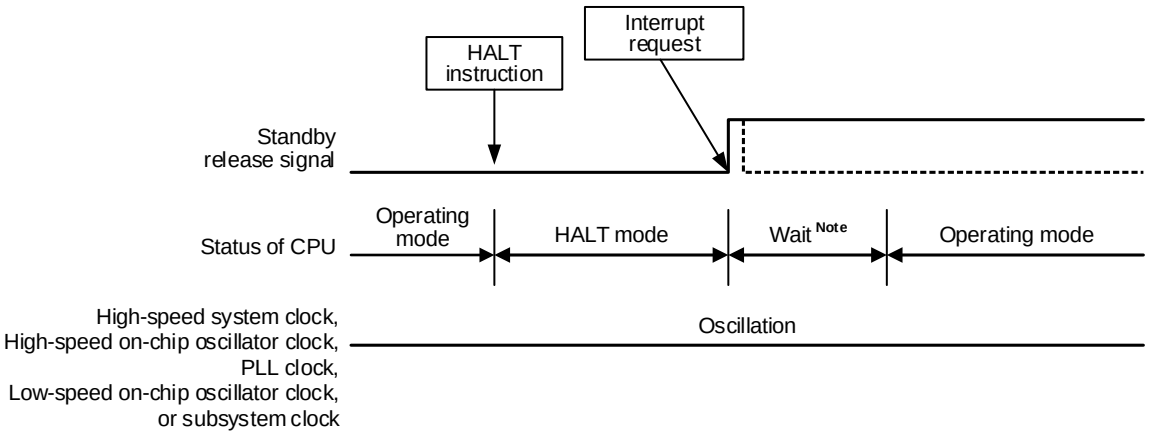
(2) HALT mode release

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 instruction at the next address after the HALT instruction is executed.

Figure 24-7. 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

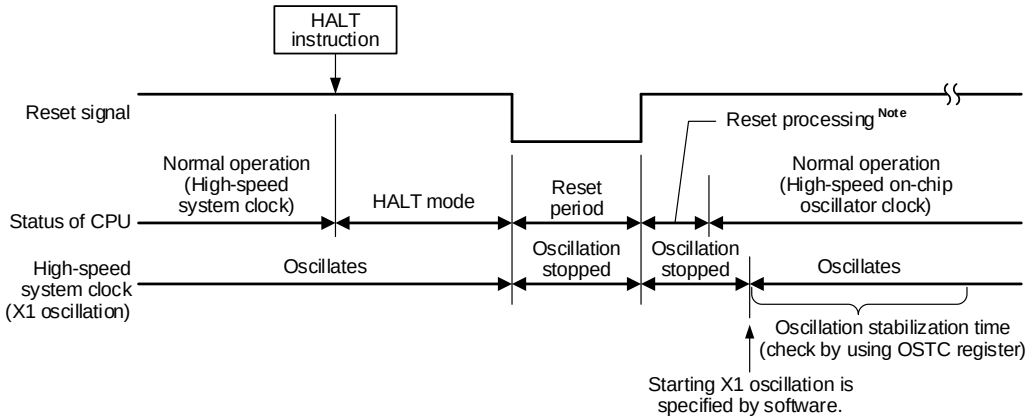
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

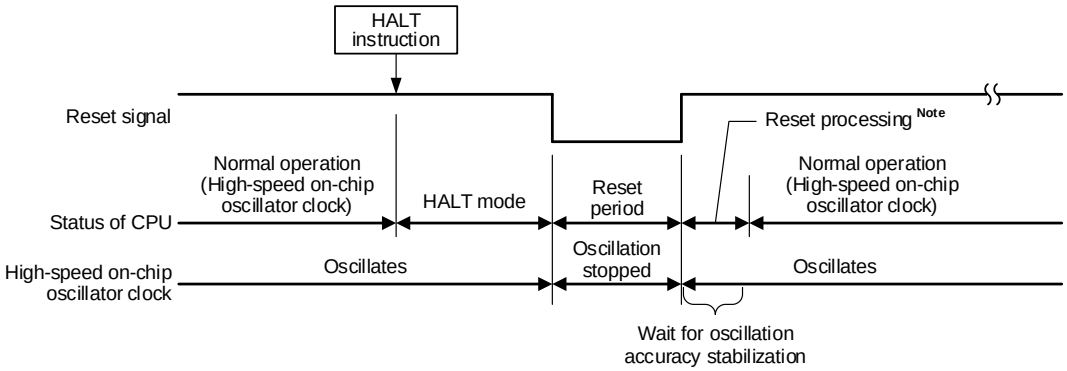
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program branches.

Figure 24-8. HALT Mode Release by Reset

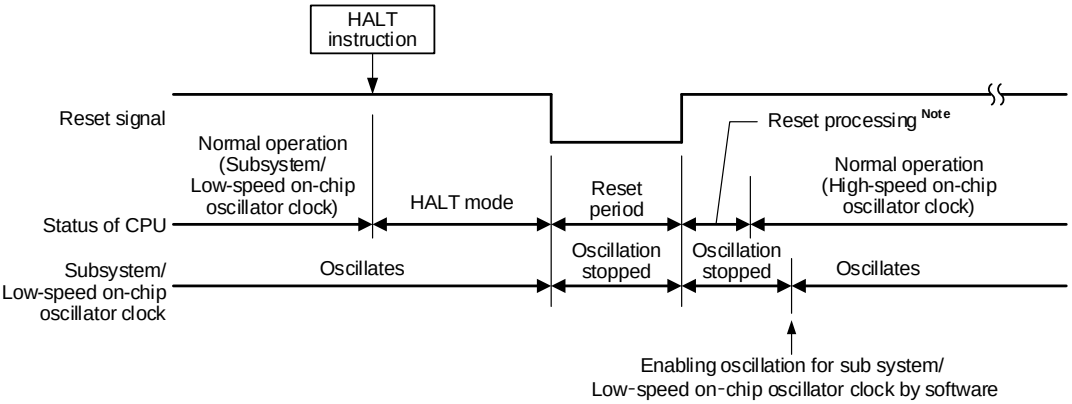
(1) When High-speed System Clock is Used as CPU Clock



(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 26 POWER-ON-RESET CIRCUIT**.

24.3.2 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.

Cautions 1. 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.

2. To enter STOP mode, stop PLL operation by the PLLON bit (bit 0 of the PLLCTL register) and then execute the STOP instruction.

The operating statuses in the STOP mode are shown on the next page.

Table 24-4. Operating Statuses in STOP Mode (1/2)

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			
Main system clock: f_X	Stopped			
Main system clock: f_{EX}	Stopped			
Main system clock: f_{PLL}	Operation disabled			
Subsystem clock: f_{XT}	Status before STOP mode was set is retained			
Subsystem clock: f_{EXS}	Status before STOP mode was set is retained			
f_{IL}	Set by bit 0 (SELLOSC) of the CKSEL register and bit 4 (WUTMMCK0) of the OSMC register. • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and SELLOSC = 1: Oscillates • WUTMMCK0 = 0 and SELLOSC = 0: Stops			
f_{WDT}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of user option byte (000C0H/040C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • WDTON = 1 and WDSTBYON = 0: Stops			
CPU	Operation stopped			
Code flash memory	Operation stopped			
Data flash memory	Operation stopped			
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 • In the event counting mode when no TRJIO0 input filters are selected. • 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. • If the low-speed on-chip oscillator is selected as the clock source for counting.			
Timer RDe	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	Operation disabled			
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)			
AAU	Operation disabled			
Security function (AES, TRNG)	Operation disabled			
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.			

(**Remark** and **Cautions** are listed on the next page.)

Table 24-4. Operating Statuses in STOP Mode (2/2)

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})
LIN/UART module (RLIN3)		Only wakeup operation of the UART is possible (switching to the SNOOZE mode).			
CANFD interface (RS-CANFD lite)		Operation disabled			
Capacitive sensing unit (CTSU2SLa)		SNOOZE operation request from the capacitive sensing unit (CTSU2SLa) is possible (switching to the SNOOZE mode)			
Power-on-reset function		Operable			
Voltage detection function					
External interrupt					
Key interrupt function					
CRC operation function	High-speed CRC	Operation stopped			
	General-purpose CRC				
Invalid memory access detection function					
Internal RAM-ECC function					
CAN RAM-ECC function					
Code Flash ECC 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_{XT}: XT1 clock
f_{EX}: External main system 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/040C0H) 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

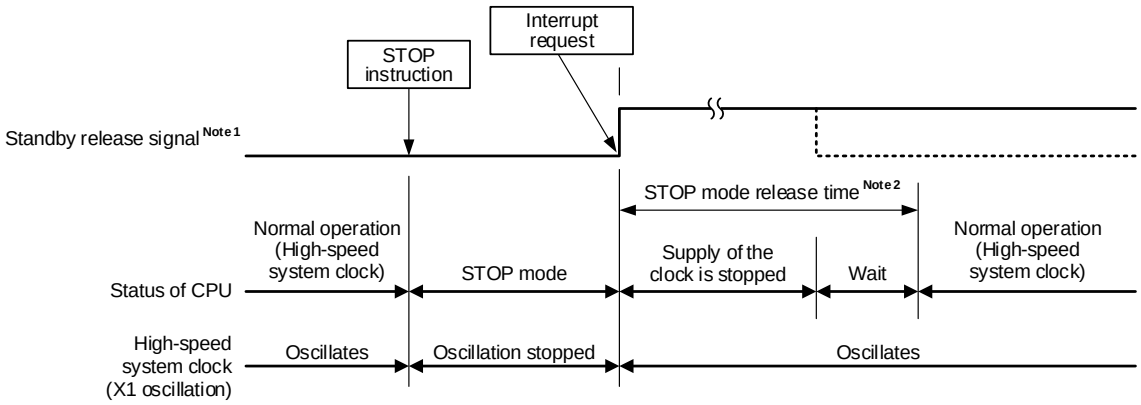
The STOP mode can be released by interrupt and reset signal generation.

(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 instruction at the next address after the STOP instruction is executed.

Figure 24-9. 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 22-1. Basic Configuration of Interrupt Function**.

2. STOP mode release time

Supply of the clock is stopped

- Oscillation stabilization time (set by OSTS register)

Wait

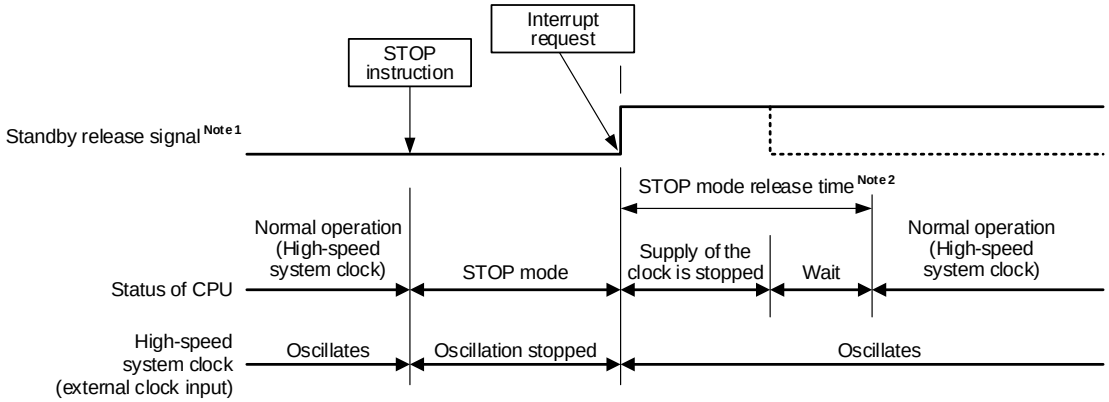
- When vectored interrupt servicing is carried out: 10 to 11 clocks
- When vectored interrupt servicing is not carried out: 4 to 5 clocks

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 24-9. STOP Mode Release by Interrupt Request Generation (2/2)

(2) When High-speed System Clock (External Clock Input) is Used as CPU Clock



Notes 1. For details of the standby release signal, see **Figure 22-1. Basic Configuration of Interrupt Function**.

2. STOP mode release time

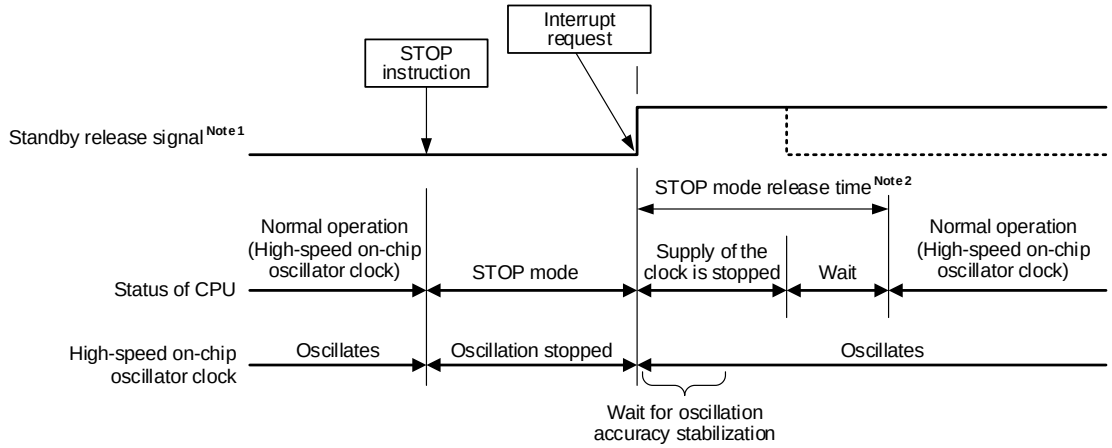
Supply of the clock is stopped

- External clock: 50 to 51 clocks

Wait

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 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 22-1. Basic Configuration of Interrupt Function**.

2. STOP mode release time

Supply of the clock is stopped

- 10 µs to 40 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.

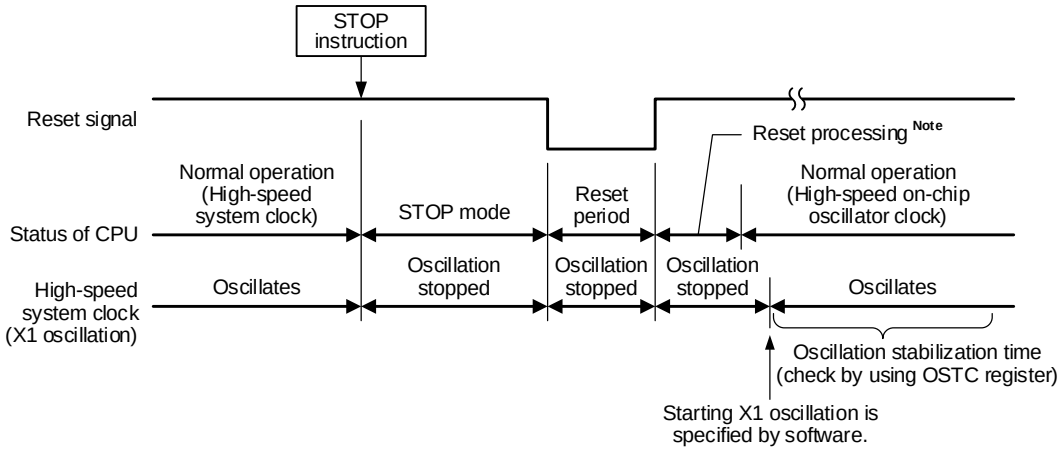
2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

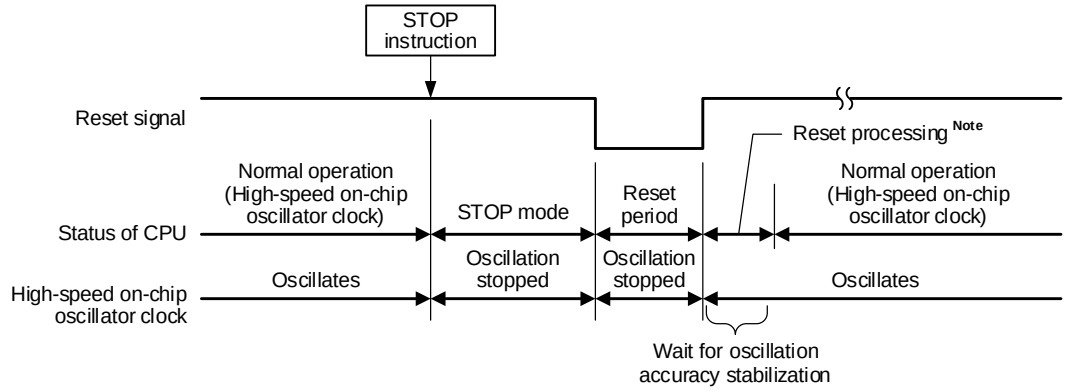
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 24-10. 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 26 POWER-ON-RESET CIRCUIT**.

24.3.3 SNOOZE Mode

(1) SNOOZE mode setting and operating statuses

SNOOZE mode can be set in the following peripheral functions.

See each chapter for details.

- **CHAPTER 15 CAPACITIVE SENSING UNIT (CTSU2SLa)**
- **CHAPTER 18 LIN/UART MODULE (RLIN3)**
- **CHAPTER 20 DTC**

SNOOZE mode can only be specified when the CPU clock is the high-speed on-chip oscillator clock.

In SNOOZE mode transition, wait status to be only following time.

Transition time from STOP mode to SNOOZE mode

- 10 μs to 40 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: (0.3 μs to 6.0 μs) + 7 clocks
- When vectored interrupt servicing is not carried out: (0.3 μs to 6.0 μs) + 1 clock

The operating statuses in the SNOOZE mode are shown on the next page.

Table 24-5. Operating Statuses in SNOOZE Mode (1/2)

STOP Mode Setting / Item			During a period in STOP mode, reception of a data signal for the LIN/UART module in the UART mode, the generation of an interrupt signal for DTC activation, or capacitive sensing unit in operation.
			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. • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and SELLOSC = 1: Oscillates • WUTMMCK0 = 0 and SELLOSC = 0: Stops
f_{WDT}			Set by bits 0 (WDSTBYON) and 4 (WDTON) of user option byte (000C0H/040C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • 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 • In the event counting mode when no TRJIO0 input filters are selected. • 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. • If the low-speed on-chip oscillator is selected as the clock source for counting.
Timer RDe			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			Operation disabled
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)
AAU			Operation stopped
Security function (AES, TRNG)			Operation stopped
Serial array unit (SAU)			Operation disabled
Serial interface (IICA)			
DTC			Operable
ELC			Linking between operation function blocks is possible.

(**Remark** is listed on the next page.)

Table 24-5. Operating Statuses in SNOOZE Mode (2/2)

STOP Mode Setting / Item		During a period in STOP mode, reception of a data signal for the LIN/UART module in the UART mode, the generation of an interrupt signal for DTC activation, or capacitive sensing unit in operation.
		When CPU Is Operating on High-speed On-chip Oscillator Clock (f_{IH})
LIN/UART module (RLIN3)		Operable (only in the UART mode)
CANFD interface (RS-CANFD lite)		Operation disabled
Capacitive sensing unit (CTSU2SLa)		Operable
Power-on-reset function		
Voltage detection function		
External interrupt		
Key interrupt function		
CRC operation function	High-speed CRC	Operation stopped
	General-purpose CRC	
Invalid memory access detection function		
Internal RAM-ECC function		
CAN RAM-ECC function		
Code Flash ECC 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_{XT}: XT1 clock
f_{EX}: External main system 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

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 RDe0, and ports simultaneously.

The below is an example to output SNOOZE mode status synthesized with A/D conversion.

- f_{SL} is used as the count source for the timer RDe0.
- By using the PWM function of the timer mode, the timer RDe0 sets a period in the TRDGRA0 register.
- The timer RDe0 also 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 STOP mode release and start of the CPU operation.
- In the CPU operation after the STOP mode release, A/D conversion starts by software trigger from CPU.
- 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.

Figure 24-11 shows the configuration of the circuit for SNOOZE mode status output. **Figure 24-12** shows the timing example of SNOOZE mode status output synthesized with A/D conversion.

Figure 24-11. Example of Configuration of Circuit for Combination of SNOOZE Mode Status Output and A/D Converter

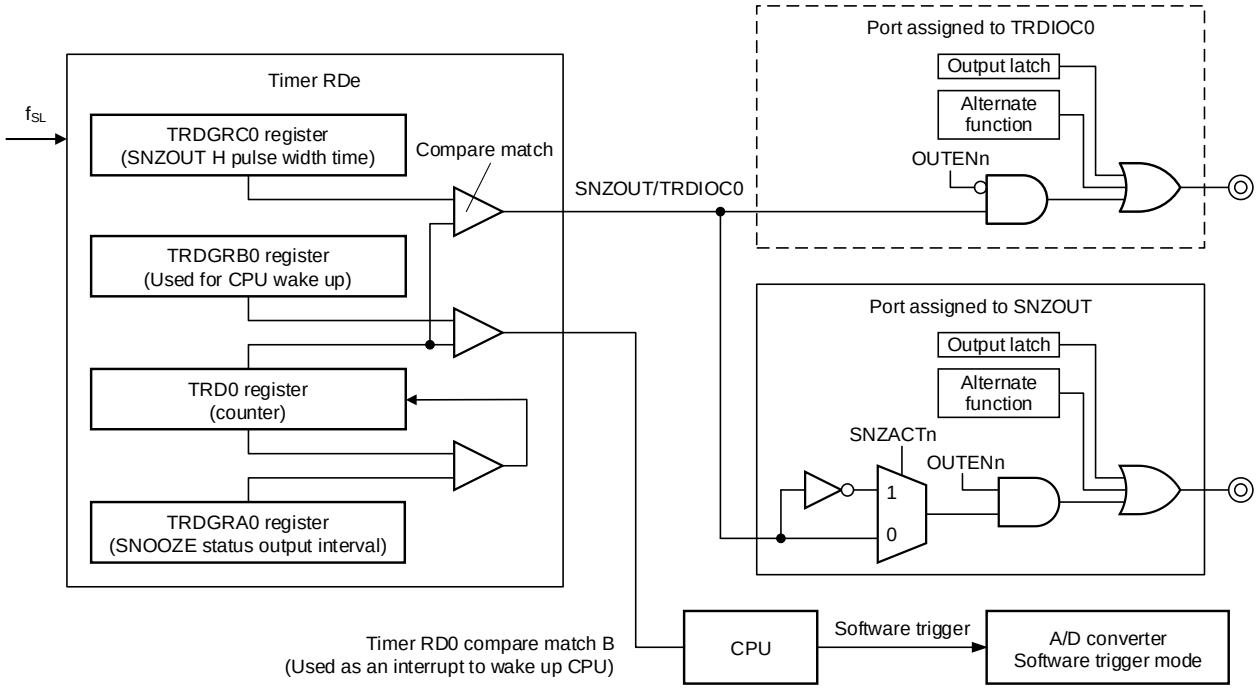


Figure 24-12. Timing Example of SNOOZE Mode Status Output and A/D Conversion

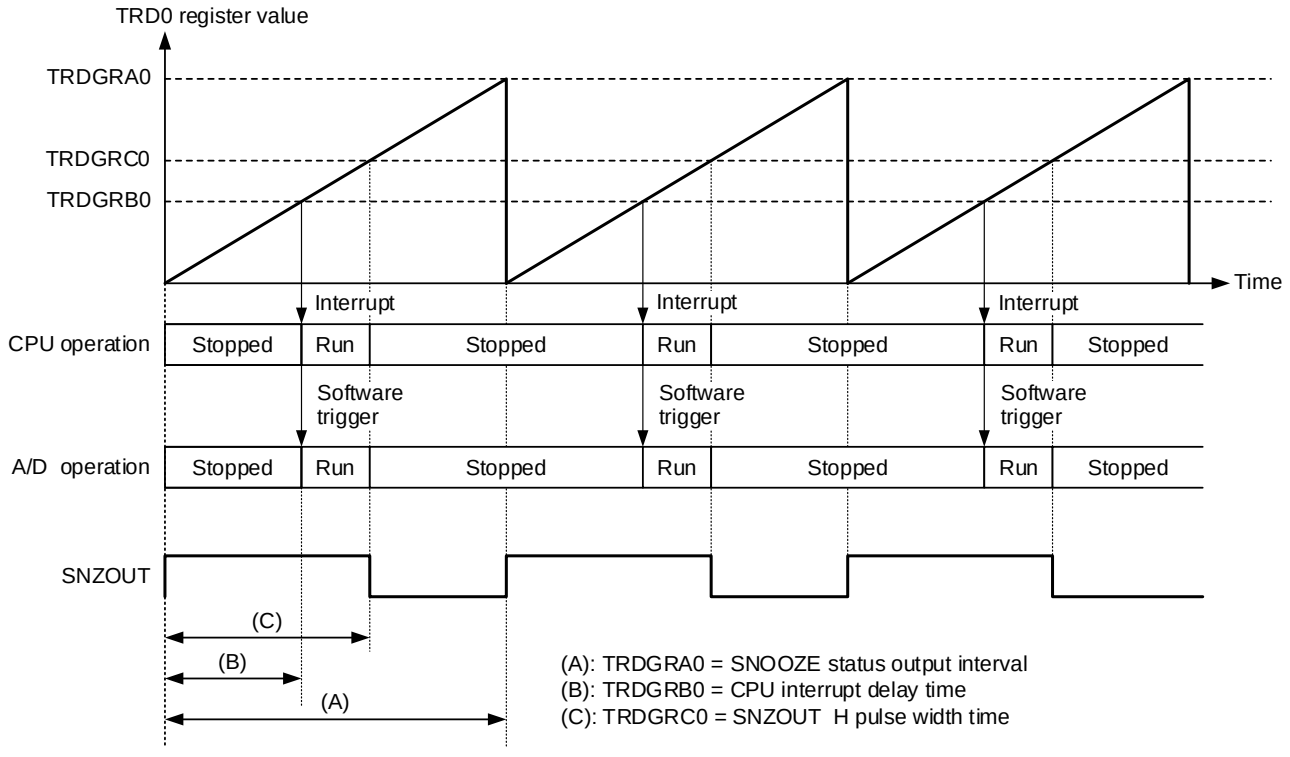
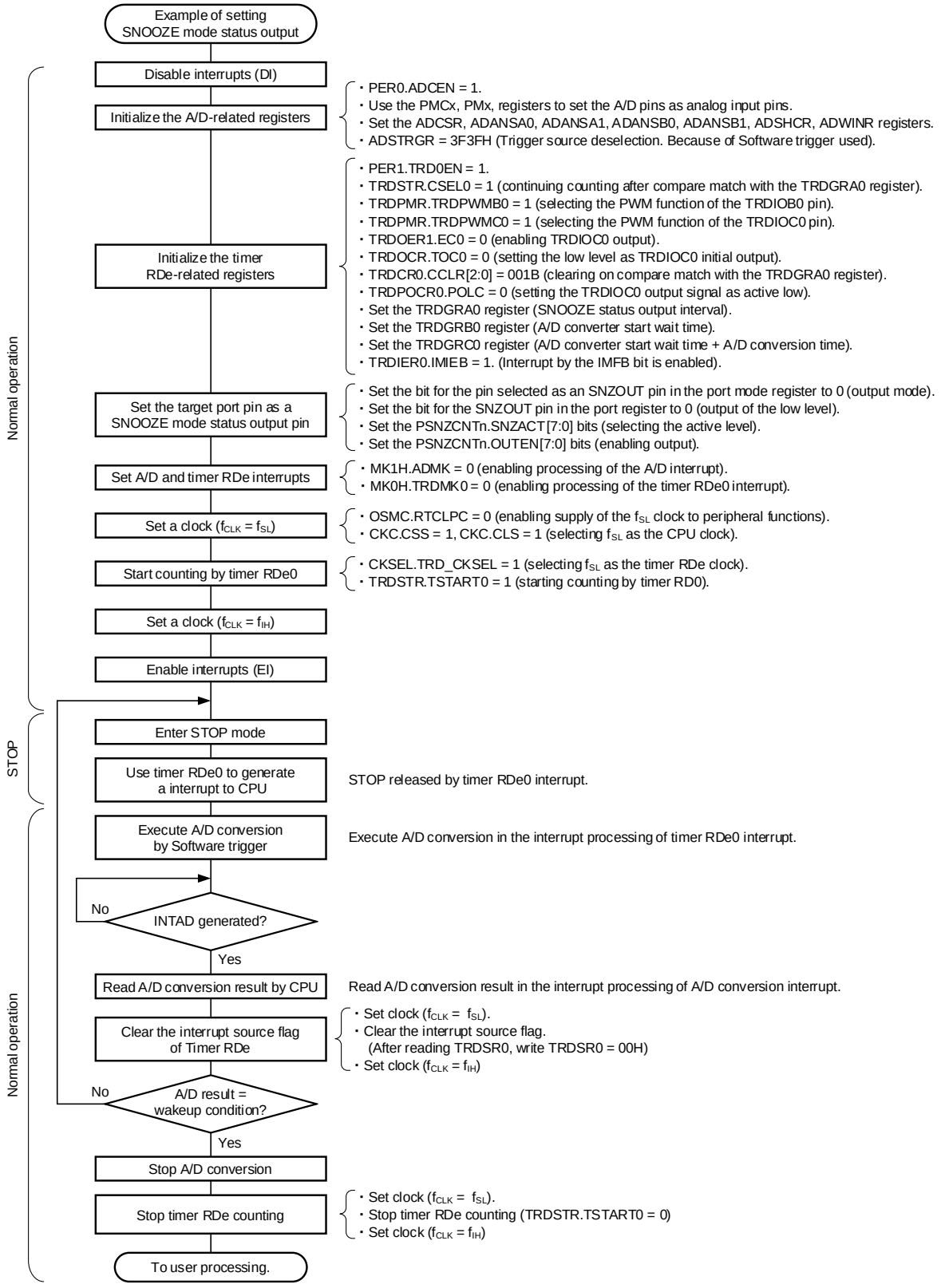


Figure 24-13. Example of Settings for SNOOZE Mode Status Output



Caution For details on the settings of registers for the A/D converter, timer RDe, port functions, etc., see the corresponding chapters.

CHAPTER 25 RESET FUNCTION

The following seven operations are available to generate a reset signal.

(1) External reset input via 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 (LVD0) 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 RESET pin, the watchdog timer detects an overflow, a voltage is detected on the POR and LVD0 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 25-1**.

When a low level is input to the RESET pin, the device is reset. It is released from the reset status when a high level is input to the 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 25-2** to **25-4**) after reset processing.

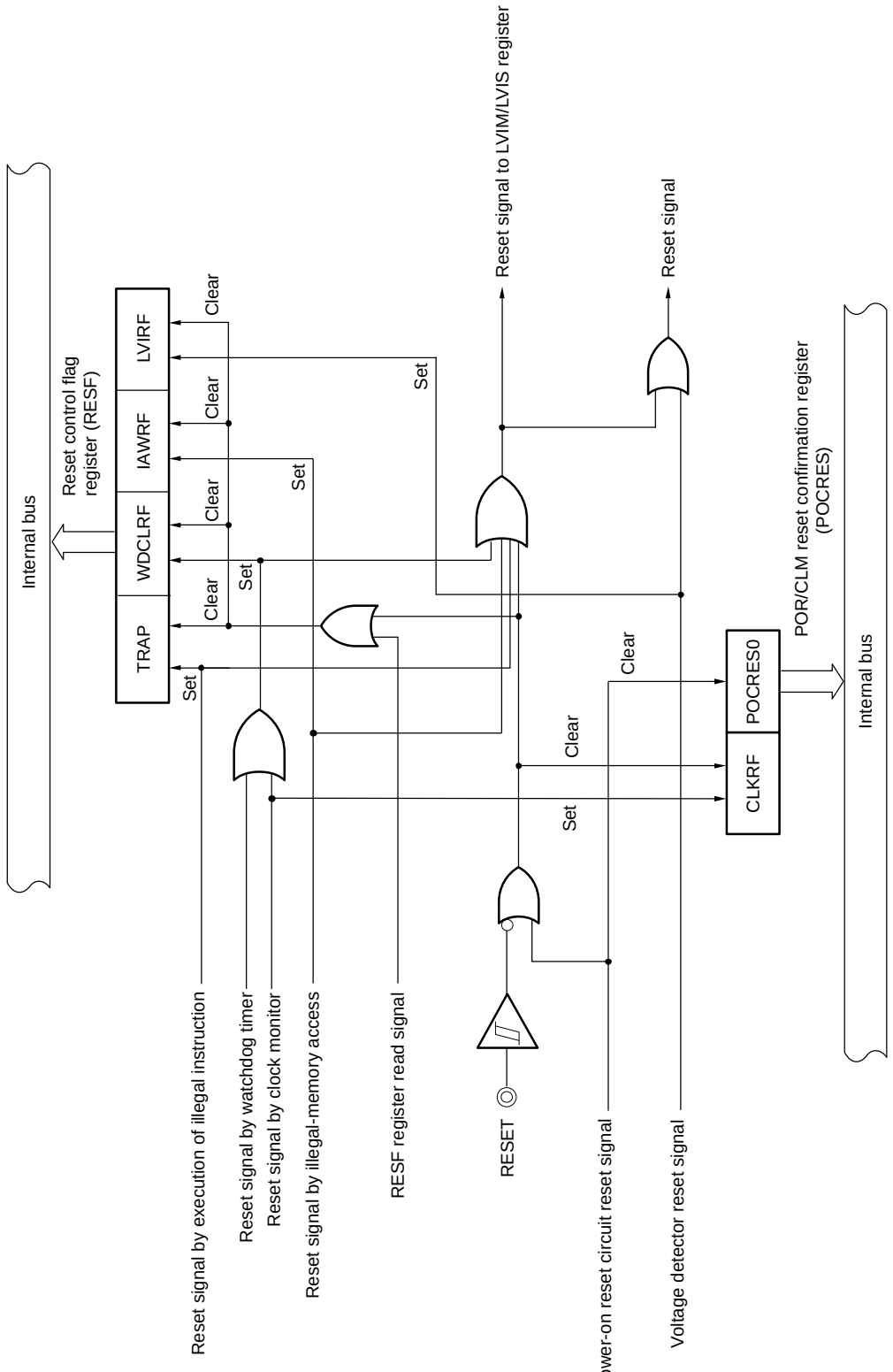
Reset by POR and LVD0 circuit supply voltage detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD0}$ is detected after the reset, and program execution starts using the high-speed on-chip oscillator clock (see **CHAPTER 26 POWER-ON-RESET CIRCUIT** and **CHAPTER 27 VOLTAGE 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 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 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.
Also, refer to CHAPTER 37 and 38 ELECTRICAL SPECIFICATIONS for the operating voltage range.
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. The port pins become the following state because each SFR and 2nd SFR are initialized after reset.
- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).**
- **P130: Low level during the reset period or after receiving a reset signal.**
- **Ports other than P40 and P130: High-impedance during the reset period or after receiving a reset signal.**

Remarks V_{POR}: POR power supply rise detection voltage, V_{LVD0}: Reset detection voltage for LVD0 circuit

Figure 25-1. Block Diagram of Reset Function



Caution An LVD0 circuit internal reset does not reset the LVD0 circuit.

Remarks 1. LVIM: Voltage detection register.

2. LVIS: Voltage detection level register.

Figure 25-2. Timing of Reset by RESET Input

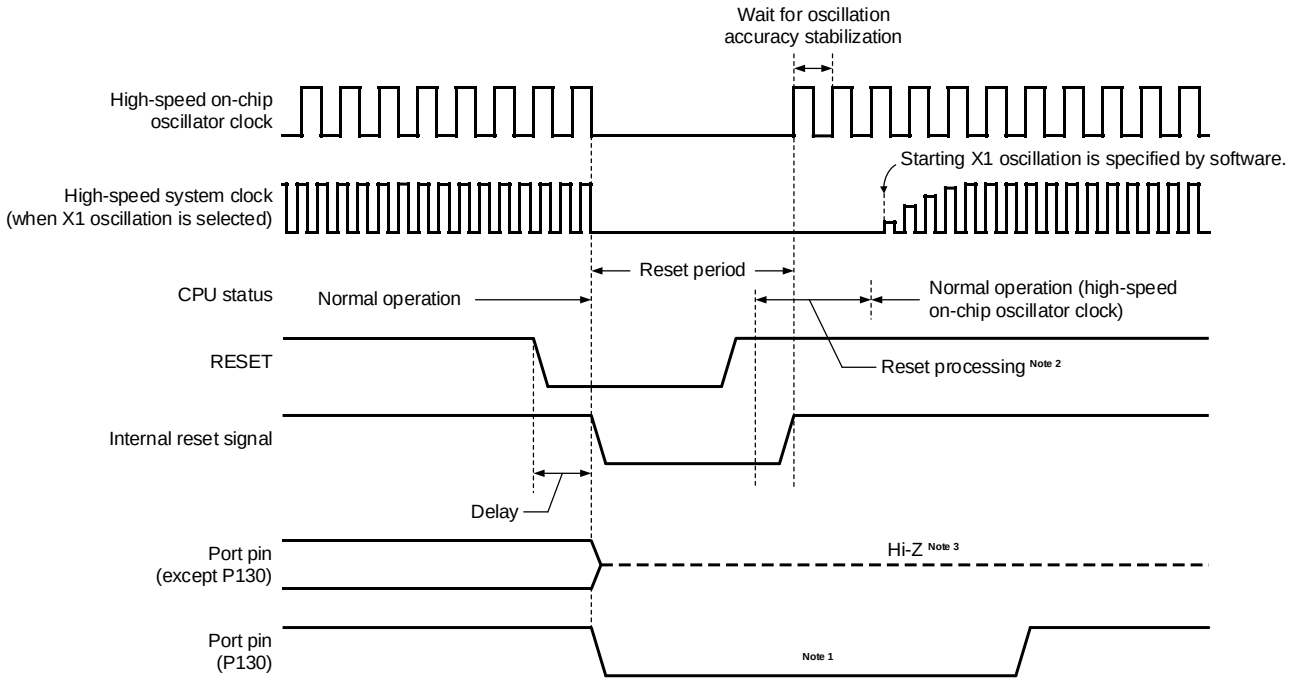
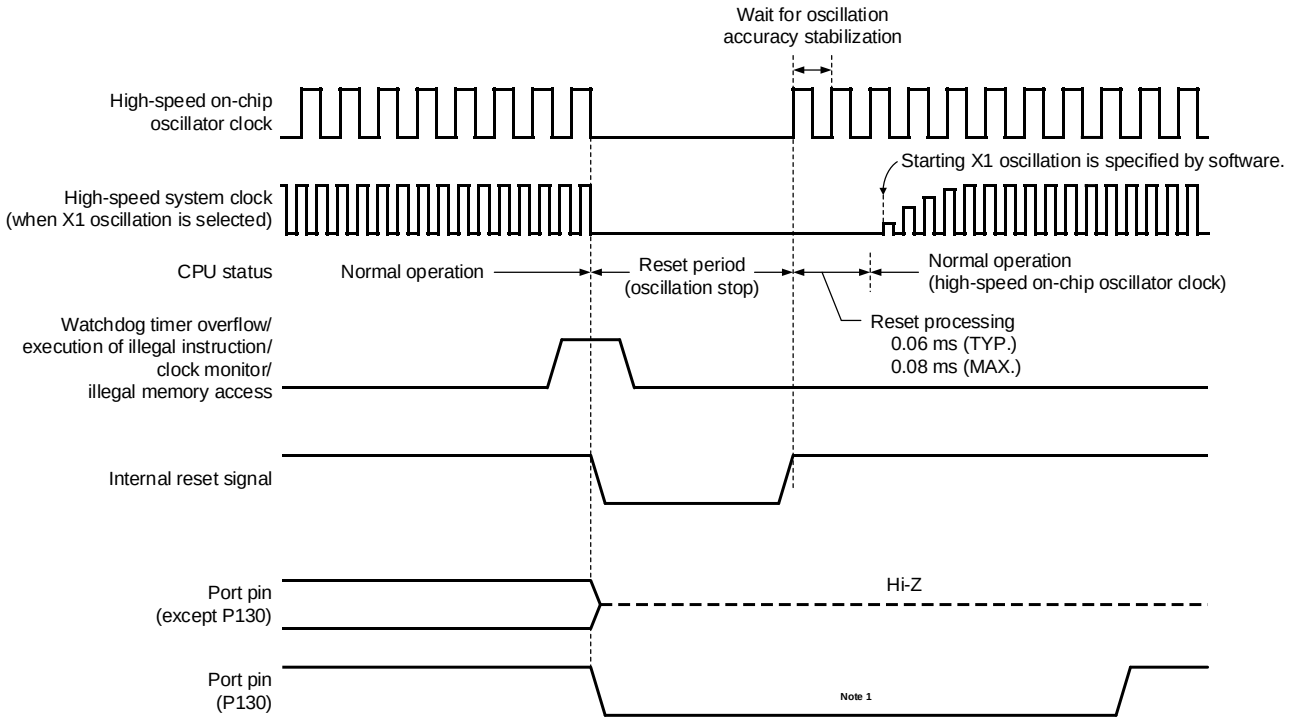
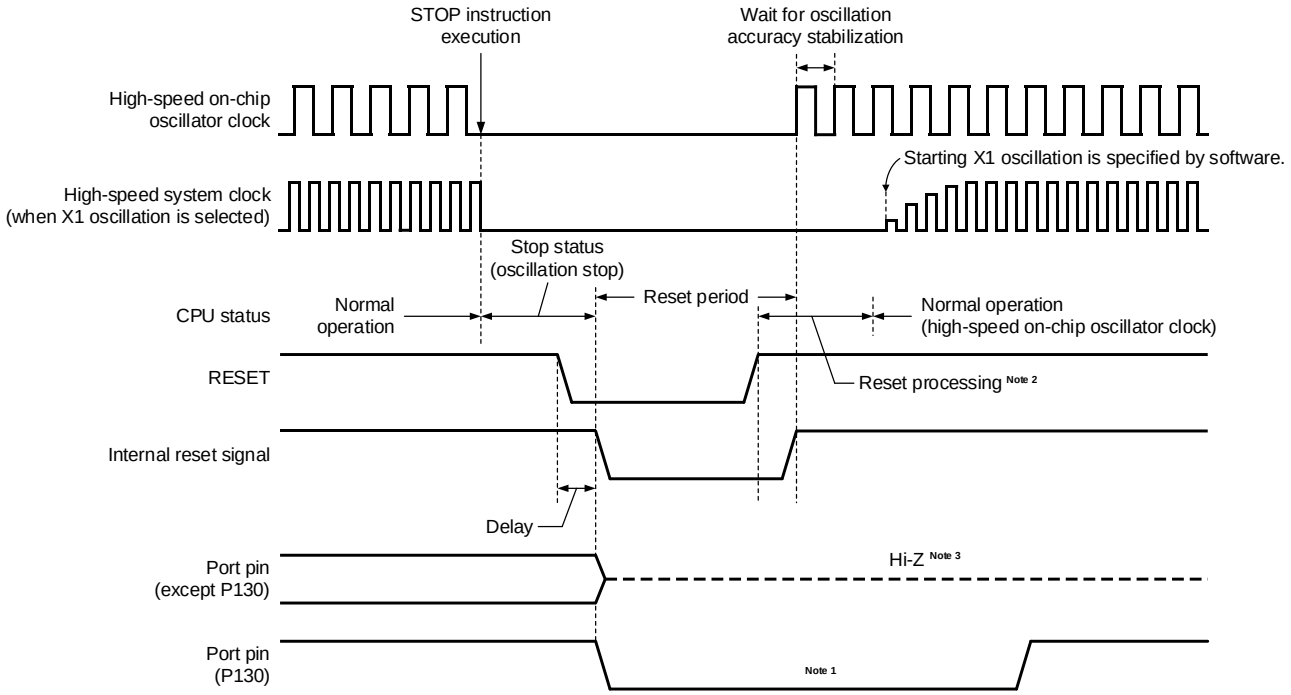


Figure 25-3. Timing of Reset by Watchdog Timer Overflow, Execution of Illegal Instruction, Clock Monitor, or Illegal-Memory Access



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

Figure 25-4. Timing of Reset in STOP Mode by 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: 0.674 ms (TYP.), 0.871 ms (MAX.) (when the LVD is in use)
0.401 ms (TYP.), 0.557 ms (MAX.) (when the LVD is off)
After the second release of POR: 0.618 ms (TYP.), 0.790 ms (MAX.) (when the LVD is in use)
0.345 ms (TYP.), 0.476 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.

3. The state of P40 is as follows.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor)

Remark For the reset timing of the power-on-reset circuit and voltage detector, see **CHAPTER 26 POWER-ON-RESET CIRCUIT** and **CHAPTER 27 VOLTAGE DETECTOR**.

Table 25-1. Operation Statuses During Reset Period (1/2)

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) 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 RDe			
Real-time clock (RTC)			
Watchdog timer			
Clock monitor			
Clock output/buzzer output			
A/D converter			
D/A converter **Note**			
Comparator **Note**			
AAU			
Serial array unit (SAU)			
Serial interface (IICA)			
LIN/UART module (RLIN3)			
CANFD interface (RS-CANFD lite) **Note**			
Capacitive sensing unit (CTSU2SLa)			
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			
Code Flash memory ECC function			

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

Table 25-1. Operation Statuses During Reset Period (2/2)

Item		During Reset Period
CRC operation function	High-speed CRC	Operation stopped
	General-purpose CRC	
Invalid memory access detection function		
Internal RAM-ECC function		
CAN RAM-ECC function		
RAM guard function		
SFR guard function		

Note RL78/F25 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 25-2. States of Hardware After Acceptance of a Reset

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
Reset control flag register (RESF)		Undefined [Note 2]
Voltage detection register (LVIM)		00H
Voltage detection level register (LVIS)		19H
Watchdog timer enable register (WDTE)		1AH/9AH [Note 3]
Special Function Register (SFR)		See related registers in this manual.
2nd Special Function Register (2nd SFR)		See related registers in this manual.

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 values depend on the source of the reset as shown in **Table 25-3**.

3. The reset value of WDTE is determined by the option byte setting.

Table 25-3. States of Bits in RESF/POCRES Registers When Reset Requests are Generated

Register \ Reset Source		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 LVD0
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

25.1 Register for Confirming Reset Source

The following registers are used to control the reset function.

Table 25-4. Reset Function Control Registers

Address	Register Name	Symbol	After Reset	Access Size
FFFA8H	Reset control flag register	RESF	00H Note 1	8
F02CCH	POR/CLM reset confirmation register	POCRES	00H Note 2	1, 8

Notes 1. The value after reset varies depending on the reset source.

2. When the reset is from a source other than the POR circuit, the value of the POCRES0 bit immediately before the reset is retained.

25.1.1 Reset Control Flag Register (RESF)

Many internal reset generation sources exist in the RL78/F22 and RL78/F25. 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.

RESET input, reset by power-on-reset (POR) circuit, and reading the RESF register clear TRAP, WDCLRF, IAWRF, and LVIRF flags.

Figure 25-5. 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 (LVD0)
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 25-5**.

Table 25-5. States of Bits in the RESF Register When Reset Requests are Generated

Reset Source / Flag	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 LVD0
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)

25.1.2 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 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 25-6. Format of POR Reset Register (POCRES)

Address: F02CCH 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.

CLKRF	Internal reset request by clock monitor (RESCLM)
0	Internal reset request (RESCLM) is not generated, or the CLKRF bit is cleared.
1	Internal reset request (RESCLM) was generated.

Note If the reset is from a source other than the POR circuit, the values of the POCRES0 bit immediately before the reset are retained.

Table 25-6 shows the states of bits in the POCRES register when reset requests are generated.

Table 25-6. States of Bits in the POCRES Register When Reset Requests are Generated

Reset Source / Flag	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 LVD0
POCRES0	Held	Cleared (0)	Held	Held	Held	Held	Held	Held
CLKRF	Cleared (0)	Cleared (0)	Held	Held	Held	Set (1)	Held	Held

CHAPTER 26 POWER-ON-RESET CIRCUIT

26.1 Functions of Power-on-reset Circuit

The power-on-reset circuit (POR) has the following functions.

- The reset signal is released when the supply voltage (V_{DD}) exceeds V_{POR}.
  However, the reset state should be maintained by the voltage detection circuit (LVD0) or external reset up to the operating voltage range that satisfies the electrical characteristics.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR}), generates internal reset signal when $V_{DD} < V_{PDR}$.
  However, at the falling edge of the power supply, before the voltage falls below the operating voltage range, shift to the STOP mode or reset by the voltage detection circuit or external reset.

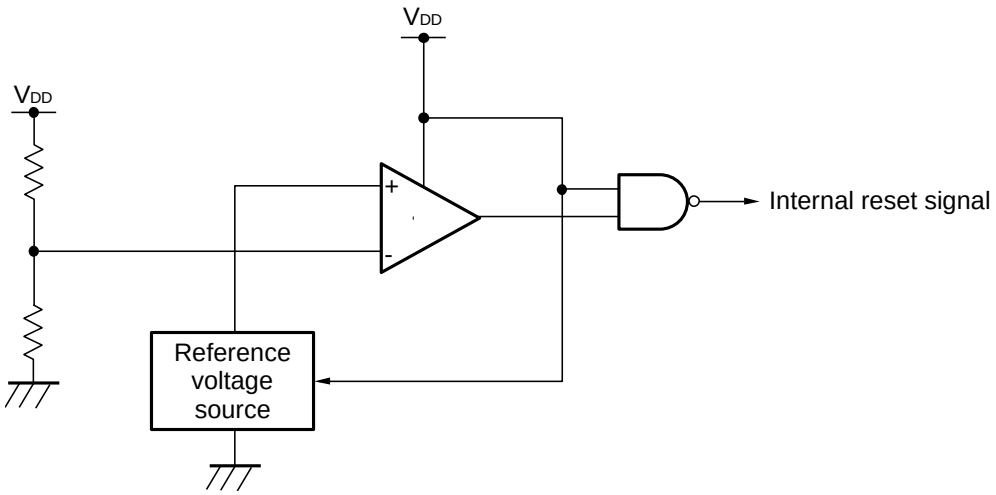
Caution **If an internal reset signal is generated in the POR circuit, the POCRES0 and CLKRF flags of the POR/CLM reset confirmation register (POCRES) and the TRAP, WDCLRF, IAWRF, and LVIRF flags of the reset control flag register (RESF) are cleared (00H).**

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 (00H) and the flag is set (1) when an internal reset signal is generated due to WDT/LVD/illegal instruction execution/clock monitoring/illegal-memory access. The CLKRF bit of the POCRES register is set to 1 when an internal reset signal is generated by the clock monitor. The POCRES0 bit of the POCRES register is cleared to 0 by the POR reset when it has been set to 1 beforehand.
For details of the POCRES and RESF registers, see **CHAPTER 25 RESET FUNCTION**.

26.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in **Figure 26-1**.

Figure 26-1. Block Diagram of Power-on-reset Circuit



26.3 Operation of Power-on-reset Circuit

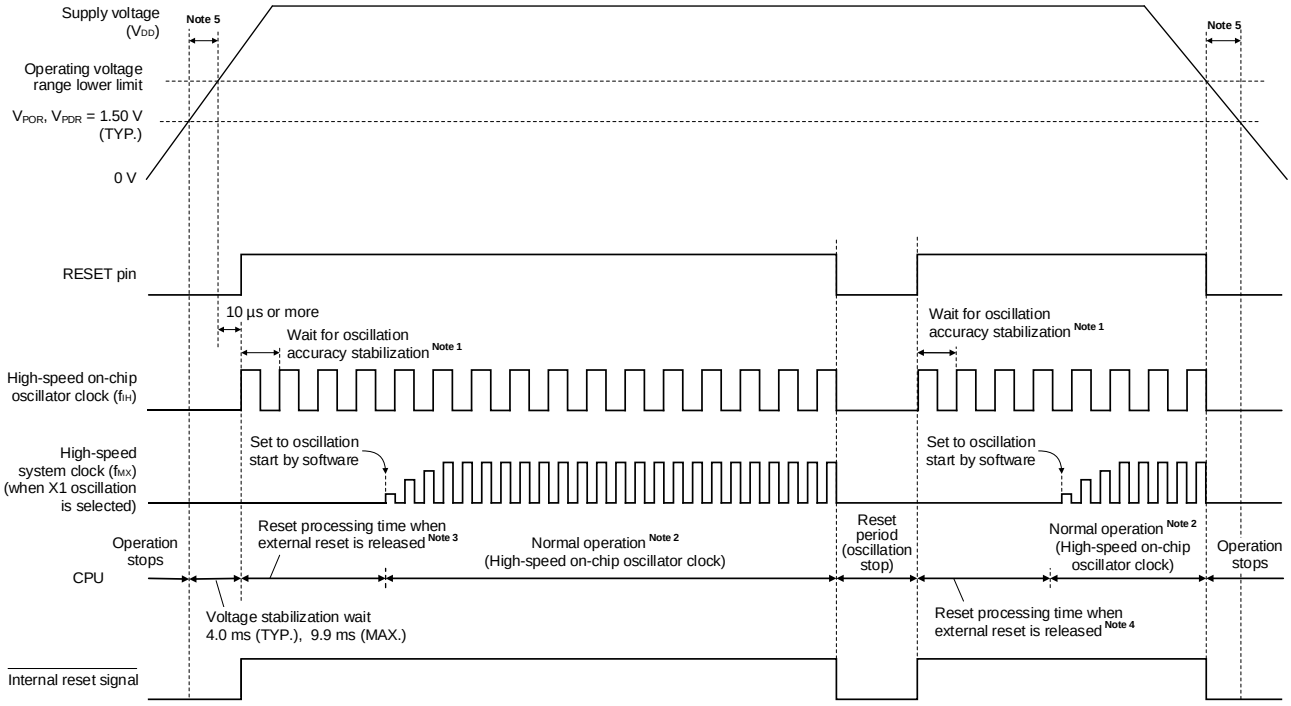
- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{POR} **Note**), the reset status is released.
- The supply voltage (V_{DD}) and detection voltage (V_{PDR} **Note**) are compared. When $V_{DD} < V_{PDR}$, the internal reset 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 26-2. 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 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 (release from the first external reset following release from the POR state) after the RESET signal is driven high (1) as well as the voltage stabilization wait time after V_{POR} (1.50 V, TYP.) is reached.
Reset processing time when the external reset is released is shown below.
Release from the first external reset following release from the POR state:
0.674 ms (TYP.), 0.871 ms (MAX.) (when the LVD is in use)
0.401 ms (TYP.), 0.557 ms (MAX.) (when the LVD is off)

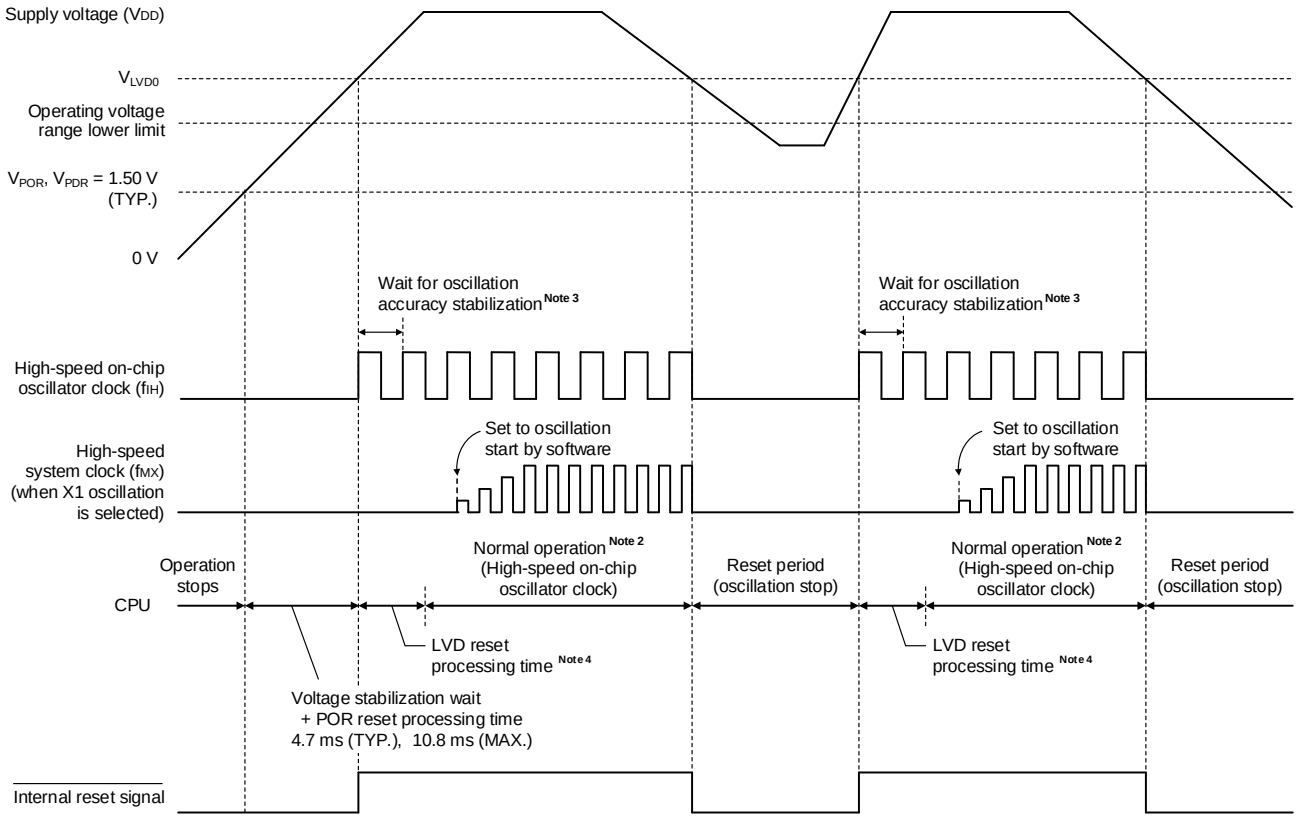
4. Reset times in cases of release from an external reset other than the above are listed below.
Release from the reset state for external resets other than the above case:
0.618 ms (TYP.), 0.790 ms (MAX.) (when the LVD is in use)
0.345 ms (TYP.), 0.476 ms (MAX.) (when the LVD is off)

5. At the rising edge of the power supply, maintain the reset state by the voltage detection circuit (LVD0) or external reset up to the operating voltage range that satisfies the electrical characteristics. At the falling edge of the power supply, before the power supply voltage falls below the operating voltage range, shift to the STOP mode, or reset by the voltage detection circuit or external reset. When restarting the operation, confirm that the power supply voltage has returned to the operating voltage range.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 26-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/2)

(2) When LVD0 reset mode (option byte 000C1/040C1H: LVD0ENB = 0)



Notes 1. The guaranteed range for operation is 2.7 V ≤ V_{DD} ≤ 5.5 V (RL78/F25 products) or 1.8 V ≤ V_{DD} ≤ 5.5 V (RL78/F22 and Grade 3 products only) .
Only proceed with normal operations after V_{DD} has reached or exceeded lower limit of guaranteed operation voltage. If an operation may be generated at lower than lower limit of guaranteed operation voltage when the supply voltage falls or power-on, use the reset function of the voltage detector, or input the low level to the 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. When the power supply voltage drops, only an internal reset by the voltage detector (LVD0) occurs, and then the power supply voltage recovers again, the following “LVD reset processing time” is required after reaching the LVD0 detection level (VLVD0).

LVD reset processing time: 0 to 0.12 ms (MAX.)

Remark V_{LVD0}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

26.4 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.

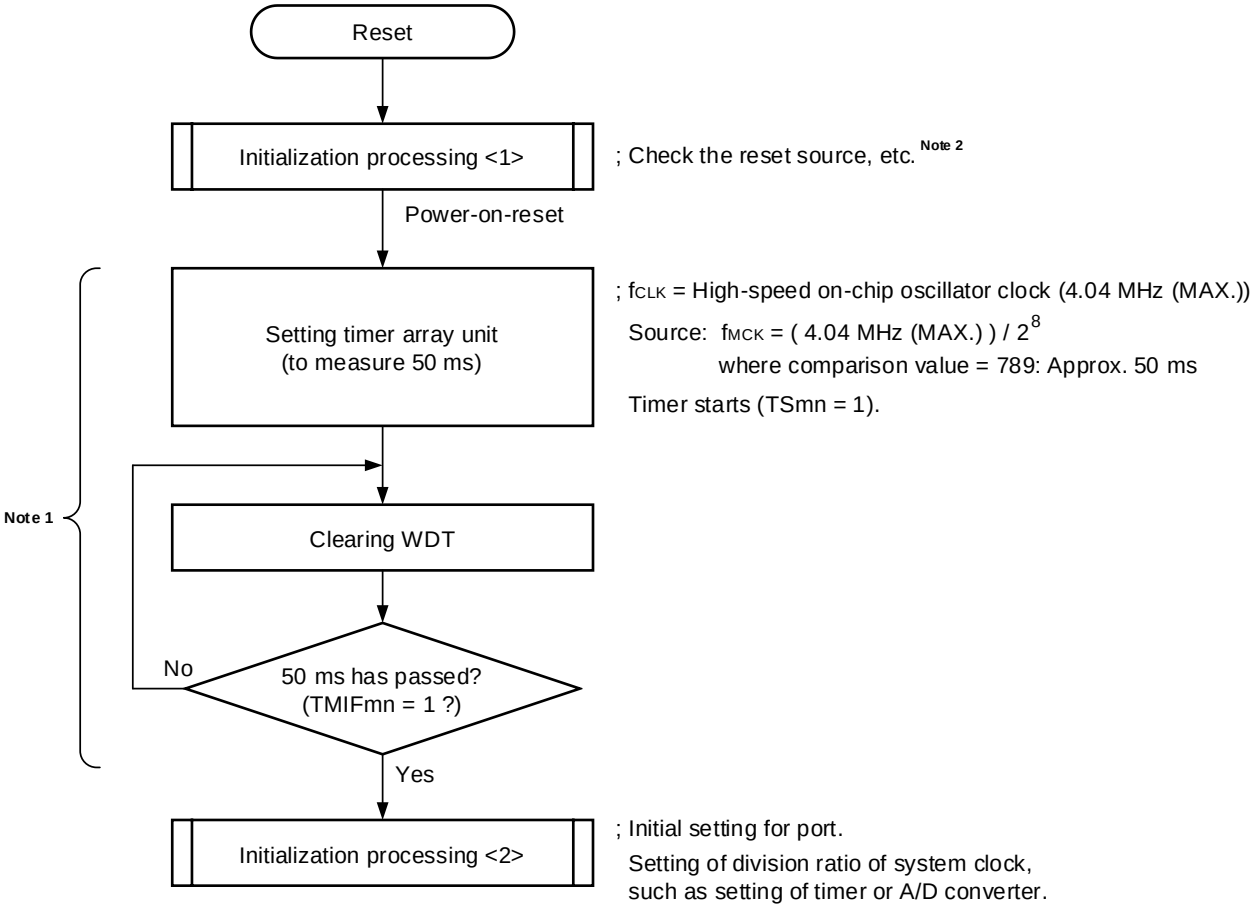
For circuits that do not use a reset circuit, enable voltage detector (LVD circuit).

<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 26-3. 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



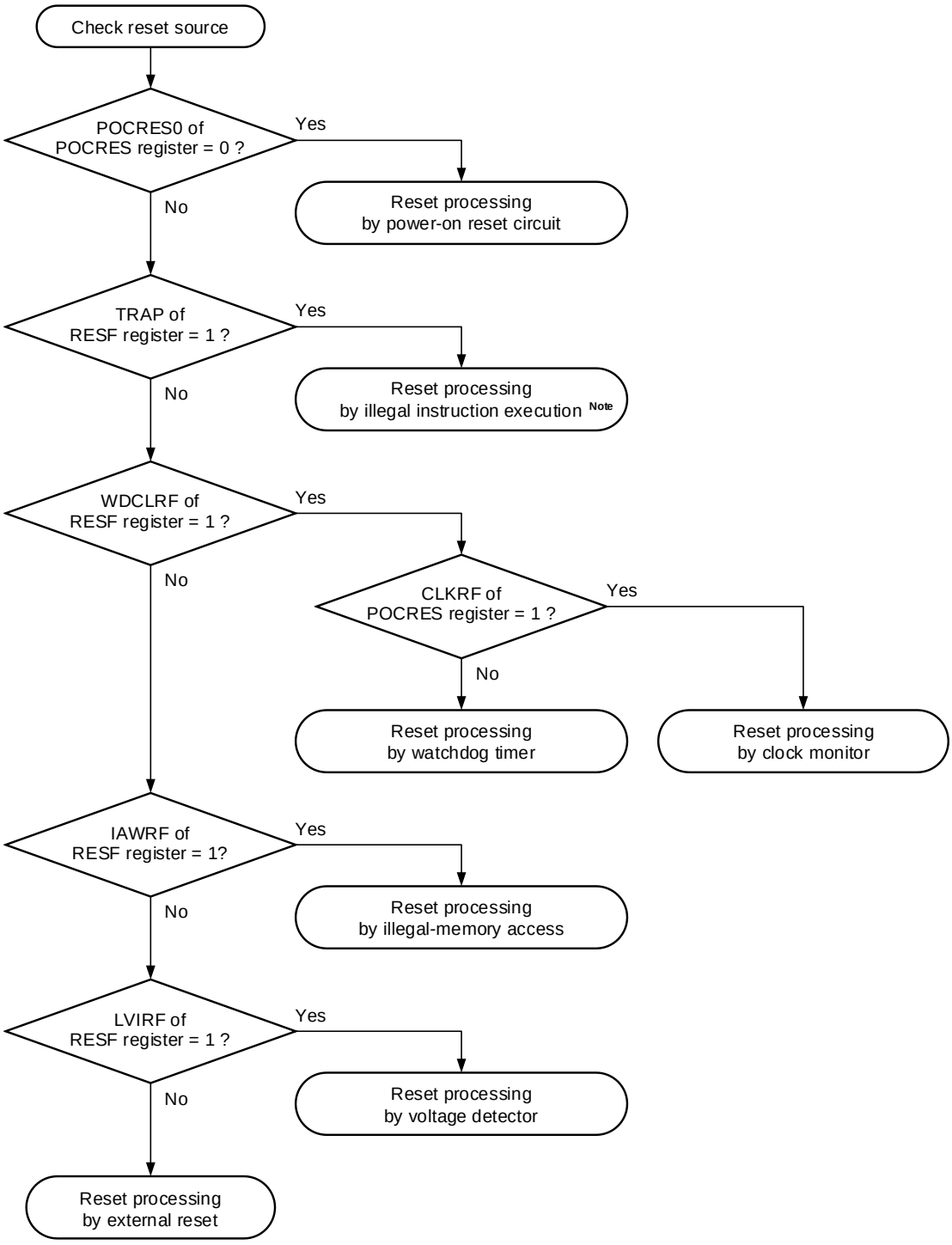
Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Remark m = 0, 1, n = 0 to 7

Figure 26-3. 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 27 VOLTAGE DETECTOR

27.1 Functions of Voltage Detector

The voltage detector 0 (LVD0) sets the reset detection voltage (V_{LVD0}) with the user option byte (000C1H).

The voltage detector 1 (LVD1) sets the reset detection voltage (V_{LVD1}) with the voltage detection level register.

The voltage detector has the following functions.

- The LVD0 circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVD0}), and generates an internal reset.
- The LVD1 circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVD1}), and generates an internal interrupt signal.
- For LVD0, the detection voltage (V_{LVD0}) can be selected from 3 levels (2 levels when used at 2.7 V ≤ V_{DD} ≤ 5.5 V) by the user option byte (For details, see **CHAPTER 32 OPTION BYTE**).
- For LVD1, the detection voltage (V_{LVD1}) can be selected from 11 levels (8 levels when used at 2.7 V ≤ V_{DD} ≤ 5.5 V) by the voltage detection level register.
- Operable in STOP mode.
- After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in the Electrical Specifications. This is done by using the LVD0 or controlling the externally input reset signal.
  After the power supply is turned off, this LSI should be entered in the STOP mode, or placed in the reset state by using the LVD0 or controlling the externally input reset signal before the supply voltage falls below operating range.

The reset and interrupt signals in each mode are generated as follows.

Reset mode LVD0	Interrupt mode LVD1
Deasserts an internal reset signal on detecting $V_{DD} \geq V_{LVD0}$. Generates an internal reset on detecting $V_{DD} < V_{LVD0}$ and retains the reset state until $V_{DD} \geq V_{LVD0}$ is detected.	Generates an interrupt request signal (INTLVI) on detecting $V_{DD} < V_{LVD1}$ after LVD1 operation has been enabled. After the first detection, generates an interrupt request signal (INTLVI) on detecting $V_{DD} < V_{LVD1}$ or $V_{DD} \geq V_{LVD1}$.

While the LVD0 and LVD1 are operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVDnF: bit 0 and 1 of the voltage detection register (LVIM)).

If reset occurs, bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1. For details of the RESF register, see **CHAPTER 25 RESET FUNCTION**.

27.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in **Figure 27-1** and **Figure 27-2**.

Figure 27-1. Block Diagram of LVD0

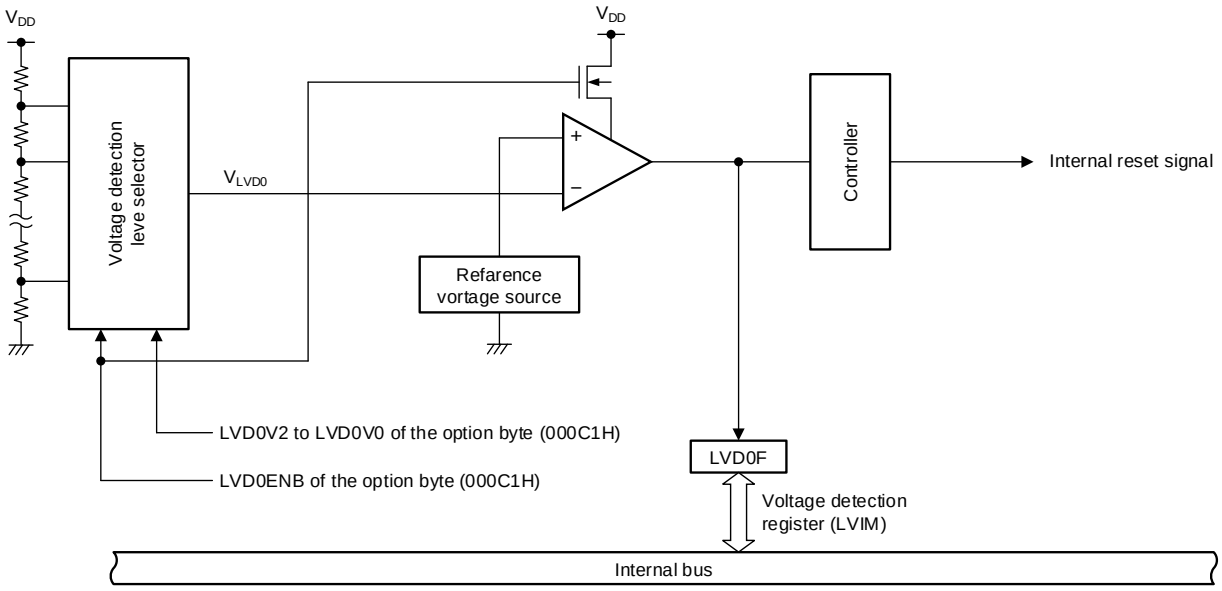
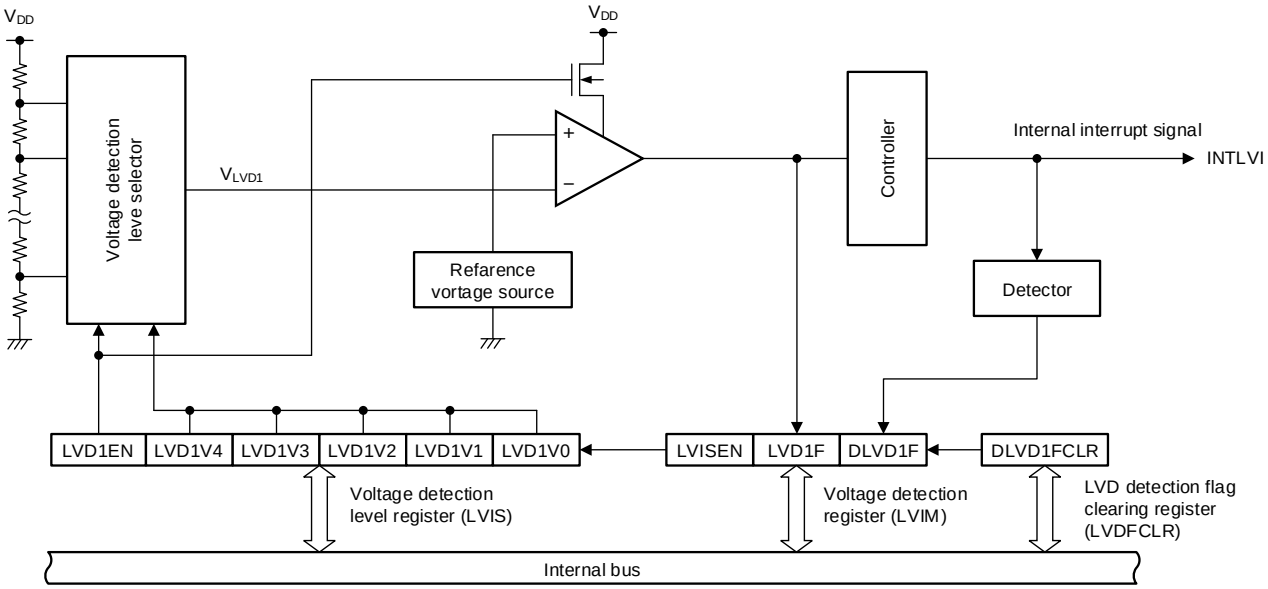


Figure 27-2. Block Diagram of LVD1



27.3 Registers Controlling Voltage Detector

The following registers are used to control the voltage detector.

Address	Register Name	Symbol	After Reset	Access Size
FFFA9H	Voltage detection register	LVIM	00H Note	1, 8
FFFAAH	Voltage detection level register	LVIS	19H	1, 8
F00FAH	LVD detection flag clearing register	LVDFCLR	00H	1, 8

Note The value after a reset is 01H when LVD0 operation is enabled and the power supply voltage (V_{DD}) is less than the detection voltage (V_{LVD0}).

27.3.1 Voltage Detection Register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), and used to check the status of LVD0 and LVD1.

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 27-3. Format of Voltage Detection Register (LVIM)

Address: FFFA9H After reset: 00H R/W Note 1, 2

Symbol	<7>	6	5	4	<3>	2	<1>	<0>
LVIM	LVISEN	0	0	0	DLVD1F	0	LVD1F	LVD0F

LVISEN	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Disabling rewriting
1 Note 3	Enabling rewriting (Interrupt generation by LVD1 is masked)

DLVD1F	LVD1 detection interrupt flag
0	LVD1 interrupt has not been detected.
1	LVD1 interrupt has been detected.

LVDnF	Voltage detection flag (n = 0, 1)
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVDn}), or when LVD operation is disabled
1	Supply voltage (V_{DD}) < detection voltage (V_{LVDn})

Notes 1. Bits 0 and 1 are read-only.

2. Bit 3 is read-only. It can be cleared with the LVD detection flag clearing register (LVDFCLR).

3. While the LVISEN bit is 1, the interrupt generation by LVD1 are masked. Therefore, clear the LVISEN bit to 0 after having written a new value to the LVIS register.

27.3.2 Voltage Detection Level Register (LVIS)

This register is used to select the voltage detection level for LVD1.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation input sets this register to 19H.

Figure 27-4. Format of Voltage Detection Level Register (LVIS)

Address: FFFAAH After reset: 19H R/W

Symbol	<7>	6	5	<4>	<3>	<2>	<1>	<0>
LVIS	LVD1EN	0	0	LVD1V4	LVD1V3	LVD1V2	LVD1V1	LVD1V0

LVD1EN	Enabling operation of LVD1
0	Operation stopped
1	Operation enabled

LVD1V4	LVD1V3	LVD1V2	LVD1V1	LVD1V0	Detection voltages for LVD1 Note 1, 3		
						Rising edge	Falling edge
1	1	1	0	1	V_{LVD115}	1.88 V Note 2, 5	1.84 V Note 2, 5
1	1	1	0	0	V_{LVD114}	1.98 V Note 2, 5	1.94 V Note 2, 5
1	1	0	1	1	V_{LVD113}	2.09 V Note 5	2.04 V Note 5
1	1	0	0	1	Value after reset Note 4		
1	0	1	0	1	V_{LVD17}	2.82 V	2.76 V
1	0	1	0	0	V_{LVD16}	2.97 V	2.91 V
1	0	0	1	1	V_{LVD15}	3.13 V	3.06 V
1	0	0	1	0	V_{LVD14}	3.35 V	3.27 V
1	0	0	0	1	V_{LVD13}	3.55 V	3.47 V
1	0	0	0	0	V_{LVD12}	3.75 V	3.67 V
0	1	1	1	1	V_{LVD11}	3.96 V	3.88 V
0	1	1	1	0	V_{LVD10}	4.16 V	4.08 V

Notes 1. The LVD1V4 to LVD1V0 bits can only be rewritten once after release from the reset state. For subsequent writes, use the same value.

2. This setting can only be used when LVD0 is off.

3. When LVD0 is used, the detection voltage of LVD1 should be set higher than that of LVD0.

4. When using LVD1, be sure to set the LVD1V4 to LVD1V0 bit to a value other than the reset value.

5. Selectable only when using 1.8 V ≤ V_{DD} < 2.7V for RL78/F22 products. Do not use under other conditions.

Caution When rewriting the LVIS register, follow the procedure described in 27.5.5 Procedure for Setting the LVD1 Detection Voltage.

The LVD0 operating mode and detection voltage (V_{LVD0}) are set in the user option byte (000C1H/040C1H). The settings of the option bytes are shown in **Table 27-1**.

Figure 27-5. Format of User Option Byte (000C1H/040C1H)

Address: 000C1H/040C1H Note 1 After reset: — (user setting value Note 2)

7	6	5	4	3	2	1	0
LVD0ENB	1	1	CLKMB Note 3	1	LVD0V2	LVD0V1	LVD0V0

Table 27-1. LVD0 Operation Mode and Detection Voltage Settings by User Option Byte (000C1H/040C1H)

(1) Settings in reset mode

Detection voltage		Option byte Setting Value			
V_{LVD0}		LVD0ENB	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge				
1.90 V Note 4	1.86 V Note 4	0	1	1	0
2.97 V	2.91 V		0	1	1
3.96 V	3.88 V		0	1	0
Other than above		Setting prohibited			

(2) Settings when LVD0 is off

Detection voltage		Option byte Setting Value			
V_{LVD0}		LVD0ENB	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge				
—	—	1	0	1	0
Other than above		Setting prohibited			

Notes 1. Set the same value as 000C1H to 040C1H when the boot swap operation is used because 000C1H is replaced by 040C1H.

2. The setting at shipment of the user option byte is FFH.

3. Write the setting value of the clock monitor bit (CLKMB).

4. Can be set only for RL78/F22 products; do not set for RL78/F25 products.

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 RESET pin. To perform an external reset upon power application, input a low level to the 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 RESET pin during a period in which the supply voltage is not within the operating range.

27.3.3 LVD Detection Flag Clearing Register (LVDFCLR)

This register is used to clear the interrupt detection flag (DLVD1F) of the voltage detection register (LVIM).

The LVDFCLR register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 27-6. Format of Voltage Detection Level Register (LVDFCLR)

Address: F00FAH After reset: 00H R/W

Symbol	7	6	5	4	<3>	2	1	0
LVDFCLR	0	0	0	0	DLVD1FCLR	0	0	0

DLVD1FCLR Note	Clearing DLVD1F
0	No effect
1	Clear the DLVD1F flag by writing 1 to this bit

Note Only 1 can be written to this bit. Writing 0 has no effect.
The bit is read as 0 even after 1 has been written to it.

27.4 Operation of Voltage Detector

27.4.1 When Used as Reset Mode

For LVD0, setting the detection voltage (V_{LVD0}) are done by using user option byte (000C1H/040C1H).

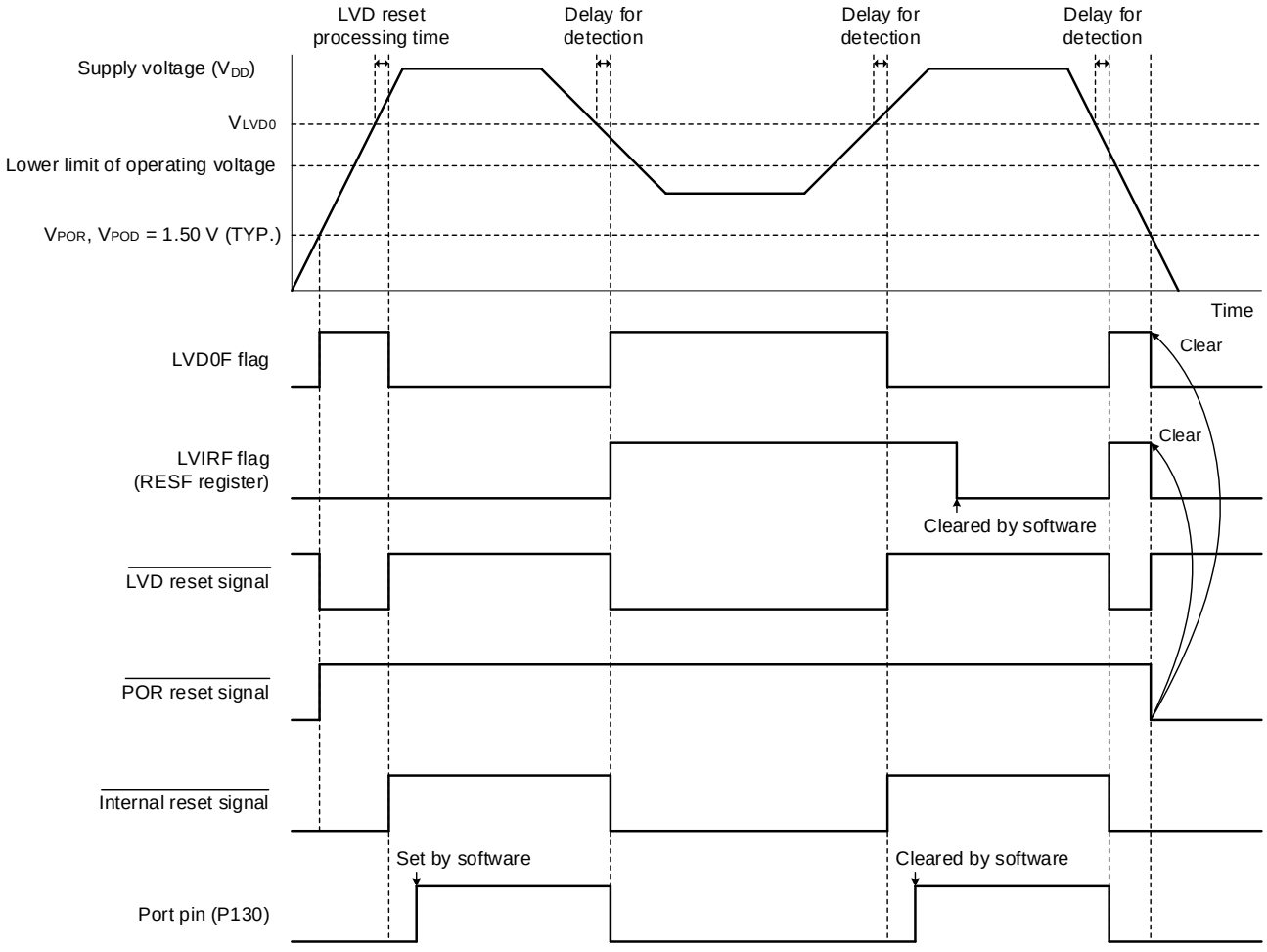
- Operation in LVD reset mode

When LVD0 is in reset mode, the state of the internal reset being applied by LVD0 is retained until the power supply voltage (V_{DD}) exceeds the rising voltage detection level (V_{LVD0}) after power has been supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the rising voltage detection level (V_{LVD0}).

At the fall of the operating voltage, an internal reset by LVD0 is generated when the power supply voltage (V_{DD}) falls below the falling voltage detection level (V_{LVD0}).

Figure 27-7 shows the timing of the internal reset signal generated in the LVD reset mode.

Figure 27-7. Timing of LVD0 Internal Reset Signal Generation



Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

27.4.2 When Used as Interrupt Mode

For LVD1, setting the detection voltage (V_{LVD1}) are done by using the voltage detection level register (LVIS).

- Operation in LVD interrupt mode

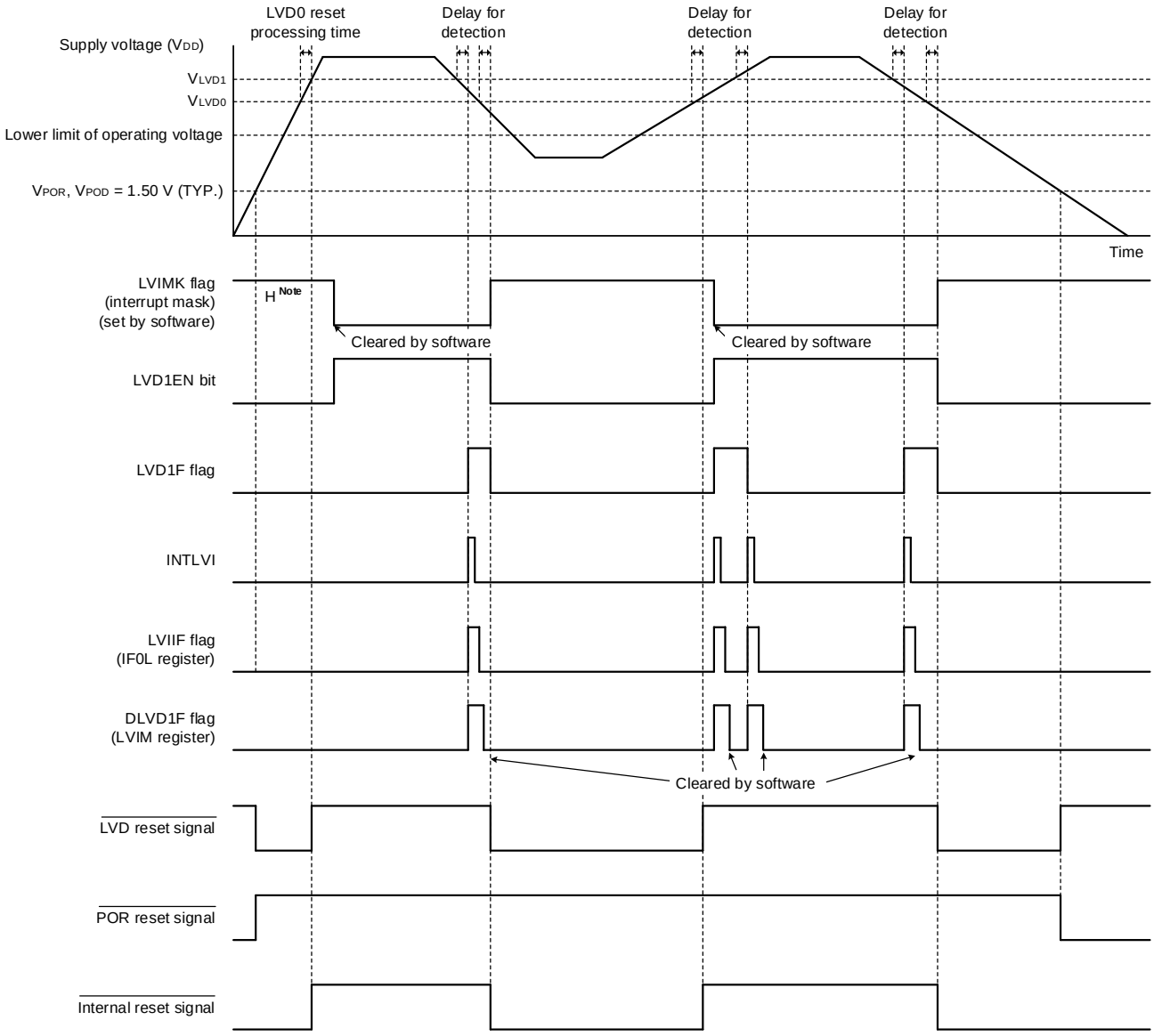
  Operation of LVD1 is stopped when power is initially supplied. When LVD1 operation is enabled, it generates an interrupt request signal (INTLVI) when the power supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD1}).

  Similarly, when the power supply voltage (V_{DD}) rises above the voltage detection level (V_{LVD1}), LVD1 also generates an interrupt request signal (INTLVI). Note that if operation of LVD1 is enabled while the power supply voltage (V_{DD}) is lower than the voltage detection level (V_{LVD1}), it generates an interrupt request signal (INTLVI) at the time its operation is enabled.

  LVD1 detection voltage can only be set once after release from the reset state.

  Figure 27-8 shows the timing of the interrupt request signal generated in the LVD interrupt mode.

Figure 27-8. Timing of LVD1 Interrupt Signal Generation (LVD0: Operation Enabled)



Note The LVIMK flag is set to 1 by reset signal generation.

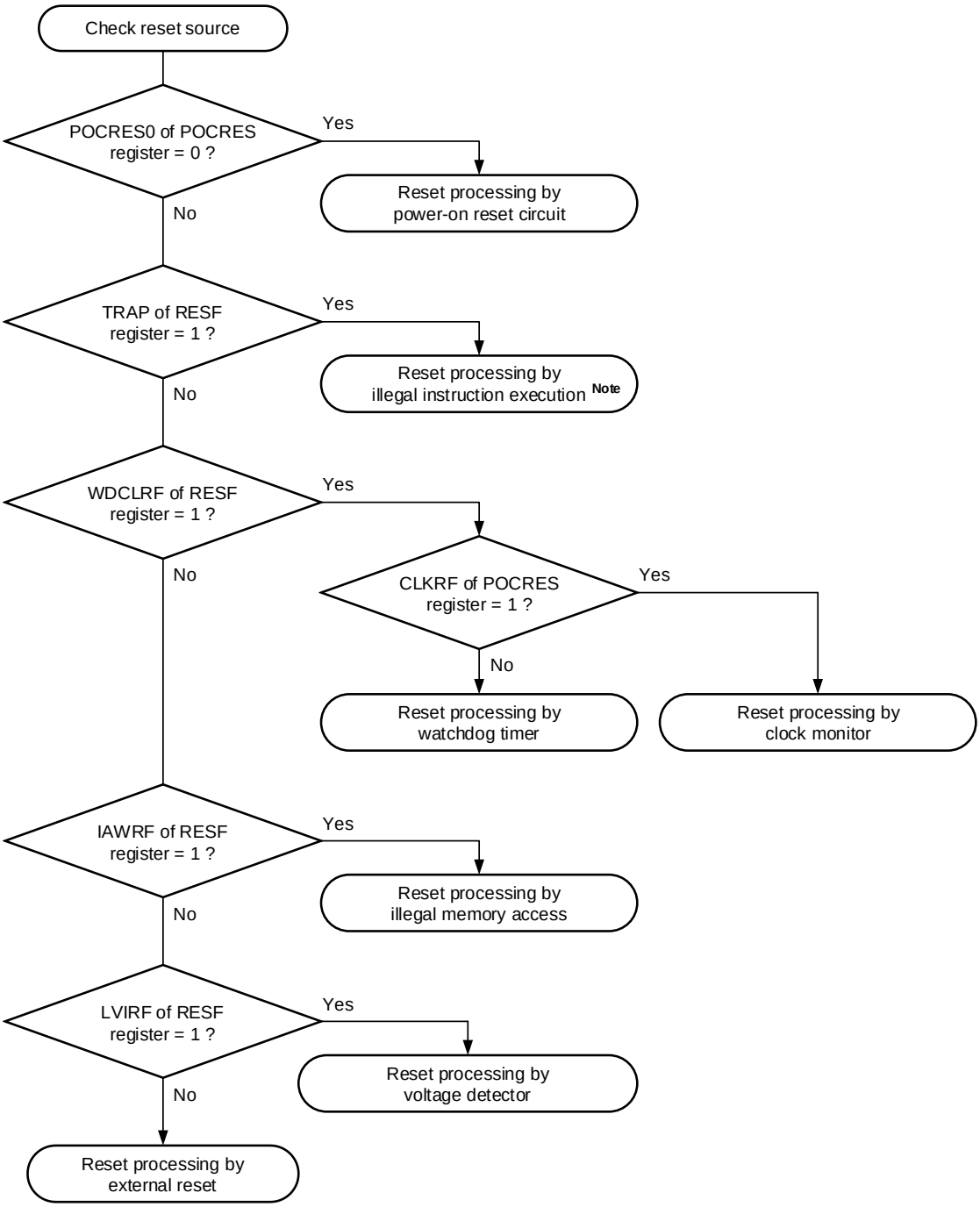
Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

27.5 Cautions for Voltage Detector

27.5.1 Checking Reset Source

When a reset occurs, check the reset source by using the following method.

Figure 27-9. Checking Reset Source



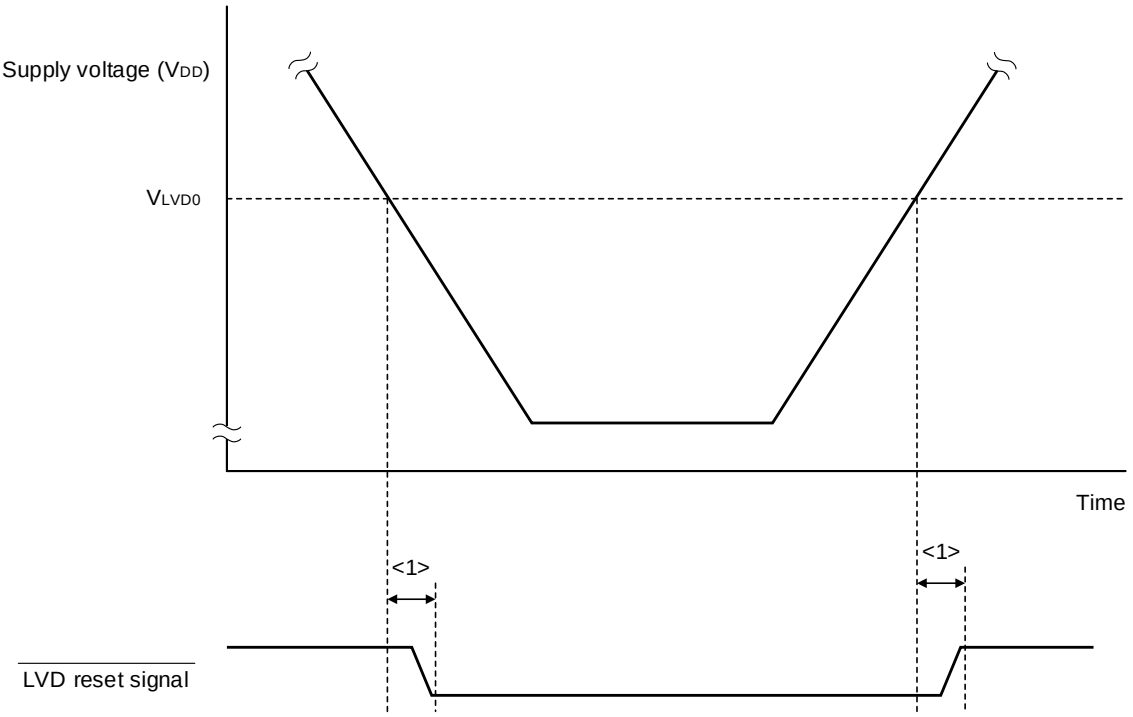
Note When instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

27.5.2 Delay from the Time LVD0 Reset Source Is Generated Until the Time LVD0 Reset Has Been Generated or Released

There is some delay from the time supply voltage (V_{DD}) < LVD0 detection voltage (V_{LVD0}) until the time LVD0 reset has been generated.

In the same way, there is also some delay from the time LVD0 detection voltage (V_{LVD0}) ≤ supply voltage (V_{DD}) until the time LVD0 reset has been released (see **Figure 27-10**).

Figure 27-10. Delay from The Time LVD Reset Source Is Generated until The Time LVD0 Reset Has Been Generated or Released



<1>: Detection delay (300 μs (MAX.))

27.5.3 Turning Power on When LVD0 Is Off

Use the external rest input via the RESET pin when LVD0 is off.

For an external reset, input a low level for 10 μs or more to the RESET pin. To perform an external reset upon power application, input a low level to the RESET pin, turn power on, continue to input a low level to the pin for 10 μs or more within the operating voltage range defined in the electrical specifications, and then input a high level to the pin.

27.5.4 Operating Voltage Fall When LVD0 Is Off

When the operating voltage falls with LVD0 off, this MCU should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal before the voltage falls below the operating voltage range defined in the electrical specifications. When restarting the operation, confirm that the power supply voltage has returned to the operating voltage range.

27.5.5 Procedure for Setting the LVD1 Detection Voltage

Follow the procedure below to set the LVD1 detection voltage. After step (3), LVD1 is enabled after the stabilization waiting time (at least 500 μs) has elapsed.

(1) Set the LVISEN bit of the LVIM register to 1.
(2) Set the LVD1EN bit of the LVIS register to 1 and change the setting of the LVD1V4 to LVD1V0 bits.
(3) Set the LVISEN bit of the LVIM register to 0.

CHAPTER 28 APPLICATION ACCELERATOR UNIT

28.1 Overview

The Application Accelerator Unit (AAU) is the dedicated arithmetic assist hardware to reduce the software load for FOC (Field Oriented Control) algorithm processing on BLDC motor control and PI (Proportional Integral) algorithm processing on DC/DC converter control.

The Application Accelerator Unit (AAU) can perform the operation in the algorithm modes shown in the **Table 28-1. Application Accelerator Unit Operation List**.

Table 28-1. Application Accelerator Unit Operation List Note 2

Algorithm Mode	Data Length (Input/Output)	Execution Cycle
Sine operation	16 bits/16 bits	1 clock
Cosine operation	16 bits/16 bits	1 clock
Clarke and Park transformation Note 1	16 bits/16 bits	7 clocks
Inverse Park (I-Park) transformation	16 bits/16 bits	6 clocks
Inverse Clarke (I-Clarke) transformation Note 1	16 bits/16 bits	5 clocks
PI control for motor operation	16 bits/16 bits	15 clocks
Clarke and Park transformation and PI control for motor operation Note 1	16 bits/16 bits	22 clocks
I-Park and I-Clarke transformation Note 1	16 bits/16 bits	11 clocks
PI control for DC/DC converter control operation (total channel: 1)	16 bits/16 bits	6 clocks
PI control for DC/DC converter control operation (total channel: 2)	16 bits/16 bits	12 clocks
PI control for DC/DC converter control operation (total channel: 3)	16 bits/16 bits	18 clocks
Multiply: 32 bits × 32 bits = 64 bits operation	32 bits/64 bits	5 clocks

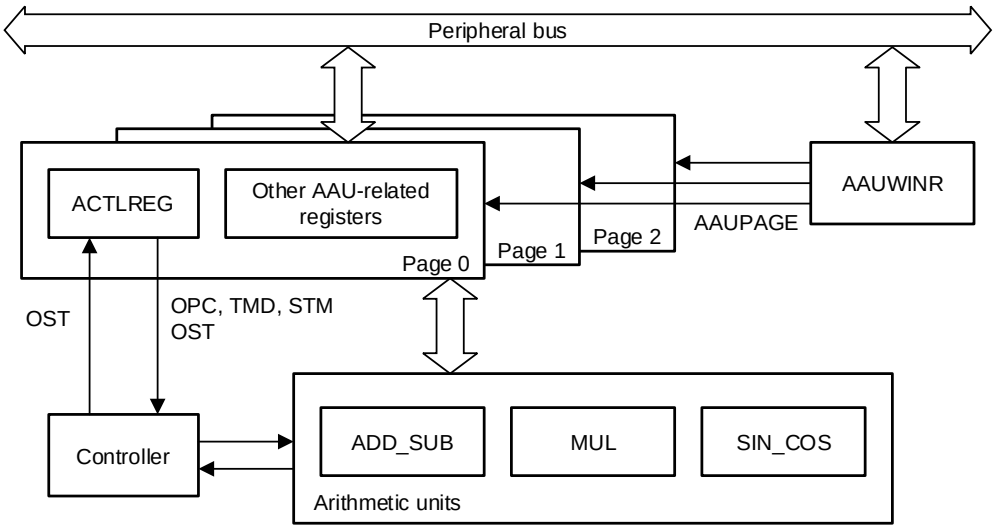
Notes 1. The AAU supports both algorithms, power invariant transformation and amplitude invariant transformation.

2. For details about the operations of each algorithm mode, see **28.3.2 Description for the operation of each algorithm mode**.

Figure 28-1 shows a block diagram of AAU.AAU consists of the following three arithmetic units, a 32-bit adder-subtractor (ADD_SUB), a 16-bit multiplier (MUL) and a sine-cosine calculator (SIN_COS).

In this chapter, PCLK is used to refer to CPU/peripheral hardware clock (f_{CLK}).

Figure 28-1. Block Diagram of Application Accelerator Unit



28.2 Registers

Table 28-2. Application Accelerator Unit Register Configuration

Address	Register Name	Symbol	After Reset	Access Size
F02C2H	Peripheral enable register 2	PER2	00H	1, 8
FFF32H	Application accelerator unit access window register	AAUWINR	00H	8
F02B0H : F02BFH	Application accelerator unit window registers For details, see **Table 28-3** to **Table 28-5**.	–	–	–

Table 28-3. Application Accelerator Unit Window Register (Page.0 Note)

Address	Register Name	Symbol	After Reset	Access Size
F02B0H	AAU data register 0	ADTREG0	0000H	16
F02B2H	AAU data register 1	ADTREG1	0000H	16
F02B4H	AAU data register 2	ADTREG2	0000H	16
F02B6H	AAU data register 3	ADTREG3	0000H	16
F02BAH	AAU control register	ACTLREG	00H	1, 8
F02BBH	PI control coefficient range setting register/ Current limit duty register	AKRAG/ ADUTYMX	00H	8
F02BCH	D-axis current reference register/ DC/DC channel.1 current reference register	AIDREF/ AL1REF	0000H	16
F02BEH	Q-axis current reference register/ DC/DC channel.2 current reference register	AIQREF/ AL2REF	0000H	16

Note AAUWINR.AAUPAGE[1:0] bits are the condition of 00B (page.0 access selection).

Table 28-4. Application Accelerator Unit Window Register (Page.1 Note)

Address	Register Name	Symbol	After Reset	Access Size
F02B0H	D-axis proposal coefficient register/ DC/DC channel.3 current reference register	AKPD/ AL3REF	0000H	16
F02B2H	D-axis integral coefficient register/ DC/DC channel.1 offset current register	AKID/ AL1OFS	0000H	16
F02B4H	Q-axis proposal coefficient register/ DC/DC channel.2 offset current register	AKPQ/ AL2OFS	0000H	16
F02B6H	Q-axis integral coefficient register/ DC/DC channel.3 offset current register	AKIQ/ AL3OFS	0000H	16
F02BCH	Current limit register/ DC/DC proportional coefficient 1 register	AILIM/ AKI1	0000H	16
F02BEH	PI control limit register/ DC/DC proportional coefficient 2 register	APILIM/ AKI2	0000H	16

Note AAUWINR.AAUPAGE[1:0] bits are the condition of 01B (page.1 access selection).

Table 28-5. Application Accelerator Unit Window Register (Page.2 [Note])

Address	Register Name	Symbol	After Reset	Access Size
F02B0H	D-axis current buffer register L/ DC/DC channel.1 previous duty register	AIDBFL/ ADUTYL1	0000H	16
F02B2H	D-axis current buffer register H/ DC/DC channel.2 previous duty register	AIDBFH/ ADUTYL2	0000H	16
F02B4H	Q-axis current buffer register L/ DC/DC channel.3 previous duty register	AIQBFL/ ADUTYL3	0000H	16
F02B6H	Q-axis current buffer register H/ DC/DC channel.1 previous current register	AIQBFH/ AIPL1	0000H	16
F02B8H	D-axis limit over current register/ DC/DC channel.2 previous current register	ADOVER/ AIPL2	0000H	16
F02BAH	Q-axis limit over current register/ DC/DC channel.3 previous current register	AQOVER/ AIPL3	0000H	16

Note AAUWINR.AAUPAGE[1:0] bits are the condition of 10B (page.2 access selection).

28.2.1 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.

When using peripheral functions controlled by PER2, set the bit corresponding to each function to “1” before initializing 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.

Figure 28-2. Format of Peripheral Enable Register 2 (PER2)

Address: F02C2H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	1	<0>
PER2	0	AAUEN	CAN1EN Note	LIN2EN Note	LIN1EN Note	LIN0EN	0	CAN0EN Note

AAUEN	Control of AAU input clock supply
0	Stops input clock supply. • Disables writing to the SFR used by AAU. • AAU is in the reset state.
1	Enables input clock supply. • Enables reading from and writing to the SFR used by AAU.

Note Only in the RL78/F25 products.

Cautions 1. When setting application accelerator unit, be sure to set the AAUEN bit to 1 first. If AAUEN = 0, writing to a control register of application accelerator unit is ignored, and all read values are default values.

2. Be sure to clear the following bits to 0.
RL78/F22 products: bits 0, 1, 3, 4, 5 and 7
RL78/F25 products: bits 1 and 7

Remark See **5.3.6 Peripheral Enable Registers 0, 1, 2 (PER0, PER1, PER2)** for a description of bits other than AAUEN.

28.2.2 Application Accelerator Unit Access Window Register (AAUWINR)

This register controls the selection of the address space. The address range from address F02B0H to F02BFH is multiplexed in to 3 pages.

Set the AAUWINR register by an 8-bit memory manipulation instruction.

Figure 28-3. Format of Application Accelerator Unit Access Window Register (AAUWINR)

Address: FFF32H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
AAUWINR	0	0	0	0	0	0	AAUPAGE1	AAUPAGE0

AAUPAGE1	AAUPAGE0	Selection of AAU registers access page [Note]
0	0	Page 0 is selected.
0	1	Page 1 is selected.
1	0	Page 2 is selected.
1	1	Setting is prohibited.

Note SFRs in each page are describe in the **Table 28-3** to **Table 28-5**.

[Registers allocated when page 0 is selected]

- ADTREGi (i =0 to 3), ACTLREG, AKRAG/ADUTYMX, AIDREF/AL1REF, AIQREF/AL2REF

[Registers allocated when page 1 is selected]

- AKPD/AL3REF, AKID/AL1OFS, AKPQ/AL2OFS, AKIQ/AL3OFS, AILIM/AKI1, APILIM/AKI2

[Registers allocated when page 2 is selected]

- AIDBFL/ADUTYL1, AIDBFH/ADUTYL2, AIQBFL/ADUTYL3, AIQBFH/AIPL1, ADOVER/AIPL2, AQOVER/AIPL3

28.2.3 AAU Data Register i (ADTREGi, i = 0 to 3)

The ADTREGi register is input/output data register for the application accelerator unit. This register is a 16-bit register.

This register can be read or written in 16-bit units.

The calculation result is overwritten to ADTREGi after the operation completion.

Figure 28-4. Format of AAU Data Register i (ADTREGi, i = 0 to 3)

Address: F02B0H Note (ADTREG0), F02B2H Note (ADTREG1), F02B4H Note (ADTREG2), F02B6H Note (ADTREG3) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADTREGi																

Note This register is allocated to the AAU page 0. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 00B.

Bits 15 to 0	AAU data register (ADTREGi)
ADTREG0	Used in each algorithm mode. For details, see the following section. • Section **28.3.2.1** to **28.3.2.14**
ADTREG1	Used in each algorithm mode. For details, see the following section. • Sections **28.3.2.3** to **28.3.2.14**
ADTREG2	Used in each algorithm mode. For details, see the following section. • Sections **28.3.2.3** to **28.3.2.8**, **28.3.2.10** to **28.3.2.14**
ADTREG3	Used in each algorithm mode. For details, see the following section. • Sections **28.3.2.3**, **28.3.2.10**, and **28.3.2.11**

28.2.4 AAU Control Register (ACTLREG)

This register controls function and start trigger of AAU and indicates operation status of AAU.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 28-5. Format of AAU Control Register (ACTLREG)

Address: F02BAH [Note 1] After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
ACTLREG	OPC4	OPC3	OPC2	OPC1	OPC0	TMD	STM	OST

Bits OPC4 to OPC0	Selection of algorithm mode
00000B	No operation
00001B	Multiply operation
00010B	Sine operation
00011B	Cosine operation
01000B	Clarke and Park transformation
01001B	Inverse Park (I-Park) transformation
01010B	Inverse Clarke (I-Clarke) transformation
01011B	PI control for motor
01100B	Clarke & Park transformation and PI control for motor
01101B	I-Park & I-Clarke transformation
10000B	PI control for DC/DC converter control 1 channel
10001B	PI control for DC/DC converter control 2 channels
10010B	PI control for DC/DC converter control 3 channels
Other than above	Setting prohibited

TMD	Selection of transformation mode [Note 2]
0	Power invariant transformation
1	Amplitude invariant transformation

STM	Selection of start trigger
0	Triggered by the last operation completion
1	Software trigger by OST bit

OST	Operation status of the AAU or the trigger to start selected algorithm operation
0	Operation completion or no trigger
1	Operation processing Operation is in progress. Or starts operation if set to 1 when STM = 1. After completion of the operation, this bit is cleared automatically.

Notes 1. This register is allocated to the AAU page 0. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 00B.

2. This bit is valid only in the following operating modes.

OPC[4:0] = 01000B (Clarke and Park transformation), 01010B (I-Clarke transformation),
01100B (Clarke and Park transformation, and PI control for motor),
01101B (I-Park and I-Clarke transformation)

28.2.5 PI Control Coefficient Range Setting Register/ Current Limit Duty Register (AKRAG/ ADUTYMX)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

This register selects the coefficient range for PI control. This range is applied to both d-axis and q-axis.

This register can be set by an 8-bit memory manipulation instruction.

This register used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Figure 28-6. Format of PI Control Coefficient Range Setting Register (AKRAG)

Address: F02BBH Note After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
AKRAG								

AKRAG	Coefficient range
00H	$1/2^4$
01H	$1/2^8$
02H	$1/2^{12}$
03H	$1/2^{16}$
Other than above	Setting prohibited

Note This register is allocated to the AAU page 0. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 00B.

[As using for DC/DC converter control operation]

This register is set the current limit duty value for PI control.

This register can be set by an 8-bit memory manipulation instruction.

Figure 28-7. Format of Current Limit Duty Register (ADUTYMX)

Address: F02BBH Note After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADUTYMX								

Bits 7 to 0	Current limit duty register (ADUTYMX)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 0. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 00B.

28.2.6 D-axis Current Reference Register/ DC/DC Channel 1 Current Reference Register (AIDREF/AL1REF)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

This register is set the reference value of d-axis current.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-8. Format of D-axis Current Reference Register (AIDREF)

Address: F02BCH Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIDREF																

Bits 15 to 0	D-axis current reference register (AIDREF)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 0. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 00B.

[As using for DC/DC converter control operation]

This register is set the reference value of channel 1 current.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-9. Format of DC/DC Channel 1 Current Reference Register (AL1REF)

Address: F02BCH Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AL1REF																

Bits 15 to 0	DC/DC channel 1 current reference register (AL1REF)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 0. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 00B.

28.2.7 Q-axis Current Reference Register/ DC/DC channel 2 Current Reference Register (AIQREF/AL2REF)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

This register is set the reference value of q-axis current.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-10. Format of Q-axis Current Reference Register (AIQREF)

Address: F02BEH Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIQREF																

Bits 15 to 0	Q-axis current reference register (AIQREF)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 0. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 00B.

[As using for DC/DC converter control operation]

This register is set the reference value of channel 2 current.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-11. Format of DC/DC channel 2 Current Reference Register (AL2REF)

Address: F02BEH Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AL2REF																

Bits 15 to 0	DC/DC channel 2 current reference register (AL2REF)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 0. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 00B.

28.2.8 D-axis Proportional Coefficient Register/ DC/DC Channel 3 Current Reference Register (AKPD/AL3REF)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

This register is set the PI control proportional coefficient for d-axis.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-12. Format of D-axis Proportional Coefficient Register (AKPD)

Address: F02B0H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AKPD																

Bits 15 to 0	D-axis proportional coefficient register (AKPD)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 1. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 01B.

[As using for DC/DC converter control operation]

This register is set the reference value of channel 3 current.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-13. Format of DC/DC Channel 3 Current Reference Register (AL3REF)

Address: F02B0H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AL3REF																

Bits 15 to 0	DC/DC channel 3 current reference register (AL3REF)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 1. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 01B.

28.2.9 D-axis Integral Coefficient Register/ DC/DC Channel 1 Offset Current Register (AKID/AL1OFS)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

This register is set the PI control Integral coefficient for d-axis.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-14. Format of D-axis Integral Coefficient Register (AKID)

Address: F02B2H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AKID																

Bits 15 to 0	D-axis integral coefficient register (AKID)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 1. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 01B.

[As using for DC/DC converter control operation]

This register is set the value of channel 1 offset current.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-15. Format of DC/DC Channel 1 Offset Current Register (AL1OFS)

Address: F02B2H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AL1OFS																

Bits 15 to 0	DC/DC channel 1 offset current register (AL1OFS)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 1. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 01B.

28.2.10 Q-axis Proportional Coefficient Register/ DC/DC Channel 2 Offset Current Register (AKPQ/AL2OFS)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

This register is set the PI control proportional coefficient for q-axis.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-16. Format of Q-axis Proportional Coefficient Register (AKPQ)

Address: F02B4H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AKPQ																

Bits 15 to 0	Q-axis proportional coefficient register (AKPQ)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 1. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 01B.

[As using for DC/DC converter control operation]

This register is set the value of channel 2 offset current.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-17. Format of DC/DC Channel 2 Offset Current Register (AL2OFS)

Address: F02B4H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AL2OFS																

Bits 15 to 0	DC/DC channel 2 offset current register (AL2OFS)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 1. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 01B.

28.2.11 Q-axis Integral Coefficient Register/ DC/DC Channel 3 Offset Current Register (AKIQ/AL3OFS)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

This register is set the PI control Integral coefficient for q-axis.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-18. Format of Q-axis Integral Coefficient Register (AKIQ)

Address: F02B6H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AKIQ																

Bits 15 to 0	Q-axis integral coefficient register (AKIQ)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 1. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 01B.

[As using for DC/DC converter control operation]

This register is set the value of channel 3 offset current.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-19. Format of DC/DC Channel 3 Offset Current Register (AL3OFS)

Address: F02B6H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AL3OFS																

Bits 15 to 0	DC/DC channel 3 offset current register (AL3OFS)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 1. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 01B.

28.2.12 Current Limit Register/ DC/DC Proportional Coefficient 1 Register (AILIM/AKI1)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

This register is set the current limitation value of integral control.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-20. Format of Current Limit Register (AILIM)

Address: F02BCH **Note** After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AILIM																

Bits 15 to 0	Current limit register (AILIM)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 1. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 01B.

[As using for DC/DC converter control operation]

This register is set the proportional coefficient 1 of PI control for DC/DC converter control.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-21. Format of DC/DC Proportional Coefficient 1 Register (AKI1)

Address: F02BCH **Note** After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AKI1																

Bits 15 to 0	DC/DC proportional coefficient 1 register (AKI1)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 1. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 01B.

28.2.13 PI Control Limit Register/ DC/DC Proportional Coefficient 2 Register (APILIM/AKI2)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

This register is set the current limitation value of FOC PI control.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-22. Format of PI Control Limit Register (APILIM)

Address: F02BEH Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APILIM																

Bits 15 to 0	PI control limit register (APILIM)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 1. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 01B.

[As using for DC/DC converter control operation]

This register is set the proportional coefficient 2 of PI control for DC/DC converter control.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-23. Format of DC/DC Proportional Coefficient 2 Register (AKI2)

Address: F02BEH Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AKI2																

Bits 15 to 0	DC/DC proportional coefficient 2 register (AKI2)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 1. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 01B.

28.2.14 D-axis Current Buffer Register L/ DC/DC Channel 1 Previous Duty Register (AIDBFL/ADUTYL1)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

The d-axis current buffer length is 32 bits. The AIDBFL register is set the lower 16-bit current buffer of d-axis.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-24. Format of D-axis Current Buffer Register L (AIDBFL)

Address: F02B0H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIDBFL																

Bits 15 to 0	D-axis current buffer register L (AIDBFL)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 2. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 10B.

[As using for DC/DC converter control operation]

The ADUTYL1 register is set the previous duty value for channel 1.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-25. Format of DC/DC Channel 1 Previous Duty Register (ADUTYL1)

Address: F02B0H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADUTYL1																

Bits 15 to 0	DC/DC channel 1 previous duty register (ADUTYL1)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 2. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 10B.

28.2.15 D-axis Current Buffer Register H/ DC/DC Channel 2 Previous Duty Register (AIDBFH/ADUTYL2)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

The d-axis current buffer length is 32 bits. The AIDBFH register is set the upper 16-bit current buffer of d-axis.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-26. Format of D-axis Current Buffer Register H (AIDBFH)

Address: F02B2H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIDBFH																

Bits 15 to 0	D-axis current buffer register H (AIDBFH)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 2. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 10B.

[As using for DC/DC converter control operation]

The ADUTYL2 register is set the previous duty value for channel 2.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-27. Format of DC/DC Channel 2 Previous Duty Register (ADUTYL2)

Address: F02B2H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADUTYL2																

Bits 15 to 0	DC/DC channel 2 previous duty register (ADUTYL2)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 2. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 10B.

28.2.16 Q-axis Current Buffer Register L/ DC/DC Channel 3 Previous Duty Register (AIQBFL/ADUTYL3)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

The q-axis current buffer length is 32 bits. The AIQBFL register is set the lower 16-bit current buffer of q-axis.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-28. Format of Q-axis Current Buffer Register L (AIQBFL)

Address: F02B4H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIQBFL																

Bits 15 to 0	Q-axis current buffer register L (AIQBFL)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 2. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 10B.

[As using for DC/DC converter control operation]

The ADUTYL3 register is set the previous duty value for channel 3.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-29. Format of DC/DC Channel 3 Previous Duty Register (ADUTYL3)

Address: F02B4H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADUTYL3																

Bits 15 to 0	DC/DC channel 3 previous duty register (ADUTYL3)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 2. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 10B.

28.2.17 Q-axis Current Buffer Register H/ DC/DC Channel 1 Previous Current Register (AIQBFH/AIPL1)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

The q-axis current buffer length is 32 bits. The AIQBFH register is set the upper 16-bit current buffer of q-axis.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-30. Format of Q-axis Current Buffer Register H (AIQBFH)

Address: F02B6H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIQBFH																

Bits 15 to 0	Q-axis current buffer register H (AIQBFH)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 2. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 10B.

[As using for DC/DC converter control operation]

The AIPL1 register is set the previous current value for channel 1.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-31. Format of DC/DC Channel 1 Previous Current Register (AIPL1)

Address: F02B6H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIPL1																

Bits 15 to 0	DC/DC channel 1 previous current register (AIPL1)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 2. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 10B.

28.2.18 D-axis Limit Over Current Register/ DC/DC Channel 2 Previous Current Register (ADOVER/AIPL2)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

This register is set the limit over current value of d-axis.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-32. Format of D-axis Limit over Current Register (ADOVER)

Address: F02B8H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADOVER																

Bits 15 to 0	D-axis limit over current register (ADOVER)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 2. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 10B.

[As using for DC/DC converter control operation]

This register is set the previous current value for channel 2.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-33. Format of DC/DC Channel 2 Previous Current Register (AIPL2)

Address: F02B8H Note After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIPL2																

Bits 15 to 0	DC/DC channel 2 previous current register (AIPL2)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 2. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 10B.

28.2.19 Q-axis Limit Over Current Register/ DC/DC Channel 3 Previous Current Register (AQOVER/AIPL3)

This register has two functions, respectively, FOC operation and DC/DC converter control operation.

[As using for FOC operation]

This register is set the limit over current value of q-axis.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-34. Format of Q-axis Limit over Current Register (AQOVER)

Address: F02BAH [Note] After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AQOVER																

Bits 15 to 0	Q-axis limit over current register (AQOVER)
–	Used in PI control for motor. For details, see section **28.3.2.9**, **28.3.2.10**, and **28.3.2.11**.

Note This register is allocated to the AAU page 2. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 10B.

[As using for DC/DC converter control operation]

This register is set the previous current value for channel 3.

This register can be set or read by a 16-bit memory manipulation instruction.

Figure 28-35. Format of DC/DC Channel 3 Previous Current Register (AIPL3)

Address: F02BAH [Note] After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIPL3																

Bits 15 to 0	DC/DC channel 3 previous current register (AIPL3)
–	Used in PI control for DC/DC converter control. For details, see section **28.3.2.14**.

Note This register is allocated to the AAU page 2. When accessing this register, set the AAUPAGE bits in the AAUWINR register to 10B.

28.3 Operation of the AAU

28.3.1 Operation Overview

AAU consists of arithmetic units, a 32-bit adder-subtractor, a 16-bit multiplier and a sine-cosine calculator. A single arithmetic operation and sequence operation are performed with these arithmetic units. The single arithmetic operation is performed with one arithmetic unit and the sequence operation is performed with combination of these arithmetic units.

There are two methods to do the operation. One is to set all operand data, and the other is to set the OST bit of the ACTLREG. The selection of calculation start is set by the STM bit of the ACTLREG. The completion of the operation status is set in the OST bit of the ACTLREG.

Algorithm mode want to execute is set by the OPC4-0 bits in the ACTLREG.

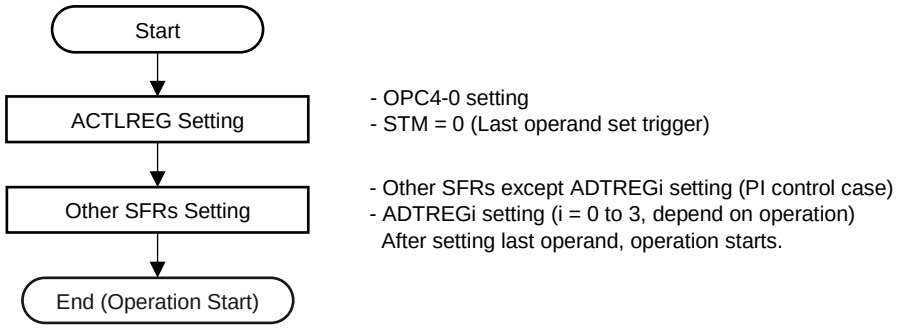
There is no interrupt source of operation completion. The OST bit of the ACTLREG should be read to know the status of the operation.

The control of AAU is done by S/W. Before using the AAU, the AAUEN bit in the PER2 register should be set to 1 to be negate reset state and supply operation clock.

Figure 28-36 shows processing flow of operation start, and **Figure 28-37** shows processing flow of read of operation results.

Figure 28-36. Processing Flow of Operation Start

(a) Start after all operand data set



(b) Start after setting of the OST bit in the ACTLREG

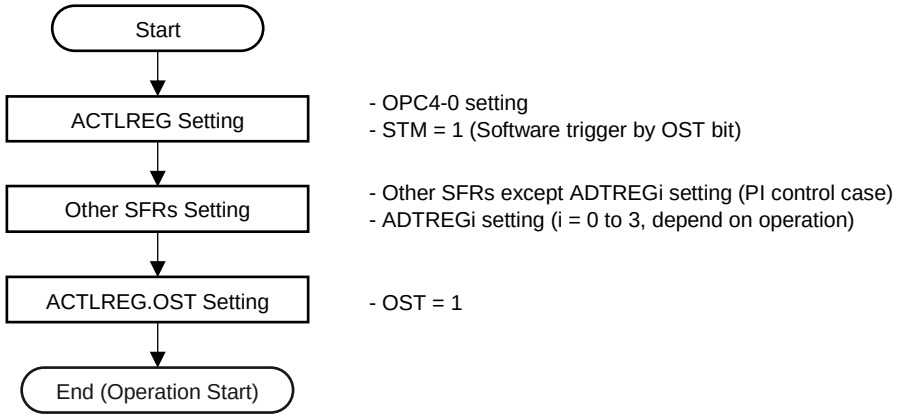
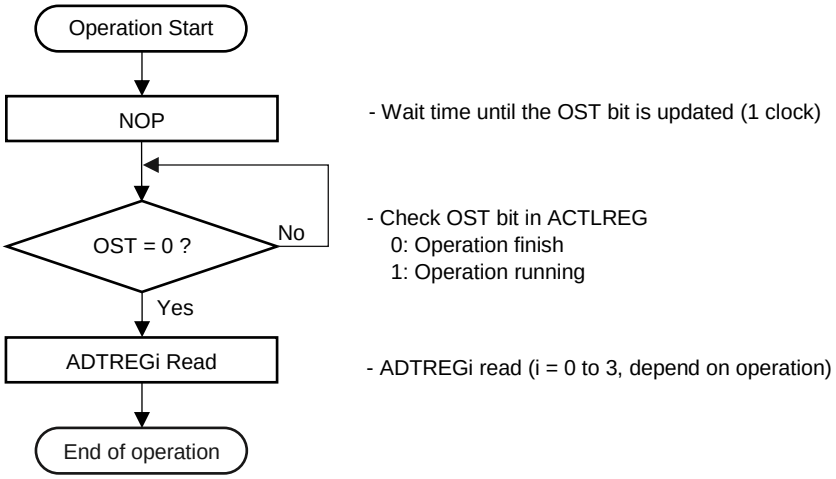
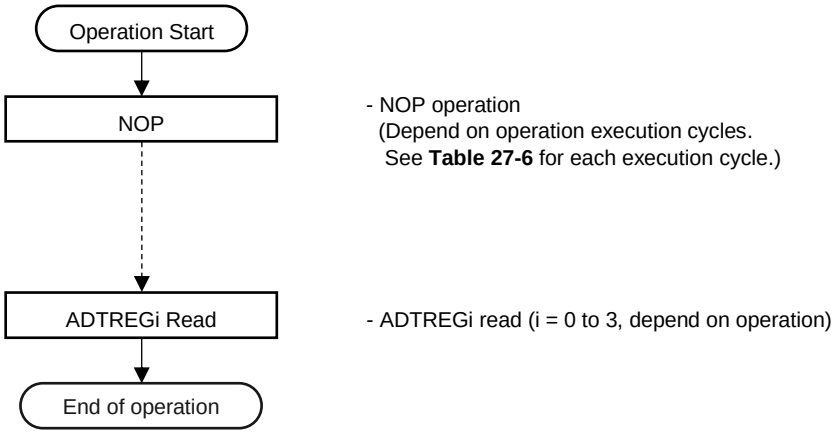


Figure 28-37. Processing Flow of Read of Operation Results

(a) Operation status check case



(b) NOP operation during operation running

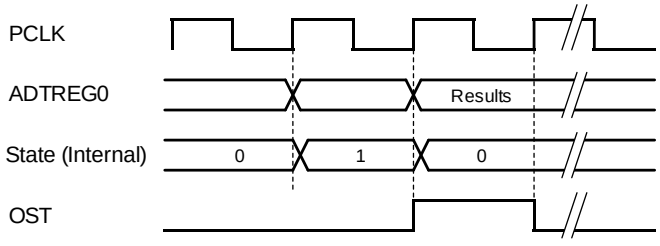


28.3.1.1 Single Arithmetic Operation

There are two operations of the single arithmetic operation, sine operation and cosine operation. For details, see **28.3.2 Description for operation of each algorithm mode**.

Figure 28-38 shows timing diagram of the single arithmetic operation.

Figure 28-38. Timing Diagram of Single Arithmetic Operation



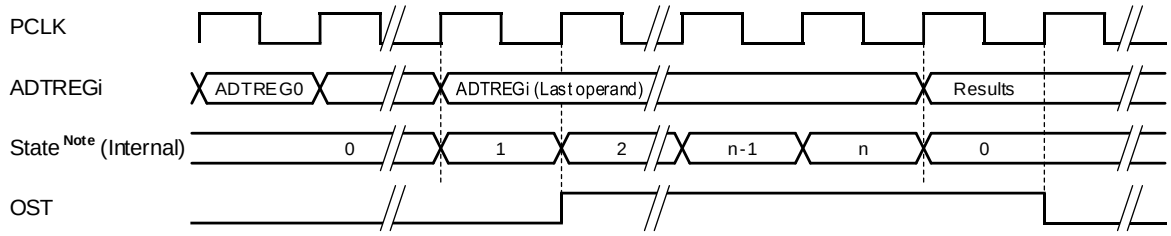
28.3.1.2 Sequence Operation

Other operations except sine and cosine operations are sequence operations. For details, see **28.3.2 Description for operation of each algorithm mode**.

Figure 28-39 shows timing diagram of the sequence operation.

The sequence operation cannot be aborted. The operation is performed until completion except reset assert and the application accelerator unit enable clear. The operation clock cycles of each sequence operation are shown in **Table 28-1**.

Figure 28-39. Timing Diagram of Sequence Operation (ACTCLREG.STM = 0 case)



Note State shows execution cycle counts of each operation.The value n = execution cycle: 1, 5 to 7, 11, 12, 15, 18, 22.

Remark i = 0 to 3 (depend on the operation)

28.3.1.3 Start Trigger of Each Operation

Table 28-6 summarizes start trigger of operations. The operation starts after the start trigger register setting.

Table 28-6. Start Trigger Register of AAU Operation

Algorithm Mode	Start Trigger Register		Execution Cycle Note 1
	STM = 0	STM = 1	
Sine operation	ADTREG0	OST bit	1 clock
Cosine operation	ADTREG0	OST bit	1 clock
Clarke and Park transformation Note 2	ADTREG2	OST bit	7 clocks
Inverse Park (I-Park) transformation	ADTREG2	OST bit	6 clocks
Inverse Clarke (I-Clarke) transformation Note 2	ADTREG1	OST bit	5 clocks
PI control for motor operation	ADTREG1	OST bit	15 clocks
Clarke and Park Transformation and PI Control for Motor Operation Note 2	ADTREG2	OST bit	22 clocks
I-Park and I-Clarke Transformation Note 2	ADTREG2	OST bit	11 clocks
PI control for DC/DC converter control operation (total channel: 1)	ADTREG0	OST bit	6 clocks
PI control for DC/DC converter control operation (total channel: 2)	ADTREG1	OST bit	12 clocks
PI control for DC/DC converter control operation (total channel: 3)	ADTREG2	OST bit	18 clocks
Multiply: 32 bits × 32 bits = 64 bits operation	ADTREG3	OST bit	5 clocks

Notes 1. If the STM bit in the ACTLREG register is 1, the execution cycle is added by 1 cycle to the value shown in the table.

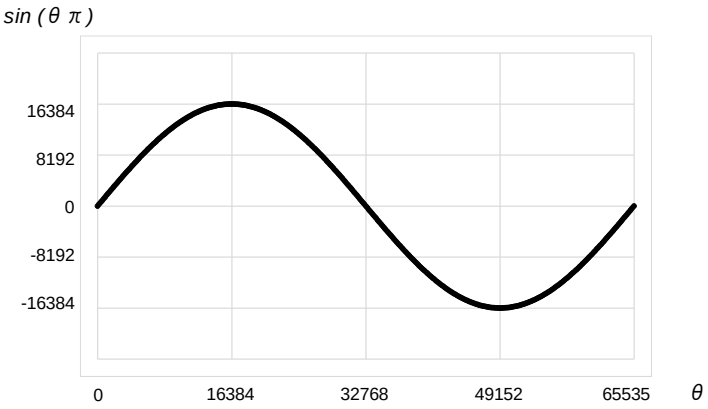
2. The AAU supports both algorithms, power invariant transformation and amplitude invariant transformation.

28.3.2 Description for the Operation of Each Algorithm Mode

28.3.2.1 Sine Operation

Sine calculation is performed.

Perform the Sine operation with an approximate expression using series expansion.



In/Out	Register name	Function	Data format
Input	ADTREG0 Note	Phase θ Range: 0 ≤ θ ≤ 65535	Data type: unsigned short
Output	ADTREG0	Sine value of θ Value = $\sin(\theta\pi)$	Data type: signed short

Note Setting value of ADTREG0 register: (rad × 65536) ÷ 2 π
= (rad × 32768) ÷ π [rad: Input radian angle]

Operation Example:

(1) Setting the ACTLREG register (OPC[4:0] = 00010B (Sine operation), STM = 0 (Enables the last operand write trigger))
ACTLREG = 10H;

(2) Set the data to be converted to ADTREG0 register and start the operation.
ADTREG0 = 8000H; Input radian angle = 3.14159

(3) Wait until the OST bit in the ACTLREG register becomes 0.

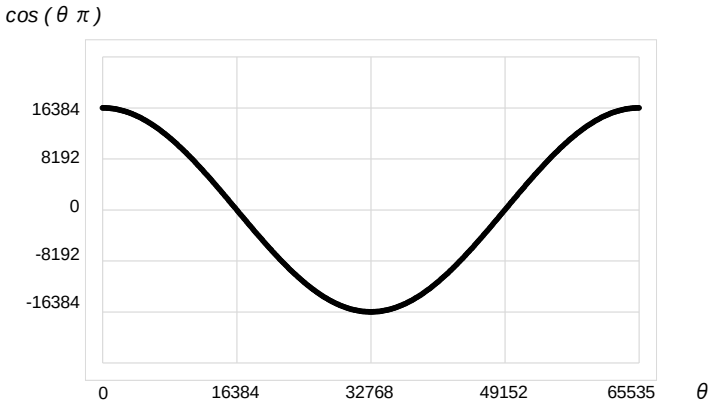
(4) Read the operation result from ADTREG0 register.
ADTREG0: 0000H

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.2 Cosine Operation

Cosine calculation is performed.

Perform the Cosine operation with an approximate expression using series expansion.



In/Out	Register name	Function	Data format
Input	ADTREG0 Note	Phase θ Range: 0 ≤ θ ≤ 65535	Data type: unsigned short
Output	ADTREG0	Cosine value of θ Value = $\cos(\theta\pi)$	Data type: signed short

Note Setting value of ADTREG0 register: (rad × 65536) ÷ 2 π
= (rad × 32768) ÷ π [rad: Input radian angle]

Operation Example:

(1) Setting the ACTLREG register (OPC[4:0] = 00011B (Cosine operation), STM = 1 (Enables software trigger))
ACTLREG = 1AH;

(2) Set the data to be converted to ADTREG0 register.
ADTREG0 = 8000H; Input radian angle = 3.14159

(3) Starts operation by setting OST bit in the ACTLREG register.
ACTLREG = ACTLREG | 01H;

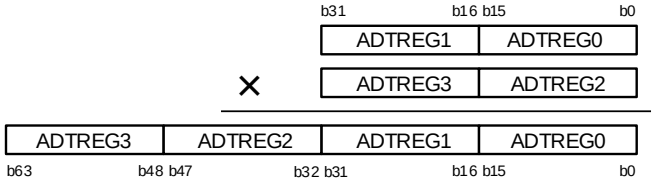
(4) Wait until the OST bit in the ACTLREG register becomes 0.

(5) Read the operation result from ADTREG0 register.
ADTREG0: C000H

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.3 Multiply Operation

32-bit multiplication is performed.



In/Out	Register name	Function	Data format
Input	ADTREG0	Input 32-bit data is generated by concatenation of 2 registers. Multiplier (32 bits) = ADTREG1:ADTREG0 Multiplicand (32 bits) = ADTREG3:ADTREG2	Data type: signed long
	ADTREG1		
	ADTREG2		Data type: signed long
	ADTREG3		
Output	ADTREG0	The 64-bit result is stored in 4 registers. Result (64 bits) = ADTREG3: ADTREG2: ADTREG1: ADTREG0	Data type: signed long long
	ADTREG1		
	ADTREG2		
	ADTREG3		

Operation Example:

(1) Setting the ACTLREG register (OPC[4:0] = 00001B (Multiply operation), STM = 1 (Enables software trigger))
ACTLREG = 0AH;

(2) Set the data for Multiplier (01234567H × 89ABCDEFH)
[ADTREG1, ADTREG0] = 0123_4567H;
[ADTREG3, ADTREG2] = 89AB_CDEFH;

(3) Starts operation by setting OST bit to 1 in the ACTLREG register.
ACTLREG = ACTLREG | 01H;

(4) Wait until the OST bit in the ACTLREG register becomes 0.

(5) Read the operation result from ADTREGi registers.
[ADTREG3, ADTREG2, ADTREG1, ADTREG0] = FF79_5E36_C94E_4629H

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.4 Clarke and Park Transformation (Power invariant transformation)

3-phase-to-2-phase transformation (Clarke transformation) and coordinate axis transformation (Park transformation) are performed sequentially.

Equation is follows.

$$Id = \left(\cos\theta\pi \times \sqrt{\frac{3}{2}} - \sin\theta\pi \times \frac{\sqrt{2}}{2}\right) Iu - \left(\sin\theta\pi \times \sqrt{2}\right) Iw$$

$$Iq = \left(-\sin\theta\ \pi \times \sqrt{\frac{3}{2}} - \cos\theta\pi \times \frac{\sqrt{2}}{2}\right) Iu - \left(\cos\theta\pi \times \sqrt{2}\right) Iw$$

In/Out	Register name	Function	Data format
Input	ADTREG0	u-phase current (Iu)	Data type: signed short
	ADTREG1	w-phase current (Iw)	Data type: signed short
	ADTREG2	Phase θ, Range: 0 ≤ θ ≤ 65535	Data type: unsigned short
Output	ADTREG0	d-axis current (Id)	Data type: signed short
	ADTREG1	q-axis current (Iq)	Data type: signed short

Calculation of Clarke and Park Transformation (Power invariant transformation):

```
// AAU: Clarke & Park transformation (Power invariant transformation)
signed short   sin_buf, cos_buf, temp16_0, temp16_1;
signed long    temp32_0, temp32_1, temp32_2, temp32_3;

sin_buf = AAU_SIN(ADTREG2);              /* -16384 to 16384: -1.0 to 1.0 << 14 */
cos_buf = AAU_COS(ADTREG2);              /* -16384 to 16384: -1.0 to 1.0 << 14 */
temp32_0 = (signed long)ADTREG0 * 20066; /* 20066: sqrt(3/2) << 14 */
temp16_0 = (signed short)(temp32_0 >> 14U);
temp32_0 = (signed long)temp16_0 * (signed long)cos_buf;
temp32_1 = ADTREG0 + ADTREG1;
temp32_1 = temp32_1 + ADTREG1;
temp32_1 = temp32_1 * 23170;             /* 23170: sqrt(2)/2 << 15 */
temp16_1 = (signed short)(temp32_1 >> 15U);
temp32_1 = (signed long)temp16_1 * (signed long)cos_buf;
temp32_1 = -temp32_1;
temp32_2 = (signed long)temp16_1 * (signed long)sin_buf;
ADTREG0 = (unsigned short)((temp32_0 - temp32_2) >> 14U);
temp32_3 = (signed long)temp16_0 * (signed long)sin_buf;
ADTREG1 = (unsigned short)((temp32_1 - temp32_3) >> 14U);
```

Operation Example:

(1) Setting the ACTLREG register (OPC[4:0] = 01000B (Clarke and Park transformation), TMD = 0 (Power invariant transformation), STM = 0 (Enables the last operand write trigger))
ACTLREG = 40H;

(2) Set the data in the following registers.
ADTREG0 = 0000H; u-phase current (Iu: 0)
ADTREG1 = 5A82H; w-phase current (Iw: 23,170)
ADTREG2 = 4000H; Phase θ: 90-degree

(3) Wait until the OST bit in the ACTLREG register becomes 0.

(4) Read the operation result from ADTREG1, 0 register.
ADTREG0 (Id): 8002H (−32,766)
ADTREG1 (Iq): 0000H (0)

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.5 Clarke and Park Transformation (Amplitude invariant transformation)

3-phase-to-2-phase transformation (Clarke transformation) and coordinate axis transformation (Park transformation) are performed sequentially.

Equation is follows.

$$Id = \left(\cos\theta\ \pi\ - \sin\theta\ \pi\ \times \frac{\sqrt{3}}{3}\right) Iu - \left(\sin\theta\ \pi\ \times \frac{2\sqrt{3}}{3}\right) Iw$$

$$Iq = \left(-\sin\theta\ \pi\ - \cos\theta\ \pi\ \times \frac{\sqrt{3}}{3}\right) Iu - \left(\cos\theta\ \pi\ \times \frac{2\sqrt{3}}{3}\right) Iw$$

In/Out	Register name	Function	Data format
Input	ADTREG0	u-phase current (Iu)	Data type: signed short
	ADTREG1	w-phase current (Iw)	Data type: signed short
	ADTREG2	Phase θ, Range: 0 ≤ θ ≤ 65535	Data type: unsigned short
Output	ADTREG0	d-axis current (Id)	Data type: signed short
	ADTREG1	q-axis current (Iq)	Data type: signed short

Calculation of Clarke and Park Transformation (Amplitude invariant transformation):

```
// AAU: Clarke & Park transformation (Amplitude invariant transformation)
signed short   sin_buf, cos_buf, temp16_0, temp16_1;
signed long    temp32_0, temp32_1, temp32_2, temp32_3;

sin_buf = AAU_SIN(ADTREG2);              /* -16384 to 16384: -1.0 to 1.0 << 14 */
cos_buf = AAU_COS(ADTREG2);              /* -16384 to 16384: -1.0 to 1.0 << 14 */
temp16_0 = ADTREG0;
temp32_0 = (signed long)temp16_0 * (signed long)cos_buf;
temp32_1 = ADTREG0 + ADTREG1;
temp32_1 = temp32_1 + ADTREG1;
temp32_1 = temp32_1 * 18919;             /* 18919: sqrt(3)/3 << 15 */
temp16_1 = (signed short)(temp32_1 >> 15U);
temp32_1 = (signed long)temp16_1 * (signed long)cos_buf;
temp32_1 = -temp32_1;
temp32_2 = (signed long)temp16_1 * (signed long)sin_buf;
ADTREG0 = (unsigned short)((temp32_0 - temp32_2) >> 14U);
temp32_3 = (signed long)temp16_0 * (signed long)sin_buf;
ADTREG1 = (unsigned short)((temp32_1 - temp32_3) >> 14U);
```

Operation Example:

(1) Setting the ACTLREG register (OPC[4:0] = 01000B (Clarke and Park transformation), TMD = 1 (Amplitude invariant transformation), STM = 0 (Enables the last operand write trigger))
ACTLREG = 44H;

(2) Set the data in the following registers.
ADTREG0 = 0000H; u-phase current (Iu: 0)
ADTREG1 = 6ED9H; w-phase current (Iw: 28,377)
ADTREG2 = 4000H; Phase θ: 90-degree

(3) Wait until the OST bit in the ACTLREG register becomes 0.

(4) Read the operation result from ADTREG1, 0 register.
ADTREG0 (Id): 8001H (−32,767)
ADTREG1 (Iq): 0000H (0)

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.6 Inverse Park (I-Park) Transformation

The α-axis and β-axis voltages are calculated based on d-axis voltage, q-axis voltage, and θ.

Equation is follows.

$$V\alpha = \left(Vd \times \cos\theta\,\pi - Vq \times \sin\theta\,\pi\right)$$
$$V\beta = \left(Vd \times \sin\theta\,\pi + Vq \times \cos\theta\,\pi\right)$$

In/Out	Register name	Function	Data format
Input	ADTREG0	d-axis voltage (Vd)	Data type: signed short
	ADTREG1	q-axis voltage (Vq)	Data type: signed short
	ADTREG2	Phase θ, Range: 0 ≤ θ ≤ 65535	Data type: unsigned short
Output	ADTREG0	α-axis voltage (Vα)	Data type: signed short
	ADTREG1	β-axis voltage (Vβ)	Data type: signed short

Caution Use input data that does not overflow the calculation results Vα and Vβ.

Calculation of I-Park Transformation:

```
// AAU: I-Park transformation
signed short sin_buf, cos_buf;
signed long temp32_0, temp32_1, temp32_2, temp32_3;

sin_buf = AAU_SIN(ADTREG2);             /* -16384 to 16384: -1.0 to 1.0 << 14 */
cos_buf = AAU_COS(ADTREG2);             /* -16384 to 16384: -1.0 to 1.0 << 14 */
temp32_0 = (signed long)ADTREG0 * (signed long)cos_buf;
temp32_1 = (signed long)ADTREG1 * (signed long)sin_buf;
temp32_2 = (signed long)ADTREG0 * (signed long)sin_buf;
temp32_3 = (signed long)ADTREG1 * (signed long)cos_buf;
ADTREG0 = (usigned short)((temp32_0 - temp32_1) >> 14);
ADTREG1 = (usigned short)((temp32_2 - temp32_3) >> 14);
```

Operation Example:

(1) Setting the ACTLREG register (OPC[4:0] = 01001B (Inverse Park (I-Park) transformation), STM = 0 (Enables the last operand write trigger))
ACTLREG = 48H;

(2) Set the data in the following registers.
ADTREG0 = 0000H; d-axis voltage (Vd: 0)
ADTREG1 = 7FFFH; q-axis voltage (Vq: 32,767)
ADTREG2 = 4000H; Phase θ: 90-degree

(3) Wait until the OST bit in the ACTLREG register becomes 0.

(4) Read the operation result from ADTREG1, 0 register.
ADTREG0 (Vα): 8001H (−32,767)
ADTREG1 (Vβ): 0000H (0)

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.7 Inverse Clarke (I-Clarke) Transformation (Power invariant transformation)

The u-phase, v-phase, and w-phase voltages are calculated based on α-axis voltage, β-axis voltage.

Equation is follows.

$$Vu = \sqrt{\frac{2}{3}}V\alpha$$

$$Vv = \sqrt{\frac{2}{3}}\left(-V\alpha \times \frac{1}{2} + V\beta \times \frac{\sqrt{3}}{2}\right)$$

$$Vw = \sqrt{\frac{2}{3}}\left(-V\alpha \times \frac{1}{2} - V\beta \times \frac{\sqrt{3}}{2}\right)$$

In/Out	Register name	Function	Data format
Input	ADTREG0	α-axis voltage (Vα)	Data type: signed short
	ADTREG1	β-axis voltage (Vβ)	Data type: signed short
Output	ADTREG0	u-phase voltage (Vu)	Data type: signed short
	ADTREG1	v-phase voltage (Vv)	Data type: signed short
	ADTREG2	w-phase voltage (Vw)	Data type: signed short

Calculation of I-Clarke Transformation (Power invariant transformation):

```
// AAU: I-Clarke transformation (Power invariant transformation)
signed long    temp32_0, temp32_1, temp32_2;

temp32_0 = 26755 * ADTREG0;              /* 26755: sqrt(2/3) << 15 */
temp32_1 = 23170 * ADTREG1;              /* 23170: sqrt(2)/2 << 15 */
temp32_2 = -temp32_0 / 2;
ADTREG0 = (unsigned short)(temp32_0 >> 15);
ADTREG1 = (unsigned short)((temp32_2 + temp32_1) >> 15);
ADTREG2 = (unsigned short)((temp32_2 - temp32_1) >> 15);
```

Operation Example:

(1) Setting the ACTLREG register (OPC[4:0] = 01010B (Inverse Clarke (I-Clarke) transformation), TMD = 0 (Power invariant transformation), STM = 0 (Enables the last operand write trigger))
ACTLREG = 50H;

(2) Set the data in the following registers.
ADTREG0 = 7FFFH; α-axis voltage (Vα: 32,767)
ADTREG1 = 0000H; β-axis voltage (Vβ: 0)

(3) Wait until the OST bit in the ACTLREG register becomes 0.

(4) Read the operation result from ADTREG2, 1, 0 register.
ADTREG0 (Vu): 6882H (26,754)
ADTREG1 (Vv): CBBFH (−13,377)
ADTREG2 (Vw): CBBFH (−13,377)

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.8 Inverse Clarke (I-Clarke) Transformation (Amplitude invariant transformation)

The u-phase, v-phase, and w-phase voltages are calculated based on α-axis voltage, β-axis voltage.

Equation is follows.

$$Vu = V\alpha$$
$$Vv = \left(-V\alpha \times \frac{1}{2} + V\beta \times \frac{\sqrt{3}}{2}\right)$$
$$Vw = \left(-V\alpha \times \frac{1}{2} - V\beta \times \frac{\sqrt{3}}{2}\right)$$

In/Out	Register name	Function	Data format
Input	ADTREG0	α-axis voltage (Vα)	Data type: signed short
	ADTREG1	β-axis voltage (Vβ)	Data type: signed short
Output	ADTREG0	u-phase voltage (Vu)	Data type: signed short
	ADTREG1	v-phase voltage (Vv)	Data type: signed short
	ADTREG2	w-phase voltage (Vw)	Data type: signed short

Calculation of I-Clarke Transformation (Amplitude invariant transformation):

```
// AAU: I-Clarke transformation (Amplitude invariant transformation)
signed long    temp32_0, temp32_1, temp32_2;

temp32_0 = 21845 * ADTREG0;              /* 21845: 2/3       << 15 */
temp32_1 = 18919 * ADTREG1;              /* 18919: sqrt(3)/3 << 15 */
temp32_2 = -temp32_0 / 2;
ADTREG0 = (uint16)(temp32_0 >> 15);
ADTREG1 = (uint16)((temp32_2 + temp32_1) >> 15);
ADTREG2 = (uint16)((temp32_2 - temp32_1) >> 15);
```

Operation Example:

(1) Setting the ACTLREG register (OPC[4:0] = 01010B (Inverse Clarke (I-Clarke) transformation), TMD = 1 (Amplitude invariant transformation, STM = 0 (Enables the last operand write trigger))
ACTLREG = 54H;

(2) Set the data in the following registers.
ADTREG0 = 7FFFH; α-axis voltage (Vα: 32,767)
ADTREG1 = 0000H; β-axis voltage (Vβ: 0)

(3) Wait until the OST bit in the ACTLREG register becomes 0.

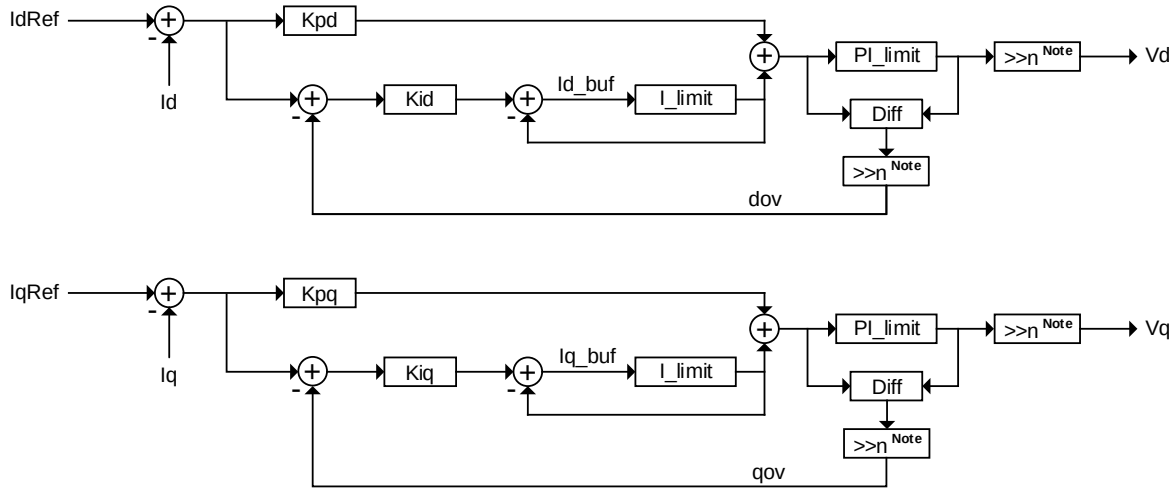
(4) Read the operation result from ADTREG2, 1, 0 register.
ADTREG0 (Vu): 7FFFH (32,767)
ADTREG1 (Vv): C001H (−16,383)
ADTREG2 (Vw): C001H (−16,383)

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.9 PI Control for Motor

PI control corresponds d-axis and q-axis independently, and controls the output with limitation function.

Operation flow diagram is follows.



Note The value n is depend on the value of AKRAG register.

In/Out	Register name	Function	Data format
Input	ADTREG0	d-axis current (Id)	Data type: signed short
	ADTREG1	q-axis current (Iq)	Data type: signed short
	AIDREF	d-axis reference current (IdRef)	Data type: signed short
	AIQREF	q-axis reference current (IqRef)	Data type: signed short
	AKPD	d-axis proportional coefficient (Kpd)	Data type: signed short
	AKID	d-axis integral coefficient (Kid)	Data type: signed short
	AKPQ	q-axis proportional coefficient (Kpq)	Data type: signed short
	AKIQ	q-axis integral coefficient (Kiq)	Data type: signed short
	AILIM	I (integral) control current limitation (I_limit)	Data type: signed short
	APILIM	PI control output voltage limitation (PI_limit)	Data type: signed short
	AIDBFH: AIDBFL	d-axis current buffer (id_buf)	Data type: singed long
	AIQBFH: AIQBFL	q-axis current buffer (iq_buf)	Data type: singed long
	ADOVER	d-axis limitation current over (dov)	Data type: signed short
	AQOVER	q-axis limitation current over (qov)	Data type: signed short
	AKRAG	PI control coefficient range setting	Data type: unsigned short
Output	ADTREG0	d-axis voltage (Vd)	Data type: signed short
	ADTREG1	q-axis voltage (Vq)	Data type: signed short

Cautions 1. Set the following registers before executing the operation.
AIDREF, AIQREF, AKPD, AKPQ, AKIQ, AILIM, and APILIM
2. The following registers are used internally during PI control for motor arithmetic.
AIDBFL, AIDBFH, AIQBFL, AIQBFH, ADOVER, and AQOVER
3. The following registers are used internally as the following values:
[AIDBFH, AIDBFL], [AIQBFH, AIQBFL]: The limit value is amplified by the AKRAG setting.
(AKRAG = 00H: AILIM × 16, = 01H: AILIM × 256, = 02H: AILIM × 4096, = 03H: AILIM × 65536)

Calculation of PI control for motor:

```
// AAU: PI control for motor
signed short   err_buf, p_buf;
signed long    i_buf, pi_buf, i_limit, pi_limit;

err_buf = AIDREF - ADTREG0;
p_buf = (signed long)AKPD * (signed long)err_buf;
i_buf = (signed long)(err_buf - ADOVER) * (signed long)AKID;
i_buf = i_buf + ((signed long)AIDBFH << 16) + (signed long)AIDBFL;
// < integral limitation >
i_limit = (signed long)AILIM << (4 * (AKRAG+1));
if (i_limit < i_buf) {
    i_buf = i_limit;
}
else if (-i_limit > i_buf) {
    i_buf = -i_limit;
}
AIDBFH = (unsigned short)(i_buf >> 16);
AIDBFL = (unsigned short)i_buf;
pi_buf = i_buf + p_buf;
// < pi limitation >
pi_limit = (signed long)APILIM << (4 * (AKRAG+1));
if (pi_limit < pi_buf) {
    ADOVER = (unsigned short)((pi_buf - pi_limit) >> (4 * (AKRAG+1)));
    pi_buf = pi_limit;
}
else if (-pi_limit > pi_buf) {
    ADOVER = (unsigned short)((pi_buf + pi_limit) >> (4 * (AKRAG+1)));
    pi_buf = -pi_limit;
}
else {
    ADOVER = 0;
}
ADTREG0 = (unsigned short)(pi_buf >> (4 * (AKRAG+1)));
err_buf = AIQREF - ADTREG1;
p_buf   = (signed long)AKPQ * (signed long)err_buf;
i_buf   = (signed long)(err_buf - AQOVER) * (signed long)AKIQ;
i_buf   = i_buf + ((signed long)AIQBFH << 16) + (signed long)AIQBFL;
// < integral limitation >
if (i_limit < i_buf) {
    i_buf = i_limit;
}
else if (-i_limit > i_buf) {
    i_buf = -i_limit;
}
AIQBFH = (unsigned short)(i_buf >> 16);
AIQBFL = (unsigned short)i_buf;

// AAU: PI control for motor (Continuation from the previous page)
pi_buf = i_buf + p_buf;

// Continued on the next page
```

```
// < PI limitation >
if (pi_limit < pi_buf) {
    AQOVER = (unsigned short)((pi_buf - pi_limit) >> (4 * (AKRAG+1)));
    pi_buf = pi_limit;
}
else if (-pi_limit > pi_buf) {
    AQOVER = (unsigned short)((pi_buf + pi_limit) >> (4 * (AKRAG+1)));
    pi_buf = -pi_limit;
}
else {
    AQOVER = 0;
}
ADTREG1 = (unsigned short)(pi_buf >> (4 * (AKRAG+1)));
```

Operation Example:

First setting:

(1) PI control for motor function requires preparation for register setting
AIDREF = 0000H; d-axis reference current (IdRef: 0)
AIQREF = 1000H; q-axis reference current (IqRef: 4096)
AKRAG = 02H; PI control coefficient range setting (coefficient range: $1/2^{12}$)
AKPD = 1000H; d-axis proportional coefficient (Kpd = 1.0, AKPD = Kpd × 2^{12})
AKID =0800H; d-axis integral coefficient (Kid = 0.5, AKID = Kid × 2^{12})
AKPQ = 1333H; q-axis proportional coefficient (Kpq = 1.2, AKPQ = Kpq × 2^{12})
AKIQ = 0999H; q-axis integral coefficient (Kiq = 0.6, AKIQ = Kiq × 2^{12})
AILIM = 4000H; I (integral) control current limitation (I_limit)
APILIM = 6000H; PI control output voltage limitation (PI_limit)
[AIDBFH, AIDBFL] = 00000000H; d-axis current buffer (id_buf: zero-initialize)
[AIQBFH, AIQBFL] = 00000000H; q-axis current buffer (iq_buf: zero-initialize)
ADOVER = 0000H; d-axis limitation current over (dov: zero-initialize)
AQOVER = 0000H; q-axis limitation current over (qov: zero-initialize)

(2) Setting the ACTLREG register (OPC[4:0] = 01011B (PI control for motor), STM = 0 (Enables the last operand write trigger))
ACTLREG = 58H;

(3) Set the motor feedback current in registers ADTREG0 and ADTREG1.
ADTREG0 = 0000H; d-axis current (Id)
ADTREG1 = 0000H; q-axis current (Iq)

(4) Wait until the OST bit in the ACTLREG register becomes 0.

(5) Read the operation result from ADTREG1, 0 register.
ADTREG0: 0000H (Vd: 0)
ADTREG1: 1CCCH (Vq: 7,372)

Second and subsequent settings: (When the AAU related registers have not been changed since the first time)

(1) Setting the ACTLREG register.

(2) Set the motor feedback current in registers ADTREG0 and ADTREG1 (Start of operation).

(3) Wait until the OST bit int the ACTLREG register becomes 0.

(4) Read the operation result from ADTREG1, 0 register.

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.10 Clarke & Park Transformation and PI Control for Motor (Power invariant transformation)

The Clarke and Park transformation (see **28.3.2.4**) and PI control for motor operation (see **28.3.2.9**) are performed sequentially.

In/Out	Register name	Function	Data format
Input	ADTREG0	u-phase current (Iu)	Data type: signed short
	ADTREG1	w-phase current (Iw)	Data type: signed short
	ADTREG2	Phase θ, Range: 0 ≤ θ ≤ 65535	Data type: unsigned short
	AIDREF	d-axis reference current (IdRef)	Data type: signed short
	AIQREF	q-axis reference current (IqRef)	Data type: signed short
	AKPD	d-axis proportional coefficient (Kpd)	Data type: signed short
	AKID	d-axis integral coefficient (Kid)	Data type: signed short
	AKPQ	q-axis proportional coefficient (Kpq)	Data type: signed short
	AKIQ	q-axis integral coefficient (Kiq)	Data type: signed short
	AILIM	I (integral) control current limitation (I_limit)	Data type: signed short
	APILIM	PI control output voltage limitation (PI_limit)	Data type: signed short
	AIDBFH: AIDBFL	d-axis current buffer (id_buf)	Data type: signed long
	AIQBFH: AIQBFL	q-axis current buffer (iq_buf)	Data type: signed long
	ADOVER	d-axis limitation current over (dov)	Data type: signed short
	AQOVER	q-axis limitation current over (qov)	Data type: signed short
	AKRAG	PI control coefficient range setting	Data type: unsigned short
Output	ADTREG0	d-axis voltage (Vd)	Data type: signed short
	ADTREG1	q-axis voltage (Vq)	Data type: signed short
	ADTREG2	d-axis current (Id)	Data type: signed short
	ADTREG3	q-axis current (Iq)	Data type: signed short

Cautions 1. Set the following registers before executing the operation.
AIDREF, AIQREF, AKPD, AKPQ, AKIQ, AILIM, and APILIM
2. The following registers are used internally during PI control for motor arithmetic.
AIDBFL, AIDBFH, AIQBFL, AIQBFH, ADOVER, and AQOVER
3. The following registers are used internally as the following values:
[AIDBFH, AIDBFL], [AIQBFH, AIQBFL]: The limit value is amplified by the AKRAG setting.
(AKRAG = 00H: AILIM × 16, = 01H: AILIM × 256, = 02H: AILIM × 4096, = 03H: AILIM × 65536)

Calculation of Clarke & Park Transformation and PI Control for Motor (Power invariant transformation):

For details of calculation, refer to section **28.3.2.4** Clarke and park transformation and **28.3.2.9** PI control for motor.

Operation Example:

First setting:

(1) PI control for motor function requires preparation for register setting.
AIDREF = 0000H; d-axis reference current (IdRef: 0)
AIQREF = 0000H; q-axis reference current (IqRef: 0)
AKRAG = 02H; PI control coefficient range setting (coefficient range: $1/2^{12}$)
AKPD = 0010H; d-axis proportional coefficient (Kpd = 0.00391, AKPD = Kpd × 2^{12})
AKID =0001H; d-axis integral coefficient (Kid = 0.00025, AKID = Kid × 2^{12})
AKPQ = 0020H; q-axis proportional coefficient (Kpq = 0.00782, AKPQ = Kpq × 2^{12})
AKIQ = 0002H; q-axis integral coefficient (Kiq = 0.0005, AKIQ = Kiq × 2^{12})
AILIM = 4000H; I (integral) control current limitation (I_limit)
APILIM = 6000H; PI control output voltage limitation (PI_limit)
[AIDBFH, AIDBFL] = 00000000H; d-axis current buffer (id_buf: zero-initialize)
[AIQBFH, AIQBFL] = 00000000H; q-axis current buffer (iq_buf: zero-initialize)
ADOVER = 0000H; d-axis limitation current over (dov: zero-initialize)
AQOVER = 0000H; q-axis limitation current over (qov: zero-initialize)

(2) Setting the ACTLREG register (OPC[4:0] = 01100B (PI control for motor), TMD = 0 (Power invariant transformation), STM = 0 (Enables the last operand write trigger))
ACTLREG = 60H;

(3) Set the motor feedback current in registers ADTREG0 and ADTREG1.
ADTREG0 = FB3DH; u-phase current (Iu)
ADTREG1 = 11D8H; w-phase current (Iw)
ADTREG2 = D555H; Phase θ: 300-degree

(4) Wait until the OST bit in the ACTLREG register becomes 0.

(5) Read the operation result from ADTREG registers.
ADTREG0: FFEEH (Vd: −18)
ADTREG1: 0021H (Vq: 33)
ADTREG2: 1026H (Id: 4,132)
ADTREG3: F025H (Iq: −4,059)

Second and subsequent settings: (When the AAU related registers have not been changed since the first time)

(1) Setting the ACTLREG register.

(2) Set the motor feedback current in registers ADTREG0, ADTREG1, and ADTREG2 (Start of operation).

(3) Wait until the OST bit int the ACTLREG register becomes 0.

(4) Read the operation result from ADTREG registers.

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.11 Clarke & Park Transformation and PI Control for Motor (Amplitude invariant transformation)

The Clarke and Park transformation (see **28.3.2.5**) and PI control for motor operation (see **28.3.2.9**) are performed sequentially.

In/Out	Register name	Function	Data format
Input	ADTREG0	u-phase current (Iu)	Data type: signed short
	ADTREG1	w-phase current (Iw)	Data type: signed short
	ADTREG2	Phase θ, Range: 0 ≤ θ ≤ 65535	Data type: unsigned short
	AIDREF	d-axis reference current (IdRef)	Data type: signed short
	AIQREF	q-axis reference current (IqRef)	Data type: signed short
	AKPD	d-axis proportional coefficient (Kpd)	Data type: signed short
	AKID	d-axis integral coefficient (Kid)	Data type: signed short
	AKPQ	q-axis proportional coefficient (Kpq)	Data type: signed short
	AKIQ	q-axis integral coefficient (Kiq)	Data type: signed short
	AILIM	I (integral) control current limitation (I_limit)	Data type: signed short
	APILIM	PI control output voltage limitation (PI_limit)	Data type: signed short
	AIDBFH: AIDBFL	d-axis current buffer (id_buf)	Data type: signed long
	AIQBFH: AIQBFL	q-axis current buffer (iq_buf)	Data type: signed long
	ADOVER	d-axis limitation current over (dov)	Data type: signed short
	AQOVER	q-axis limitation current over (qov)	Data type: signed short
	AKRAG	PI control coefficient range setting	Data type: unsigned short
Output	ADTREG0	d-axis voltage (Vd)	Data type: signed short
	ADTREG1	q-axis voltage (Vq)	Data type: signed short
	ADTREG2	d-axis current (Id)	Data type: signed short
	ADTREG3	q-axis current (Iq)	Data type: signed short

Cautions 1. Set the following registers before executing the operation.
AIDREF, AIQREF, AKPD, AKPQ, AKIQ, AILIM, and APILIM
2. The following registers are used internally during PI control for motor arithmetic.
AIDBFL, AIDBFH, AIQBFL, AIQBFH, ADOVER, and AQOVER
3. The following registers are used internally as the following values:
[AIDBFH, AIDBFL], [AIQBFH, AIQBFL]: The limit value is amplified by the AKRAG setting.
(AKRAG = 00H: AILIM × 16, = 01H: AILIM × 256, = 02H: AILIM × 4096, = 03H: AILIM × 65536)

Calculation of Clarke & Park Transformation and PI Control for Motor (Amplitude invariant transformation):

For details of calculation, refer to section **28.3.2.5** Clarke and park transformation and **28.3.2.9** PI control for motor.

Operation Example:

First setting:

(1) PI control for motor function requires preparation for register setting.
AIDREF = 0000H; d-axis reference current (IdRef: 0)
AIQREF = 0000H; q-axis reference current (IqRef: 0)
AKRAG = 00H; PI control coefficient range setting (coefficient range: $1/2^4$)
AKPD = 0018H; d-axis proportional coefficient (Kpd = 1.5, AKPD = Kpd × 2^4)
AKID =0003H; d-axis integral coefficient (Kid = 0.1875, AKID = Kid × 2^4)
AKPQ = 0028H; q-axis proportional coefficient (Kpq = 2.5, AKPQ = Kpq × 2^4)
AKIQ = 0004H; q-axis integral coefficient (Kiq = 0.25, AKIQ = Kiq × 2^4)
AILIM = 4000H; I (integral) control current limitation (I_limit)
APILIM = 6000H; PI control output voltage limitation (PI_limit)
[AIDBFH, AIDBFL] = 00000000H; d-axis current buffer (id_buf: zero-initialize)
[AIQBFH, AIQBFL] = 00000000H; q-axis current buffer (iq_buf: zero-initialize)
ADOVER = 0000H; d-axis limitation current over (dov: zero-initialize)
AQOVER = 0000H; q-axis limitation current over (qov: zero-initialize)

(2) Setting the ACTLREG register (OPC[4:0] = 01100B (PI control for motor), TMD = 1 (Amplitude invariant transformation), STM = 0 (Enables the last operand write trigger))
ACTLREG = 64H;

(3) Set the motor feedback current in registers ADTREG0 and ADTREG1.
ADTREG0 = 03E8H; u-phase current (Iu)
ADTREG1 = FAAAH; w-phase current (Iw)
ADTREG2 = 0000H; Phase θ: 0-degree

(4) Wait until the OST bit in the ACTLREG register becomes 0.

(5) Read the operation result from ADTREG registers.
ADTREG0: F968H (Vd: −1,688)
ADTREG1: F542H (Vq: −2,750)
ADTREG2: 03E8H (Id: 1,000)
ADTREG3: 03E8H (Iq: 1,000)

Second and subsequent settings: (When the AAU related registers have not been changed since the first time)

(1) Setting the ACTLREG register.

(2) Set the motor feedback current in registers ADTREG0, ADTREG1, and ADTREG2 (Start of operation).

(3) Wait until the OST bit int the ACTLREG register becomes 0.

(4) Read the operation result from ADTREG registers.

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.12 I-Park & I-Clarke Transformation (Power invariant transformation)

The I-Park transformation (see **28.3.2.6**) and the I-Clarke transformation (see **28.3.2.7**) are performed sequentially.

In/Out	Register name	Function	Data format
Input	ADTREG0	d-axis voltage (Vd)	Data type: signed short
	ADTREG1	q-axis voltage (Vq)	Data type: signed short
	ADTREG2	Phase θ, Range: 0 ≤ θ ≤ 65535	Data type: unsigned short
Output	ADTREG0	u-phase voltage (Vu)	Data type: signed short
	ADTREG1	v-phase voltage (Vv)	Data type: signed short
	ADTREG2	w-phase voltage (Vw)	Data type: signed short

Calculation of I-Park & I-Clarke Transformation (Power invariant transformation):

For details of calculation, refer to section **28.3.2.6** I-Park transformation and **28.3.2.7** I-Clarke transformation.

Operation Example:

(1) Setting the ACTLREG register (OPC[4:0] = 01101B (I-Park & I-Clarke transformation), TMD = 0 (Power invariant transformation), STM = 0 (Enables the last operand write trigger))
ACTLREG = 68H;

(2) Set the data in the following registers.
ADTREG0 = 1000H; d-axis voltage (Vd: 4,096)
ADTREG1 = 1000H; q-axis voltage (Vq: 4,096)
ADTREG2 = 2000H; Phase θ: 45-degree

(3) Wait until the OST bit in the ACTLREG register becomes 0.

(4) Read the operation result from ADTREG2, 1, 0 register.
ADTREG0: 0000H (Vu: 0)
ADTREG1: 0FFFH (Vv: 4,095)
ADTREG2: F000H (Vw: −4,096)

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.13 I-Park & I-Clarke Transformation (Amplitude invariant transformation)

The I-Park transformation (see **28.3.2.6**) and the I-Clarke transformation (see **28.3.2.8**) are performed sequentially.

In/Out	Register name	Function	Data format
Input	ADTREG0	d-axis voltage (Vd)	Data type: signed short
	ADTREG1	q-axis voltage (Vq)	Data type: signed short
	ADTREG2	Phase θ, Range: 0 ≤ θ ≤ 65535	Data type: unsigned short
Output	ADTREG0	u-phase voltage (Vu)	Data type: signed short
	ADTREG1	v-phase voltage (Vv)	Data type: signed short
	ADTREG2	w-phase voltage (Vw)	Data type: signed short

Calculation of I-Park & I-Clarke Transformation (Amplitude invariant transformation):

For details of calculation, refer to section **28.3.2.6** I-Park transformation and **28.3.2.8** I-Clarke transformation.

Operation Example:

(1) Setting the ACTLREG register (OPC[4:0] = 01101B (I-Park & I-Clarke transformation)), TMD = 1 (Amplitude invariant transformation), STM = 0 (Enables the last operand write trigger))
ACTLREG = 6CH;
(2) Set the data in the following registers.
ADTREG0 = 1000H; d-axis voltage (Vd: 4,096)
ADTREG1 = 1000H; q-axis voltage (Vq: 4,096)
ADTREG2 = C000H; Phase θ: 270-degree
(3) Wait until the OST bit in the ACTLREG register becomes 0.
(4) Read the operation result from ADTREG2, 1, 0 register.
ADTREG0: 1000H (Vu: 4,096)
ADTREG1: EA24H (Vv: −5,596)
ADTREG2: 05DBH (Vw: 1,499)

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.3.2.14 PI Control for DC/DC Converter Control

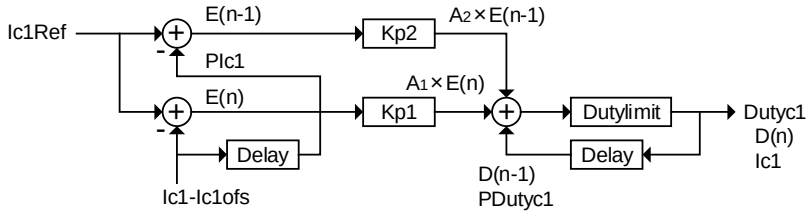
PI control corresponds channel 1, channel 2, and channel 3 independently, and controls the output with limitation function.
An offset cancel of current input value is possible for each channel operation.
PI control algorithm of this operation is follows.

$$D(n) = D(n-1) + A_1 \times E(n) + A_2 \times E(n-1)$$

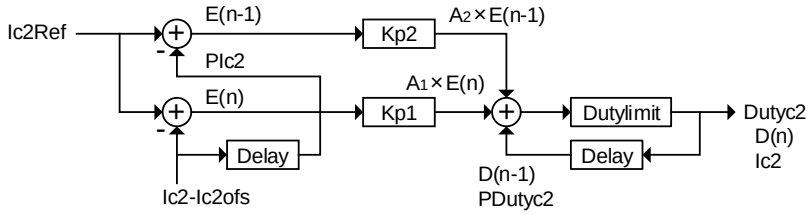
- D (n): Current output duty
- D (n-1): Last current output duty
- E (n): Current error value = current reference value – measurement current value
- E (n-1): Last error value = current reference value – last measurement current value
- A_1, A_2: Coefficient
- Remark n: Sampling count number

Operation flow diagram is follows.

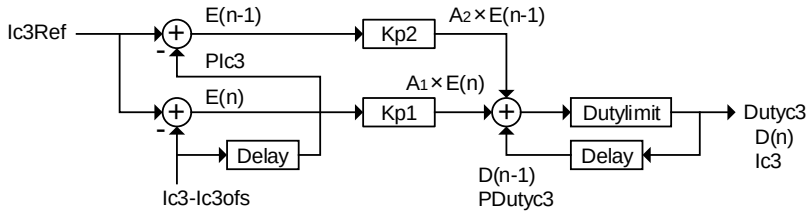
i) Channel 1



ii) Channel 2



iii) Channel 3



- **Channel 1 input/output register table**

In/Out	Register name	Function	Settable Range	Data format
Input	ADTREG0	Channel 1 current (Ic1)	0000H to FFFFH	Data type: unsigned short
	AL1REF	Channel 1 current reference (Ic1Ref)		
	AL1OFS	Channel 1 current offset (Ic1ofs)		
	ADUTYL1	Channel 1 previous duty (PDutyc1)		
	AIPL1	Channel 1 previous current (PIc1)		
Output	ADTREG0	Channel 1 duty (Dutyc1)	0000H to FFFFH	Data type: unsigned short
Input (common)	AKI1	Proportional coefficient 1 (Kp1)	0000H to FFFFH	Data type: signed short
	AKI2	Proportional coefficient 2 (Kp2)		
	ADUTYMX	Current limit duty (Dutylimit)	00H to FFH	Data type: unsigned char

- **Channel 2 input/output register table**

In/Out	Register name	Function	Settable Range	Data format
Input	ADTREG1	Channel 2 current (Ic2)	0000H to FFFFH	Data type: unsigned short
	AL2REF	Channel 2 current reference (Ic2Ref)		
	AL2OFS	Channel 2 current offset (Ic2ofs)		
	ADUTYL2	Channel 2 previous duty (PDutyc2)		
	AIPL2	Channel 2 previous current (PIc2)		
Output	ADTREG1	Channel 2 duty (Dutyc2)	0000H to FFFFH	Data type: unsigned short
Input (common)	AKI1	Proportional coefficient 1 (Kp1)	0000H to FFFFH	Data type: signed short
	AKI2	Proportional coefficient 2 (Kp2)		
	ADUTYMX	Current limit duty (Dutylimit)	00H to FFH	Data type: unsigned char

- **Channel 3 input/output register table**

In/Out	Register name	Function	Settable Range	Data format
Input	ADTREG2	Channel 3 current (Ic3)	0000H to FFFFH	Data type: unsigned short
	AL3REF	Channel 3 current reference (Ic3Ref)		
	AL3OFS	Channel 3 current offset (Ic3ofs)		
	ADUTYL3	Channel 3 previous duty (PDutyc3)		
	AIPL3	Channel 3 previous current (PIc3)		
Output	ADTREG2	Channel 3 duty (Dutyc3)	0000H to FFFFH	Data type: unsigned short
Input (common)	AKI1	Proportional coefficient 1 (Kp1)	0000H to FFFFH	Data type: signed short
	AKI2	Proportional coefficient 2 (Kp2)		
	ADUTYMX	Current limit duty (Dutylimit)	00H to FFH	Data type: unsigned char

Cautions 1. Registers AKI1, AKI2, and ADUTYMX are used in common for channels 1 to 3.

2. The ADUTYMX register is used as a value multiplied by 256 for internal comparison.

3. If the value calculated internally overflows, the value set in the ADUTYMX register is output as a result (bits 15 to 8: Value of the ADUTYMX register, bits 7 to 0: 00H).

4. The previous registers (ADUTYLx, AIPLx [x = 1, 2, 3]) does not need to be set from the second operation. Because the data is set automatically after calculation is completed.

Calculation of PI Control for DC/DC Converter Control:

```
// AAU: PI control for DC/DC converter (Channel.1)
signed long    temp32_0, temp32_1;

temp32_0 = (signed long)(AL1REF – AIPL1) * (signed long)AKI2;
temp32_1 = (signed long)(ADTREG0 – AL1OFS);
temp32_0 = temp32_0 + ((signed long)AL1REF – temp32_1) * (signed long)AKI1;
temp32_0 = temp32_0 + (signed long)ADUTYL1;
// < Overflow/underflow check >
if (temp32_0 > (signed long)(ADUTYMX * 256)) {
    temp32_0 = (signed long)(ADUTYMX * 256);
}
else if (temp32_0 < 0) {
    temp32_0 = 0;
}
AIPL1 = (unsigned short)temp32_1;
ADTREG0 = (unsigned short)temp32_0;
```

Operation Example (Channel 1 calculation):

First setting:

(1) Setting the ACTLREG register (OPC[4:0] = 10000B (PI control for DC/DC converter control channel 1), STM = 0 (Enables the last operand write trigger))
ACTLREG = 80H;

(2) Set data in each register (AL1REF/AL1OFS/AKI1/AKI2/ADUTYMX/AIPL1)
AL1REF = Channel 1 current reference value;
AL1OFS = Channel 1 current offset value;
AKI1/AKI2 = Proportional coefficient (Kp1, Kp2) value;
ADUTYMX = Current limit duty value;
AIPL1 = Channel 1 previous current value (e.g. Latest A/D data minus offset value (AL1OFS));

(3) Set the ADTREG0 register (Start of operation)
ADTREG0 = Channel 1 current data;

(4) Wait until the OST bit in the ACTLREG register becomes 0.

(5) Read the result from ADTREG0 register
[Bits 15 to 8 in the ADTREG0 register: Next PWM duty value]
[Bits 7 to 4 in the ADTREG0 register: Value after the decimal point (e.g. Used as a PWM dithering counter value)]
If the AAU related registers are changed before the next operation, the value of AIPL1 register is stored.

Second and subsequent settings: (When the AAU related registers have not been changed since the first time)

(1) Setting the ACTLREG register

(2) Setting the ADTREG0 register (Start of operation)

(3) Wait until the OST bit int the ACTLREG register becomes 0.

(4) Read the result from ADTREG0 register
[Bits 15 to 8 in the ADTREG0 register: Next PWM duty value]
[Bits 7 to 4 in the ADTREG0 register: Value after the decimal point]

Caution Be sure to set the bits AAUPAGE1 and AAUPAGE0 of the AAUWINR register to the corresponding values before accessing the AAU-related registers.

28.4 Notes on Application Accelerator Unit

(1) Rewriting the ADTREGi (i = 0 to 3) is prohibited when the application accelerator unit is in operation (when the OST bit in the ACTLREG register = 1). In this case, the operation results are not guaranteed.

(2) Rewriting the ACTLREG is prohibited when the application accelerator unit is in operation (when the OST bit in the ACTLREG register = 1). In this case, the operation results are not guaranteed.

(3) Before execution of the STOP instruction, the operation of the application accelerator unit must be completed. If the STOP instruction is executed when the application accelerator unit is under operation, the operation results are not guaranteed.

CHAPTER 29 FUNCTIONAL SAFETY

29.1 Overview of Functional Safety

The following safety mechanisms are provided in the RL78/F22 and RL78/F25 to comply with the IEC60730, and ISO26262 (ASIL B) 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/F22 and RL78/F25 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-purpose CRC: This can be used for checking various data in addition to the code flash memory area while the CPU is running.

(2) Internal RAM-ECC function

2-bit error detection and 1- bit error correction are available.

(3) CAN RAM-ECC function (RL78/F25 Only)

2-bit error detection and 1- bit error correction are available.

(4) Code Flash memory ECC function

2-bit error detection and 1- bit error correction are available.

(5) CPU stack pointer monitor function

This detects underflows and overflows of the stack pointer.

(6) Clock monitoring function

The system clock (f_{MAIN}) and main/PLL select clock (f_{MP}) are monitored to detect oscillation stopping.

(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 reference voltage and perform detection of open mode fault for analog input pins.

- Self-diagnosis of 12-bit A/D converter
  Analog conversion level can be selected from Zero code test (AV_{REFM}), Full code test (AV_{REFP}) or 1/2 code test ($AV_{REFP} \times 1/2$).
- Disconnection detection assist function (precharge or discharge)
  Built in precharge and discharge function to be able to assist disconnection detection of the wire which is connected to analog inputs.

(10) Watchdog timer function

The watchdog timer function is used to detect an inadvertent program, such as unintended program loops. If a program loop is detected, an internal reset signal is generated.

29.2 Registers Used by Functional Safety

The functional safety uses the following registers:

Table 29-1. List of Registers Used for Functional Safety

Address	Register Name	Symbol	After Reset	Access Size	Each Function of Functional Safety
F02F0H	Flash memory CRC control register	CRC0CTL	00H	1, 8	Flash memory CRC operation function (high-speed CRC)
F02F2H	Flash memory CRC operation result register	PGCRCL	0000H	16	
FFFACH	CRC input register	CRCIN	00H	8	CRC operation function (general-purpose CRC)
F02F9H	CRC operation mode control register	CRCMD	00H	8	
F02FAH	CRC data register	CRCD	0000H	16	
F0200H	Error address store register	ERADR	0000H	16	Internal RAM-ECC function
F0202H	1-bit error detection interrupt enable register	ECCIER	00H	8	
F0203H	Bit error detection register	ECCER	00H	8	
F0204H	ECC test protect register	ECCTPR	00H	8	
F0205H	ECC test mode register	ECCTMDR	00H	8	
F0206H	Write data inversion register	ECCDWRVR	0000H	16	
F07C0H	CAN RAM-ECC control register	CFDECCTLs	0010H	8, 16	CAN RAM-ECC function
F07C1H		CFDECCTLsL / CFDECCTLsH			
F07C4H	CAN RAM-ECC test mode control register	CFDECTMCs	0000H	8, 16	
F07C5H		CFDECTMCsL / CFDECTMCsH			
F07C8H	CAN RAM-ECC decoder input ECC bit replacement test register	CFDECERDBs	00H	8	
F07CAH	CAN RAM-ECC redundant bit test register	CFDECHORDs	00H	8	
F07CBH	CAN RAM-ECC syndrome test register	CFDECSYNDs	00H	8	
F07CCH	CAN RAM-ECC decoder input data replacement test register L	CFDECTEDsL	0000H	16	
F07CEH	CAN RAM-ECC decoder input data replacement test register H	CFDECTEDsH	0000H	16	
F07D0H	CAN RAM-ECC error address register	CFDECEADs	0000H	16	
F0098H	Code flash bit error detection function control register	CFERRCTLR	40H	8	Code flash memory ECC function
F0099H	Code flash bit error detection function status register	CFERRSTR	0x000000B	8	
F009CH	Code flash bit error detection address register L	ERRADRL	xxxxH	16	
F009EH	Code flash bit error detection address register H	ERRADRH	0xH	8	
F00CCH	Flash write buffer register L	FLWL	0000H	16	
F00CEH	Flash write buffer register H	FLWH	0000H	16	
FFFC6H	Flash ECC write buffer register	FLWE	00H	8	
F00D8H	SPM control register	SPMCTRL	00H	8	Stack pointer monitor function
F00DAH	SP overflow address setting register	SPOFR	FFFEH	16	
F00DCH	SP underflow address setting register	SPUFR	0000H	16	
F02CEH	Clock monitor test register	CLMTES	00H	1, 8	Clock monitor function
F0078H	Invalid memory access detection control register	IAWCTL	00H	8	Invalid memory access detection function
F0074H	Timer input select register 0	TIS0	00H	8	Frequency detection function
FFFA4H	System clock control register	CKC	00H	1, 8	
F06BEH	A/D control expansion register [page.0]	ADCER	0000H	16	A/D test function
F06BEH	A/D self-test mode data register [page.1]	ADRD	0000H	16	
F06BAH	A/D disconnection detection control register [page.7]	ADDISCR	00H	8	
FFFABH	Watchdog timer enable register	WDTE	1AH/ 9AH Note	8	Watchdog timer function

Note The WDTE register reset value differs depending on the WDTON bit in the user option byte (000C0H/040C0H).

Remark s = 0, 1

29.3 Operation of Functional Safety

29.3.1 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/F22 and RL78/F25 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 410 µs when the operating clock is at 40 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 29-1. 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)
0	1	0	0	0	0	00000H to 43FFBH (272 Kbytes − 4 bytes)
0	1	0	0	0	1	00000H to 47FFBH (288 Kbytes − 4 bytes)
0	1	0	0	1	0	00000H to 4BFFBH (304 Kbytes − 4 bytes)
0	1	0	0	1	1	00000H to 4FFFBH (320 Kbytes − 4 bytes)
0	1	0	1	0	0	00000H to 53FFBH (336 Kbytes − 4 bytes)
0	1	0	1	0	1	00000H to 57FFBH (352 Kbytes − 4 bytes)
0	1	0	1	1	0	00000H to 5BFFBH (368 Kbytes − 4 bytes)
0	1	0	1	1	1	00000H to 5FFFBH (384 Kbytes − 4 bytes)
0	1	1	0	0	0	00000H to 63FFBH (400 Kbytes − 4 bytes)
0	1	1	0	0	1	00000H to 67FFBH (416 Kbytes − 4 bytes)
0	1	1	0	1	0	00000H to 6BFFBH (432 Kbytes − 4 bytes)
0	1	1	0	1	1	00000H to 6FFFBH (448 Kbytes − 4 bytes)
0	1	1	1	0	0	00000H to 73FFBH (464 Kbytes − 4 bytes)
0	1	1	1	0	1	00000H to 77FFBH (480 Kbytes − 4 bytes)
0	1	1	1	1	0	00000H to 7BFFBH (496 Kbytes − 4 bytes)
0	1	1	1	1	1	00000H to 7FFFBH (512 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 29-2. 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

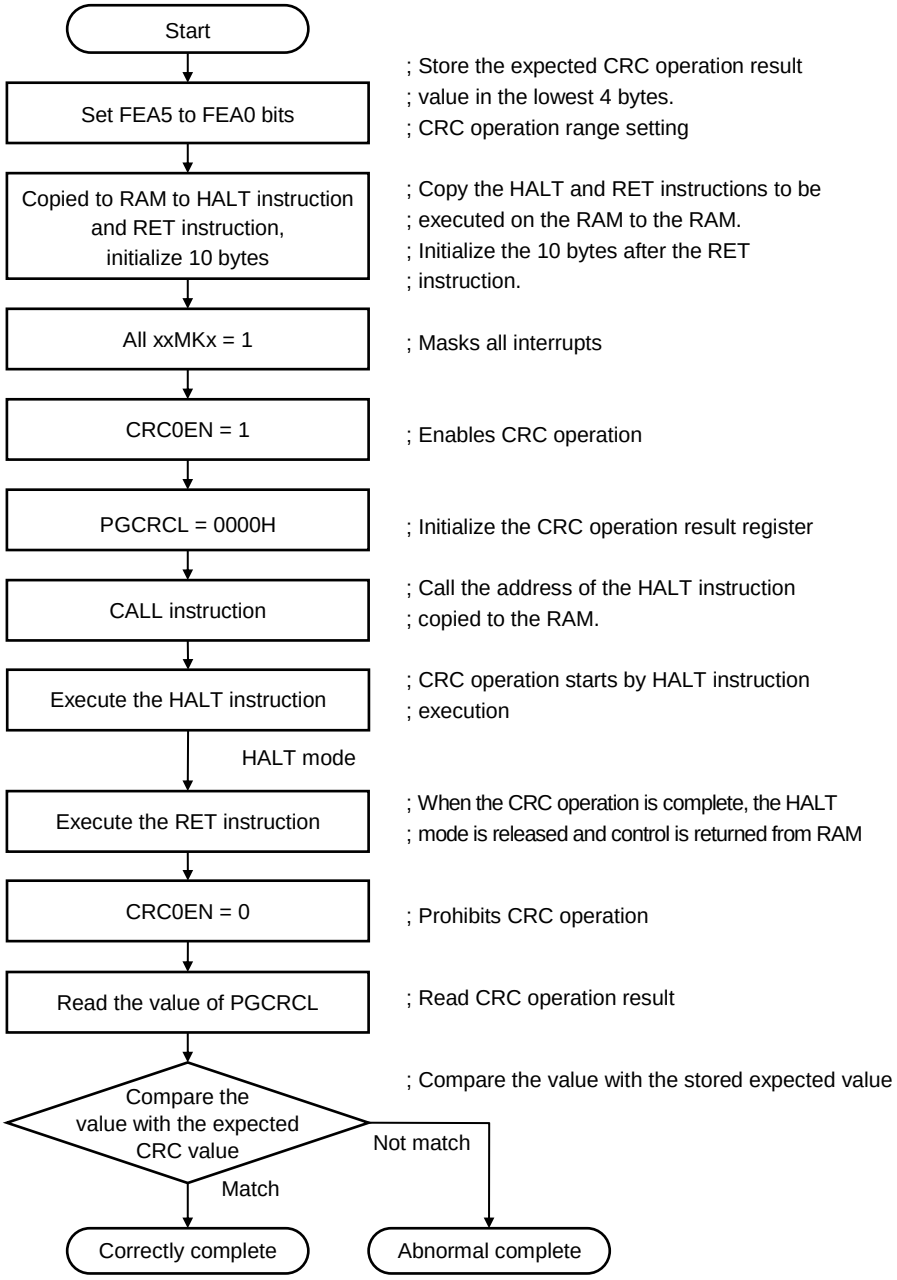
PGCRC15 to PGCRC0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 29-3 shows the flowchart of flash memory CRC operation function (high-speed CRC).

<Operation Flow>

Figure 29-3. Flowchart of Flash Memory CRC Operation Function (High-speed CRC)



Cautions 1. The CRC operation is executed only on the code flash memory.

2. Store the expected CRC operation value in the area below the operation range in the code flash memory.

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.

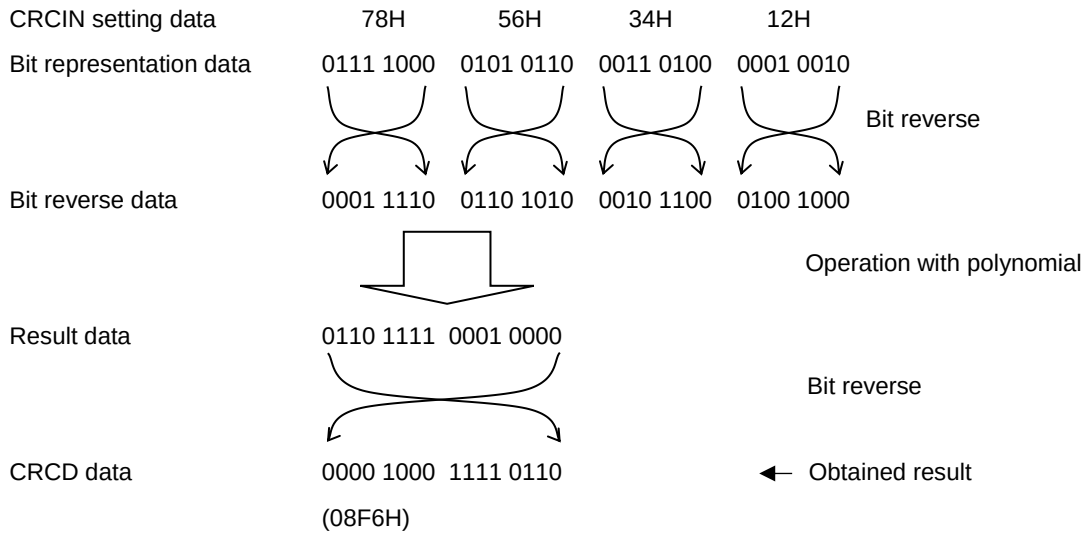
The expected CRC value can be calculated by using tools such as the CS+ Integrated Development Environment (see the **CS+ User's Manual**: CC-RL Build Tool Operation for details).

29.3.2 CRC Operation Function (General-purpose CRC)

In the RL78/F22 and RL78/F25, 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 29-4. Format of CRC Input Register (CRCIN)

Address: FFFACH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRCIN	CRCIN7	CRCIN6	CRCIN5	CRCIN4	CRCIN3	CRCIN2	CRCIN1	CRCIN0

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 29-5. 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	0	0	0	0	0	0	0	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 29-6. 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	CRCD[15:0]															

CRCD[15:0]	Store the general-purpose CRC operation result Notes 1, 2
0000H to FFFFH	CRC operation result when supporting CRC-CCITT
0000H to 000FH	CRC operation result when conforming to SENT Note 3

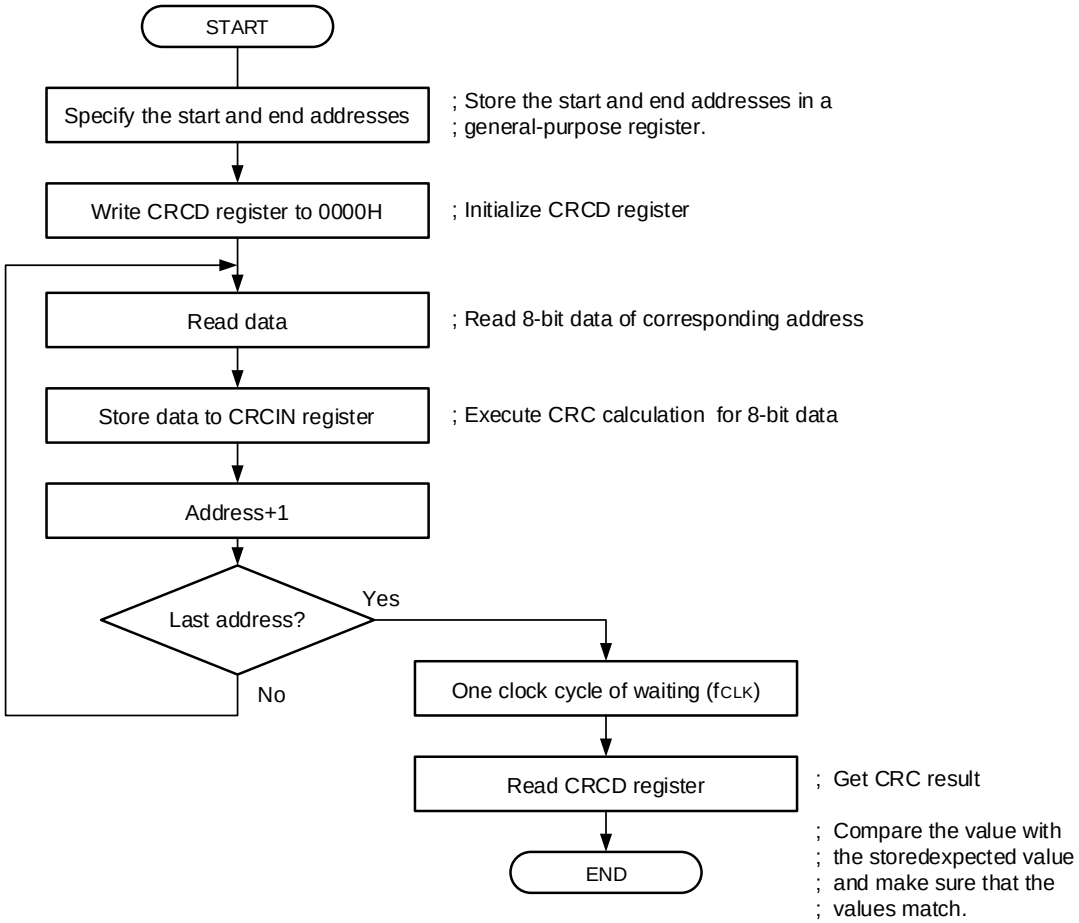
Notes 1. Read the value written to CRCD register before writing to CRCIN register.

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 29-7. CRC Operation Function (General-Purpose CRC)



29.3.3 Internal RAM-ECC Function

The RL78/F22 and RL78/F25 have the Internal 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 Internal 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 29-8. Format of Error Address Store Register (ERADR)

Address: F0200H After reset: 0000H R

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	The lower 16-bit of the address when a bit error interrupt request is generated. Example: In the case of FED0H, a bit error occurred when reading the RAM at address FFED0H.

Cautions 1. The ERADR register should be accessed with a 16-bit memory manipulation instruction.
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 29-9. 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	0	0	0	0	0	0	0	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.
2. INTRAM interrupt request occurs regardless of the value of ECCIER on two bits error.

(3) Bit Error Detection Register (ECCER)

Figure 29-10. Format of Bit Error Detection Register (ECCER)

Address: F0203H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ECCER	0	0	0	0	0	0	0	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 29-11. Format of ECC Test Protect Register (ECCTPR)

Address: F0204H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ECCTPR	0	0	0	0	0	TPR2	TPR1	TPR0

TRP2	TRP1	TRP0	ECC test protect bits
1	1	1	Access to the ECCTMDR register is enabled.
Other than above			Access to the ECCTMDR register is disabled.

(5) ECC Test Mode Register (ECCTMDR)

Figure 29-12. Format of ECC Test Mode Register (ECCTMDR)

Address: F0205H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ECCTMDR	0	0	0	0	0	TMD2	TMD1	TMD0

TMD2	TMD1	TMD0	ECC test mode bits
0	0	0	Normal operating mode
0	0	1	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 29-13. Format of Write Data Inversion Register (ECCDWRVR)

Address: F0206H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
ECCDWRVR	0	0	0	PRTYRV	ECCRV3	ECCRV2	ECCRV1	ECCRV0

7	6	5	4	3	2	1	0
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. The ECCDWRVR register can be set by a 16-bit memory manipulation instruction.

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 area, 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 on-chip debug mode.

<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.

If RAM fetch area being not initialized, there is a possibility that an ECC error occurs by pre-reading instruction code.

If the RAM is used without being initialized, ECC errors may occur by pre-reading even in areas not used in user program.

<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 area. 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.

29.3.4 CAN RAM-ECC Function (RL78/F25 Only)

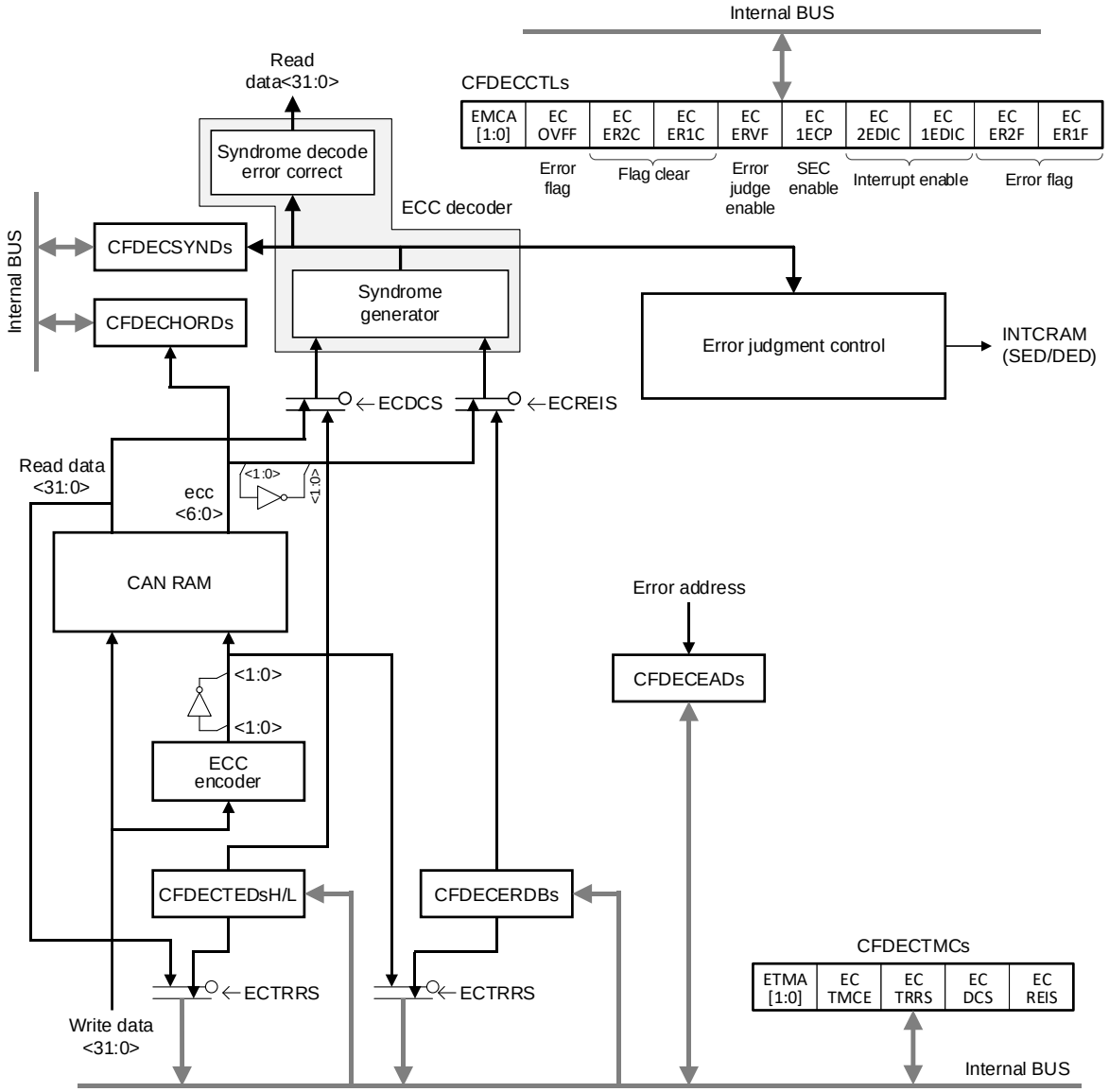
The RL78/F25 has the CAN RAM-ECC functions as listed below. In regard to SEC, SED, DED and interrupt of SED and DED, disabling or enabling is selectable. These functions can be tested by using ECC test mode.

- Single-bit error correction (SEC)
- Single-bit error detection (SED)
- Double-bit error detection (DED)
- Error address capturing (CFDECEADs register)
- Error address overflow (OVF)
- Maskable interrupt (INTCRAM)

<Configuration>

Figure 29-14 shows a block diagram of the CAN RAM-ECC function.

Figure 29-14. Block Diagram of CAN RAM-ECC function



Remark s: Channel number (s = 0, 1)

<Control Register>

Register Name		Description	Access Size
8-bit	16-bit		
CFDECCTLsL	CFDECCTLs	CAN RAM-ECC Control Register	8 bits/16 bits Note
CFDECCTLsH			
CFDECTMCsL	CFDECTMCs	CAN RAM-ECC Test Mode Control Register	8 bits/16 bits Note
CFDECTMCsH			
–	CFDECTEDsH/ CFDECTEDsL	CAN RAM-ECC Decoder Input Data Replacement Test Register	16 bits
CFDECSYNDs	–	CAN RAM-ECC Syndrome Test Register	8 bits
CFDECHORDs	–	CAN RAM-ECC Redundant Bit Test Register	8 bits
CFDECERDBs	–	CAN RAM-ECC Decoder input ECC Bit Replacement Test Register	8 bits
–	CFDECEADs	CAN RAM-ECC Error Address Register	16 bits

Note Set the bit 6 of CFDECCTLs and the bit 7 of CFDECTMCs by a 16-bit memory manipulation instruction.

Remark s: Channel number (s = 0, 1)
Channel number s is selected by CSEL0 bit of the CFDWINR register.

(1) CAN RAM-ECC Control Register (CFDECCTLs) [s=0,1]

This register is used to control the CAN RAM-ECC functions. This register can be read and written by a 16-bit or 8-bit manipulation instruction.

Figure 29-15. Format of CAN RAM-ECC Control Register (CFDECCTLs)

Address: CFDECCTLs: F07C0H
CFDECCTLsL: F07C0H, CFDECCTLsH: F07C1H
After reset: 0010H R/W

Symbol	15	14	13 Note 1	12 Note 1	11 Note 1	10	9	8 Note 1
CFDECCTLsH	EMCA1	EMCA0	0	0	ECOVFF	ECER2C	ECER1C	0

Symbol	7	6	5	4	3	2 Note 1	1 Note 1	0 Note 1
CFDECCTLsL	0	ECERVF	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	0

EMCA1	EMCA0	Access control of ECERVF bit
0	1	Write access enabled.
Other than above		Write access disabled.
These bits enable or disable writing to bit 6. The read value is always 00B.		

ECOVFF	Flag bit of ECC overflow detection
0	Overflow is not occurred after reset or clearing ECER2F and ECER1F.
1	Error address register overflowed.
This flag is set when 1-bit or 2-bit error is detected under the condition that the error address has been already captured in CFDECEADs register and the error address is not same as captured address or error address is same as captured address but error type is not same. One of the following clears this flag: ・An internal or external reset ・Writing 1 to ECER2C or ECER1C or ECERVF bit	

ECER2C	Flag clear bit of ECC 2-bit error detection
0	No operation.
1	ECER2F bit clear.
This bit clears the bit 2 (ECER2F) status flag. The read value is always 0 and writing 0 to ECER2C does not change the state. If writing 1 to ECER2C conflicts with the condition for setting bit 2, the former takes priority.	

ECER1C	Flag clear bit of ECC 1-bit error detection
0	No operation.
1	ECER1F bit clear.
This bit clears the bit 1 (ECER1F) status flag. The read value is always 0 and writing 0 to ECER1C does not change the internal state. If writing 1 to ECER1C conflicts with the condition for setting bit 1, the former takes priority.	

Bit 7	Reserved
0	The write value must be 0. The read value is 0.

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

ECERVF	Control bit to enable ECC error judgment Note 2
0	Error judgement disabled.
1	Error judgement enabled.
ECERVF enables or disables ECC error judgement. Write access to ECERVF is enabled when the value of bits 15 and 14 is 01B.	

EC1ECP	Control bit of ECC 1-bit error correction Note 3
0	Enable 1-bit error correction upon error detection.
1	Disable 1-bit error correction upon error detection.

EC2EDIC	Control bit of interrupt at ECC 2-bit error detection
0	When a 2-bit error is detected, interrupt disabled.
1	When a 2-bit error is detected, interrupt enabled.

EC1EDIC	Control bit of interrupt at ECC 1-bit error detection
0	When a 1-bit error is detected, interrupt disabled.
1	When a 1-bit error is detected, interrupt enabled.

ECER2F	Flag bit of ECC 2-bit error detection Note 4
0	A 2-bit error has not occurred since this bit was cleared.
1	A 2- bit error has occurred.
This bit indicates that a 2-bit error has been detected in bits 0 to 38 of the data read from the CAN RAM while error detection is enabled. This bit is a read only flag. Clearing conditions: (1) Reset is applied. (2) 1 is written to ECER2C or ECERVF.	

ECER1F	Flag bit of ECC 1-bit error detection Note 5
0	A 1-bit error has not occurred since this bit was cleared.
1	A 1- bit error has occurred.
This bit indicates that a 1-bit error has been detected in bits 0 to 38 of the data read from the CAN RAM while error detection is enabled. This bit is a read only flag. Clearing conditions: (1) Reset is applied. (2) 1 is written to ECER1C or ECERVF.	

Notes 1. Bits 0 to 2, 8, and 11 to 13 are read-only.

2. Writing to this bit is valid only by a 16-bit memory manipulation instruction.

3. This bit does not affect GERFLHs.EEF. For details of GERFLHs.EEF, see **Section 19.3.14** of **CHAPTER 19 CAN INTERFACE (RS-CANFD lite)**.

4. After the CAN RAM initialization, clearing this bit is recommended. The same state as ECER2F is also indicated in GERFLHs.EEF. For details of GERFLHs.EEF, see **Section 19.3.14** of **CHAPTER 19 CAN INTERFACE (RS-CANFD lite)**.

5. After the CAN RAM initialization, clearing this bit is recommended.

Caution **When the CAN RAM is in initial state, do not do error judgement. It is recommended to enable error judgement after CAN RAM initialization by setting ECERVF bit to 1.**

Remark s: Channel number (s = 0, 1)
Channel number s is selected by CSEL0 bit of the CFDWINR register.

(2) CAN RAM-ECC Test Mode Control Register (CFDECTMCs) [s=0,1]

This register is used to control the CAN RAM-ECC test function. This register can be read and written by a 16-bit or 8-bit manipulation instruction.

Figure 29-16. Format of CAN RAM-ECC Test Mode Control Register (CFDECTMCs)

Address: CFDECTMCs: F07C4H
CFDECTMCsL: F07C4H, CFDECTMCsH: F07C5H

After reset: 0000H R/W

Symbol	15	14	13 Note 1	12 Note 1	11 Note 1	10 Note 1	9 Note 1	8 Note 1
CFDECTMCsH	ETMA1	ETMA0	0	0	0	0	0	0

Symbol	7	6 Note 1	5 Note 1	4	3	2	1	0
CFDECTMCsL	ECTMCE	0	0	ECTRRS	0	0	ECDCS	ECREIS

ETMA1	ETMA0	Access control of ECTMCE bit
1	0	Write access enabled.
Other than above		Write access disabled.
These bits enable or disable writing to bit 7. The read value is always 00B.		

ECTMCE	Enable bit of ECC test mode control Note 2
0	Disable access to the test registers and test control bits.
1	Enable access to the test registers and test control bits.
ECTMCE enables or disables access to the test registers and test control bits. Write access to ETCMCE is enabled when the value of bits 15 and 14 is 10B.	

ECTRRS	Select bit of test register read data
0	When CFDECTEDsH / CFDECTEDsL is read, register is read. When CFDECERDBs is read, register is read.
1	When CFDECTEDsH / CFDECTEDsL is read, CAN RAM output data is read. When CFDECERDBs is read, CAN RAM-ECC input is read.
By the setting of ECTRRS bit, reading source data of CFDECTEDsH / CFDECTEDsL and CFDECERDBs registers can be selected. Write access to ECTRRS is enabled only when ECTMCE is 1 (can be set simultaneously). ECTRRS is cleared by writing 0 to ECTMCE (can be cleared synchronously).	

Bit3,2	Reserved
0	The write value must be 0. The read value is 0.

ECDCS	Select bit of ECC decoder input data
0	Select lower 32-bit CAN RAM output data to be input to the decoder.
1	Select CFDECTEDsH / CFDECTEDsL to be input to the decoder.
ECDCS selects either the lower 32-bit data from the CAN RAM or the internal test register (CFDECTEDsH / CFDECTEDsL) as the lower 32-bit data of the input signal to be decoded. Write access to ECDCS is enabled only when ECTMCE is 1 (can be set simultaneously). ECDCS is cleared by writing 0 to ECTMCE (can be cleared synchronously).	

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

ECREIS	Select bit of ECC decoder input ECC redundant bits
0	Select upper 7-bit CAN RAM output data to be input to the decoder.
1	Select CFDECERDBs to be input to the decoder.
ECREIS selects the input signal to be decoded from the upper 7-bit data from the CAN RAM (redundant bit area) or the value set in the internal test registers (CFDECERDBs). Write access to ECREIS is enabled only when ECTMCE is 1 (can be set simultaneously). ECREIS is cleared by writing 0 to ECTMCE (can be cleared synchronously).	

Notes 1. Bits 5, 6, and 8 to 13 are read-only.

2. Writing to this bit is valid only by a 16-bit memory manipulation instruction.

Remark s: Channel number (s = 0, 1)

Channel number s is selected by CSEL0 bit of the CFDWINR register.

(3) CAN RAM-ECC Decoder Input Data Replacement Test Register (CFDECTEDsH / CFDECTEDsL) [s=0,1]

This register is used for ECC function test. In test mode (the ECTMCE bit in the CFDECTMCs register is set to 1), the data of this register is selected as ECC decoder input. When the ECTMCE bit is clear to 0, this register is always read as 0. This register can be read and written by a 16-bit manipulation instruction.

Figure 29-17. Format of CAN RAM-ECC Decoder Input Data Replacement Test Register (CFDECTEDsH / CFDECTEDsL)

Address: F07CEH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
CFDECTEDsH	ECEDBH[15:8]							

7	6	5	4	3	2	1	0
ECEDBH[7:0]							

Address: F07CCH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
CFDECTEDsL	ECEDBL[15:8]							

7	6	5	4	3	2	1	0
ECEDBL[7:0]							

ECEDBH / ECEDBL	Decoder input data for ECC test mode
0000H to FFFFH	When the ECDCS bit of the CFDECTMCs register is 1, ECEDBH[15:0] bits and ECEDBL[15:0] bits are input to the decoder. When the ECTRRS bit of the CFDECTMCs register is 1, reading ECEDBH[15:0]/ECEDBL[15:0] bits return the CAN RAM output data instead of the data written to the ECEDBH[15:0] /ECEDBL[15:0] bits.

Remarks 1. Changing the ECTMCE bit of the CFDECTMCs register from 1 to 0 resets CFDECTEDsH/ CFDECTEDsL synchronously.

2. s: Channel number (s = 0, 1)
Channel number s is selected by CSEL0 bit of the CFDWINR register.

(4) CAN RAM-ECC Syndrome Test Register (CFDECSYNDs) [s=0,1]

This register is read only register to confirm the syndrome code generated in decode circuit when ECC test mode is enabled by setting the ECTMCE bit of the CFDECTMCs register to 1. This register can be read by an 8-bit manipulation instruction.

Figure 29-18. Format of CAN RAM-ECC Syndrome Test Register (CFDECSYNDs)

Address: F07CBH After reset: 00H R

Symbol	7 Note	6	5	4	3	2	1	0
CFDECSYNDs	0	SYND[6:0]						

SYND[6:0]	ECC Decode Syndrome Data for ECC test mode
00H to 7FH	The value of syndrome code (SYND[6:0]) generated from input data in decode circuit is readout as this register's bits. SYND bits are not in the holding circuit. So, input signal changes, then this register's value changes, too. CFDECSYNDs is valid only when ECTMCE = 1, and is always read as 00H when ECTMCE = 0.

Note Bit 7 is reserved. When read, the value after reset is returned.

Remarks 1. Changing the ECTMCE bit of the CFDECTMCs register from 1 to 0 resets CFDECSYNDs synchronously.

2. s: Channel number (s = 0, 1)
Channel number s is selected by CSEL0 bit of the CFDWINR register.

(5) CAN RAM-ECC Redundant Bit Test Register (CFDECHORDs) [s=0,1]

This register holds the ECC redundant 7-bit data (upper 7-bit of CAN RAM data) when ECC test mode is enabled by setting the ECTMCE bit of the CFDECTMCs register to 1. This register can be read by an 8-bit manipulation instruction.

Figure 29-19. Format of CAN RAM-ECC Redundant Bit Test Register (CFDECHORDs)

Address: F07CAH After reset: 00H R

Symbol	7 Note	6	5	4	3	2	1	0
CFDECHORDs	0	HORD[6:0]						

HORD[6:0]	ECC Redundant bit hold data for ECC test mode
00H to 7FH	The upper 7-bit of CAN RAM output (ECC redundant bits) are captured when the following 2 cases. - When a peripheral read access to the CAN RAM is performed during ECC test mode (CFDECTMCs.ECTMCE = 1). - When CFDECTEDsL register is read while CFDECTMCs.ECTMCE and CFDECTMCs.ECTRRS is set to 1. CFDECHORDs is valid only when ECTMCE=1, and is always read as 00H when ECTMCE=0.

Note Bit 7 is reserved. When read, the value after reset is returned.

Remarks 1. Changing the ECTMCE bit of the CFDECTMCs register from 1 to 0 resets CFDECHORDs synchronously.

2. s: Channel number (s = 0, 1)
Channel number s is selected by CSEL0 bit of the CFDWINR register.

(6) CAN RAM-ECC Decoder Input ECC Bit Replacement Test Register (CFDECERDBs) [s=0,1]

This register is a substitution buffer register to be able to ECC error injection to the ECC decoder. ECC redundant 7-bit data can be set to this register when in the ECC test mode (the ECTRRS bit of the CFDECTMCs register is 1). This register can be read and written only when ECC test mode.

This register can be read and written by an 8-bit manipulation instruction.

Figure 29-20. Format of CAN RAM-ECC Decoder Input ECC Bit Replacement Test Register (CFDECERDBs)

Address: F07C8H After reset: 00H R/W

Symbol	7 Note	6	5	4	3	2	1	0
CFDECERDBs	0	ERDB[6:0]						

ERDB	ECC Redundant bit error injection data for ECC test mode
00H to 7FH	When the ECREIS bit of the CFDECTMCs register is 1, ERDB[6:0] bits value, instead of the upper seven data bits to be input to the decoding circuit, is handled by the decoding circuit. When the ECTRRS bit of the CFDECTMCs register is 1, reading CFDECERDBs register returns the signal value supplied to CAN RAM instead of the data written to CFDECERDBs register.

Note Bit 7 is reserved. When read, the value after reset is returned. When writing, write the value after reset.

Remarks 1. Changing the ECTMCE bit of the CFDECTMCs register from 1 to 0 resets CFDECERDBs synchronously.

2. s: Channel number (s = 0, 1)
Channel number s is selected by CSEL0 bit of the CFDWINR register.

(7) CAN RAM-ECC Error Address Register (CFDECEADs) [s=0,1]

This register is a read-only register used to hold the address at which an ECC error has occurred.

If an ECC error is detected while ECC error detection is enabled, the CAN RAM address is latched using the detection signal as a trigger, and the address is stored in CFDECEADs register as the address at which the ECC error occurred. But the error address is not captured when the error occurred again if error address and error type (SED or DED) are same as the captured one.

The address is stored upon detection of the first ECC error while no error status is set. However, only when 2-bit error is detected after 1-bit error detection, the address of second 2-bit error will be overwritten to this register regardless of the address is same or not.

Only one address can be held in this register.

This register can be read by a 16-bit manipulation instruction.

Figure 29-21. Format of CAN RAM-ECC Error Address Register (CFDECEADs)

Address: F07D0H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
CFDECEADs	0	0	0	0	0	0	ECEAD[9:8]	

7	6	5	4	3	2	1	0
ECEAD[7:0]							

ECEAD	ECC error address
0000H to 03FFH	This register is cleared by clearing the ECER2F or ECER1F bit of CAN RAM-ECC control register (CFDECCTLs).

Remark s: Channel number (s = 0, 1)
Channel number s is selected by CSEL0 bit of the CFDWINR register.

<Bit Error Detection Interrupt>

The ECC 1-bit error detection interrupt control bit (the EC1EDIC bit of the CFDECCTLs register) can be used to specify whether to output or not an interrupt request signal when the bit error is 1 bit. The ECC 2-bit error detection interrupt control bit (the EC2EDIC bit of the CFDECCTLs register) can be used to specify whether to output or not an interrupt request signal when the bit error is 2-bit. When any bit error is detected, an interrupt request signal (INTCRAM) is generated, and the address of the bit error is held in the ECC error address register (CFDECEADs). If the bit error is 1 bit, the bit error detection flag (the ECER1F bit of the CFDECCTLs register) is set to 1. If the bit error is 2-bit, the bit error detection flag (the ECER2F bit of the CFDECCTLs register) is set to 1. When a 2-bit error is detected, the application shall be transferred to safe state.

<Bit Error Detection During CPU Write Operation>

When writing to the RAM area, a write access from CPU with a size of 8 or 16 bits is done then the RS-CANFD lite module does a read modify write access to the CAN RAM, because the CAN RAM requires a 32-bit access via the ECC module. Therefore, by the write access from the CPU with a size of 8 or 16 bits, if a bit corruption is included in read data, an ECC error occurs due to the read access of read modify write from the RS-CANFD lite module.

<Bit Error Detection During CAN RAM Initialization>

The initial value of the CAN RAM is undefined. Therefore, at the read access including read modify write from the RS-CANFD lite to the CAN RAM before finishing the CAN RAM initialization, ECC bit error may occur. Hence, it is recommended that error judgement is not enabled (the ECERVF bit of the CFDECCTLs register is 0) until CAN RAM initialization is completed.

<Error Address>

The error address is stored upon detection of the first ECC error while no error status is set. When a 2-bit error of different address is detected under the condition that error address captured by 1-bit error, error address register is holds 2-bit error address and bits ECER2F and ECOVFF of the CFDECCTLs register are set. When 2-bit error of same address is detected under the condition that error address captured by 1-bit error, ECER2F bit and ECOVFF bit are set and same address is overwritten.

The error address register keeps the previous address until cleared by ECER2F or ECER1F bit of the CFDECCTLs register clearing.

Note The error address of the CAN RAM access will be captured in the capture register which is part of the ECC module (not RS-CANFD lite module). Since the capture register holds the internal address, the captured CAN RAM address will not match with the SFR address.

RAM address configuration and calculation of CAN RAM test parameters (Pn, r):

If ECC error occurs when CPU is reading RX message buffer ID register 3 (**RMID3**), then the SFR address of **RMID3** (RPAGE=10B) is F0504H. However, the CAN RAM Address for the same is 270H (if **RMNB.RMPLS** = 011B), refer to **Section 19.13.1** of **CHAPTER 19 CAN INTERFACE (RS-CANFD lite)**. Based on the CAN RAM address given by the ECC module when an ECC error occurred the related buffer can be determined with the information given in **Section 19.13** of **CHAPTER 19 CAN INTERFACE (RS-CANFD lite)**. As example the configuration given at **Figure 19-48** is considered. Since the data of each RAM is 32 bits, a RAM address is an address of 4 bytes boundaries. Therefore, as for the address of an ECC error, the address of 4 bytes boundaries is shown. When the ECC module returns the address 17CH (380) as address where the ECC error occurred, then the error occurred at COM FIFO.

In order to access the ECC error RAM address by the CAN RAM test mode, the related page number (Pn) which is defined by the **GTSTCFGs.RTMPS[3:0]** and **RPGACCr** register index (r) can be calculated as below.

- Pn = floor(address / (256/4)) = floor(380 / (256/4)) = 5
- r = mod(address / (256/4)) = mod(380 / (256/4)) = 60

Regarding the details of CAN RAM test mode, **GTSTCFGs** register and **RPGACCrHs/RPGACCrLs**, see **Section 19.3.48**, **19.3.52**, and **19.11.2.1** of **CHAPTER 19 CAN INTERFACE (RS-CANFD lite)**.

The relation between CAN RAM address, Pn, r, and SFR address is as follows.

Figure 29-22. Relationship Between CAN RAM Address, Pn, r, and SFR Address

GTSTCFGs.RTMPS[3:0]: RAM Test Page Access

Pn	Register Index: r	Byte Address	Word Address		SFR Address
8	0 - 5			AFL	
7	6 - 63	718H	1C6H		F0420H
7	0 - 5			PFL	
6	52 - 63	6D0H	1B4H		F0520H
6	0 - 51				
5	0 - 63			C FIFO	F04B8H
4	0 - 63			RX FIFO	F046CH
3	0 - 63			RX MB	F0420H
2	4 - 63	210H	084H		
2	0 - 3			OTB	-
1	28 - 63	170H	05CH		
1	12 - 27	130H	04CH	THL	F0640H
1	0 - 11			TX MB	
0	0 - 63	000H	000H		F0504H

Remarks AFL: AFL rule table area
PFL: PNF rule table area
C FIFO: Common FIFO buffer area
RX FIFO: RX FIFO buffer area
RX MB: RX buffer area
OTB: OTB area (Temporary area)
THL: TX history list buffer area
TX MB: TX buffer area

<ECC Test Flow>

The main test method of the CAN RAM-ECC module diagnosis is described below.

(1) Enable ECC error judgement by setting ECERVF bit of CFDECCTLs register to 1.
(2) Enable CAN RAM-ECC test mode by setting ECTMCE bit of CFDECTMCs register to 1.
(3) Set ECDCS bit and ECREIS bit of CFDECTMCs register to 1.
(4) Set EC2EDIC bit and EC1EDIC bit of CFDECCTLs register to 1 and EC1ECP bit of CFDECCTLs register to 0.
(5) Write test data word (32 bits) teo CFDECTEDsH/CFDECTEDsL registers and write test ECC code (7 bits) to CFDECERDBs register.
(6) Read data from CAN RAM by reading RPGACCrHs/Ls registers.
(7) Check expected error status and error address by reading CFDECCTLs and CFDECEADs registers.
(8) Check CAN RAM read data if 1-bit error is corrected when 1-bit error is injected as test data.
(9) Clear error flag ECER2F bit or ECER1F bit of CFDECCTLs register.
(10) Set ECTMCE bit of CFDECTMCs register to 0 to set normal mode.

29.3.5 Code Flash Memory ECC Function

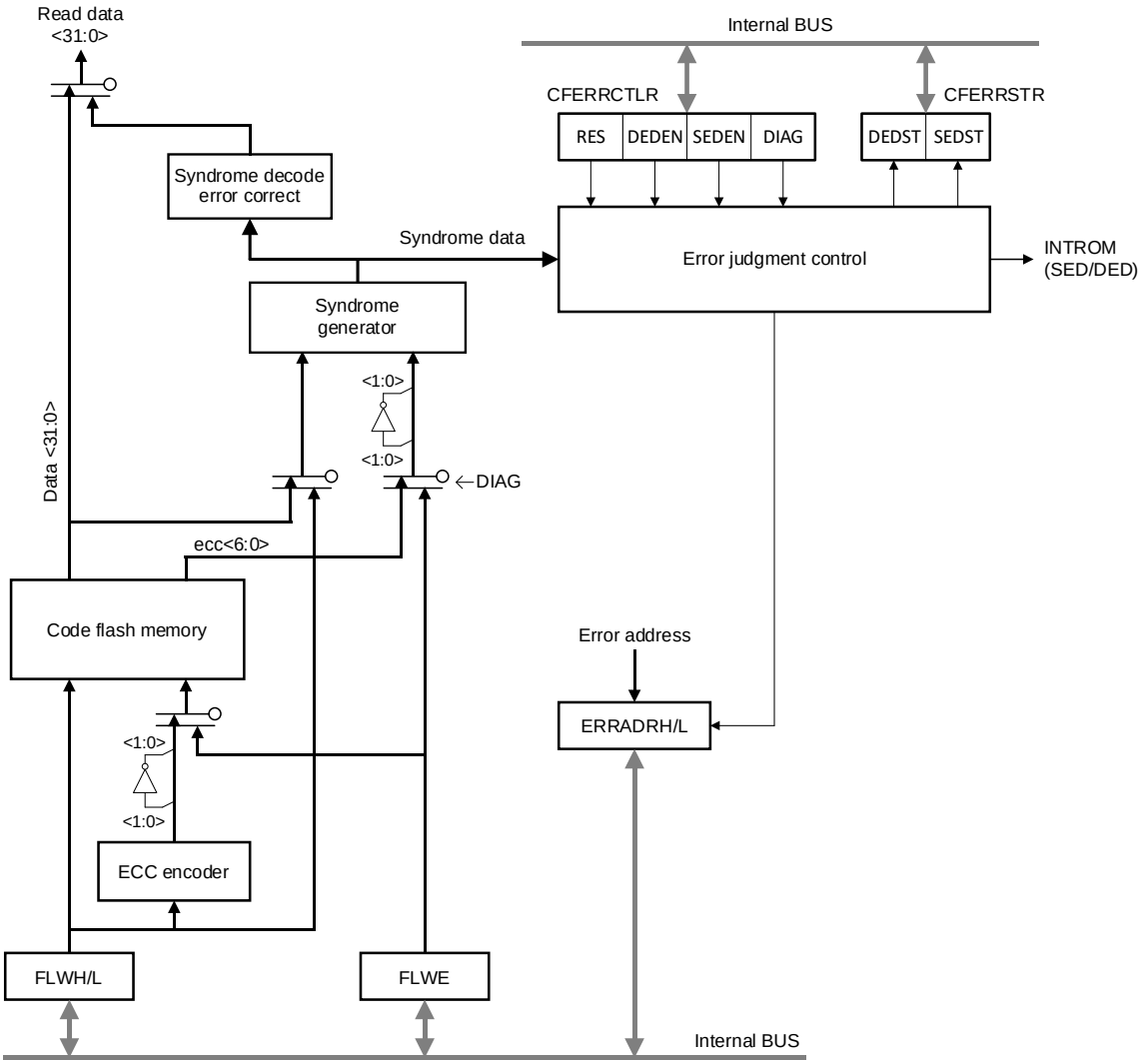
The RL78/F22 and RL78/F25 have the code flash-memory ECC function with 7-bits ECC code for 32 bits data. This function allows 1-bit error detection (SED) and correction (SEC), and 2-bit error detection (DED). Functions of SED, SEC, and DED, and interrupts of SED and DED can be selected to be disabled or enabled. These functions can be tested by using ECC self-diagnosis setting (the DIAG bit in the CFERRCTLR register is set to 1).

- Single-bit error correction (SEC)
- Single-bit error detection (SED)
- Double-bit error detection (DED)
- Error address capturing
- Maskable interrupt (INTROM)

<Configuration>

Figure 29-23 shows a block diagram of the code flash memory ECC function.

Figure 29-23. Block Diagram of Code Flash Memory ECC Function



Lower 2 bit of ECC codes are inverted at after ECC encoder and before syndrome generator to detect stack all_0 faults as 2-bit error. Therefore if 32-bit data of code flash memory is expected as FFFF FFFFH, user shall write FFFF FFFFH to the blank cells after erase. In this case actual ECC code written in code flash memory is 7CH instead of 7FH.

<Control Register>

Register Name	Description	Access Size
CFERRCTLR	Code flash bit error detection function control register	8 bits
CFERRSTR	Code flash bit error detection function status register	8 bits
ERRADRH	Code flash bit error detection address register H	8 bits
ERRADRL	Code flash bit error detection address register L	16 bits
FLWH/L	Flash write buffer register	16 bits
FLWE	Flash ECC write buffer register	8 bits

(1) Code Flash Bit Error Detection Function Control Register (CFERRCTLR)

This register is used to control code flash memory ECC function. The CFERRCTLR register can be read and written by using an 8-bit manipulation instruction.

Figure 29-24. Format of Code Flash Bit Error Detection Function Control Register (CFERRCTLR)

Address: F0098H After reset: 40H R/W

Symbol	7	6	5	4 Note 1	3	2 Note 2	1 Note 1	0 Note 1
CFERRCTLR	RES	DEDEN	SEDEN	0	DIAG	0	0	0

RES	Code flash memory error detection function reset bit
0	No reset request
1	Code flash memory error detection function registers are reset
Reset bit is available for CFERRCTLR, CFERRSTR, ERRADRL/H registers. This bit is cleared to 0 on the next clock after writing 1.	

DEDEN	DED enable bit
0	DED disabled
1	DED enabled

SEDEN	SED enable bit
0	SED disabled
1	SED enabled

DIAG	Code flash memory error detection function self-diagnosis enable bit
0	Self-diagnosis disabled
1	Self-diagnosis enabled

Notes 1. Bits 0, 1, and 4 are read only. The read value is fixed to 0. Writing to these bits are ignored.

2. When writing to bit 2, write 0. When read, the value read depends on the operation mode.

In normal operation mode: 0 is read.

In on-chip debug mode: An undefined value is read.

Cautions 1. Setting DIAG = 1 during flash memory programing is prohibited.
When DEDEN and SEDEN bits of the CFERRCTLR register are set disable, each status flag and interrupt are not triggered. But 1-bit error correction is executed when 1-bit error is detected.

2. When reset is released, the initial values of the DEDST bit in the CFERRSTR register, ERRADRL, and ERRADRH registers are undefined. Before enabling the INTROM (code flash ECC error detection) interrupt, write 1 to the RES bit of the CFERRCTLR register to initialize the CFERRCTLR, CFERRSTR, and ERRADRL/H registers. Also, clear the ROMIF flag in the IF2H register to 0 by software.

(2) Code Flash Bit Error Detection Function Status Register (CFERRSTR)

This register indicates code flash memory ECC error status. The CFERRSTR register can be read and written to using an 8-bit manipulation instruction.

Figure 29-25. Format of Code Flash Bit Error Detection Function Status Register (CFERRSTR)

Address: F0099H After reset: 0x000000B R/W

Symbol	7 Note 1	6	5	4 Note 1	3 Note 1	2 Note 1	1 Note 1	0 Note 1
CFERRSTR	0	DEDST	SEDST	0	0	0	0	0

DEDST Note 2	2-bit error detection status flag
0	No 2-bit error detection.
1	2-bit error is detected. When writing 1, DEDST flag is cleared to 0. This bit is cleared to 0 on the next clock after writing 1.

SEDST	1-bit error correction status flag
0	No 1-bit error correction.
1	1-bit error correction is detected. When writing 1, SEDST flag is cleared to 0. This bit is cleared to 0 on the next clock after writing 1.

Notes 1. Bits 7, 4 to 0 are read only. The read value is fixed to 0. Writing to these bits are ignored.

2. When reset is released, the DEDST bit is undefined. Before enabling the INTROM (code flash ECC error detection) interrupt, write 1 to the RES bit in the CFERRCTLR register to initialize the CFERRCTLR, CFERRSTR, and ERRADRL/H registers (the DEDST bit is cleared to 0).

(3) Code Flash Bit Error Detection Address Register L (ERRADRL)

This register is used to hold error address (bits 15 to 2) temporarily when ECC syndrome error is detected. Once error address is captured to this register and later next syndrome error is detected, the new error address is overwritten to this register.

When the DEDST bit in the CFERRSTR register is 1, the error address due to 1-bit error correction is not overwritten in this register. However, in the same case, the error address by 2-bit error detection is overwritten in this register.

In normal mode (the DIAG bit in the CFERRCTLR register is set to 0), this register is read only, but in diagnostic self-test mode (the DIAG bit is set to 1), this register can be read and written by a 16-bit manipulation instruction.

Figure 29-26. Code Flash Bit Error Detection Address Register L (ERRADRL)

Address: F009CH After reset: xxxxH Note 1 R (DIAG = 0), R/W (DIAG = 1)

Symbol	15	14	13	12	11	10	9	8
ERRADRL	ERRADR[15:8]							

7	6	5	4	3	2	1 Note	0 Note
ERRADR[7:2]						0	0

ERRADR[15:2] Note 2	Latest error detection address
0000H to FFFCH	The value of latest error detection address (bits 15 to 2) is stored.

Notes 1. Bits 1, 0 are read only. The read value is fixed to 0. Writing to these bits are ignored.

2. When reset is released, the ERRADR[15:2] bits are undefined. The ERRADRL register can be initialized to FFFCH by writing 1 to the RES bit in the CFERRCTLR register.

Cautions 1. When DEDEN and SEDEN bits of the CFERRCTLR register are set disable, each status flag and interrupt are not triggered. However, if a bit error is detected, the error address is stored in the ERRADRL/H register. But error address is captured in the ERRADRL register when bit error is detected.

2. When in diagnostic self-test mode (DIAG bit is 1), writing to the ERRADRL and ERRADRH registers may cause the DEDST and SEDST bits in the CFERRSTR register and the ROMIF flag in the IF2H register to become 1. When the diagnosis is completed, clear the DEDST and SEDST bits and the ROMIF flag to 0 by software.

(4) Code Flash Bit Error Detection Address Register H (ERRADRH)

This register is used to hold error address (bits 19 to 16) temporarily when ECC syndrome error is detected. Once error address is captured to this register and later next syndrome error is detected, the new error address is overwritten to this register.

When the DEDST bit in the CFERRSTR register is 1, the error address due to 1-bit error correction is not overwritten in this register. However, in the same case, the error address by 2-bit error detection is overwritten in this register.

In normal mode (the DIAG bit in the CFERRCTLR register is set to 0), this register is read only, but in diagnostic self-test mode (the DIAG bit is set to 1), this register can be read and written by a 8-bit manipulation instruction.

Figure 29-27. Code Flash Bit Error Detection Address Register H (ERRADRH)

Address: F009EH After reset: 0xH R (DIAG = 0), R/W (DIAG = 1)

Symbol	7 Note 1	6 Note 1	5 Note 1	4 Note 1	3	2	1	0
ERRADRH	0	0	0	0	ERRADR[19:16]			

ERRADR[19:16] Note 2	Latest error detection address
0000B to 1111B	The value of latest error detection address (bits 19 to 16) is stored.

Notes 1. Bits 7 to 4 are read only. The read value is fixed to 0. Writing to these bits are ignored.

2. When reset is released, the ERRADR[19:16] bits are undefined. The ERRADRH register can be initialized to 0FH by writing 1 to the RES bit in the CFERRCTLR register.

Cautions 1. When DEDEN and SEDEN bits of the CFERRCTLR register are set disable, each status flag and interrupt are not triggered. But error address is captured in the ERRADRH register when bit error is detected.

2. When in diagnostic self-test mode (DIAG bit is 1), writing to the ERRADRL and ERRADRH registers may cause the DEDST and SEDST bits in the CFERRSTR register and the ROMIF flag in the IF2H register to become 1. When the diagnosis is completed, clear the DEDST and SEDST bits and the ROMIF flag to 0 by software.

(5) Flash Write Buffer Register (FLWH/FLWL)

This register has 2 functions. One is Flash memory write data register for flash memory programing, and another is flash memory data register for code flash memory ECC diagnostic self-test to inject intentional ECC error. When the DIAG bit in the CFERRCTLR register is set to 1, this register can be used as flash memory data register for code flash memory ECC self-test. This register can be read and written by a 16-bit manipulation instruction.

Figure 29-28. Format of Flash Write Buffer Register (FLWH/FLWL)

Address: F00CEH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
FLWH	FLW[31:24]							

7	6	5	4	3	2	1	0
FLW[23:16]							

Address: F00CCH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
FLWL	FLW[15:8]							

7	6	5	4	3	2	1	0
FLW[7:0]							

FLW[31:16]	Code flash memory write data
0000H to FFFFH	If the DIAG bit in the CFERRCTLR register is set to 1, ECC self-test data (bits 31 to 16) to inject intentional ECC error.

FLW[15:0]	Code flash memory write data
0000H to FFFFH	If the DIAG bit in the CFERRCTLR register is set to 1, ECC self-test data (bits 15 to 0) to inject intentional ECC error.

Cautions 1. This register is cleared when a reset signal is generated or the FLRST bit of Flash Initialize Register (FLRST) is set to 1.

2. Set the write data for data flash memory to the lower 8 bits of the FLWL register.

3. See CHAPTER 33 FLASH MEMORY for details on FLWH and FLWL registers.

(6) Flash ECC Write Buffer Register (FLWE)

As same as FLWH/L registers, this register has 2 functions. One is Flash memory write ECC code register for Flash memory programing, and another is flash memory ECC code register for code flash memory ECC diagnostic self-test to inject intentional ECC error. When the DIAG bit in the CFERRCTLR register is set to 1, this register can be used as flash memory data register for code flash memory ECC self-test. This register can be read and written by an 8-bit manipulation instruction.

Figure 29-29. Format of Flash ECC Write Buffer Register (FLWE)

Address: FFFC6H After reset: 00H R/W

Symbol	7 Note	6	5	4	3	2	1	0
FLWE	0	FLWE[6:0]						

FLWE[6:0]	Flash memory ECC write data
0000000B to 1111111B	If the DIAG bit in the CFERRCTLR register is set to 1, ECC self-test data to inject intentional ECC error.

Note Bits 7 is read only. The read value is fixed to 0. Writing to this bit is ignored.

Cautions 1. This register is cleared when a reset signal is generated.

2. When the DCLR bit (in the FSSQ register) is 0, the FLWE register stores ECC data from the ECC area sequencer in the flash memory. When the DCLR bit is 1 (ECC area sequencer stopped), the FLWE register can be used as a diagnostic self-test.

3. See CHAPTER 33 for details on the FLWE register.

<Diagnostic Self-test>

By setting the DIAG bit in the CFERRCTLR register is set to 1, code flash memory ECC function diagnostic self-test can be performed and following functions will be available.

- Syndrome generator input is changed from flash memory output (7-bit ECC and 32-bit data) to FLWE, FLWH/L registers output. By doing this, ECC error injection will be available. After setting the FLWE and FLWH/L, ECC error detection can be done as usual by doing code flash memory read operation of RAM fetch. As a result, error correction, error detections, interrupt, flags can be tested.
- Write operation of code flash bit error detection address register H/L (ERRADRH/L) will be enabled. Due to this, the code flash bit error detection address register H/L can be tested easily.

When the ECC diagnostic function is in self-test mode (DIAG bit is 1), writing to the ERRADRL and ERRADRH registers may cause the DEDST and SEDST bits in the CFERRSTR register and the ROMIF flag in the IF2H register to become 1. When the diagnosis is completed, clear the DEDST and SEDST bits and the ROMIF flag to 0 by software.

<Sector CRC Test>

The sector CRC test is used to determine whether an error detected by the SED is a 1-bit error or a multi-bit error. A CRC value is calculated using a generic CRC for the sector (specified by the user) that contains the address where the bit error was detected. By comparing the calculated CRC value with the expected CRC value stored in another sector, it is possible to detect miscorrection by the ECC function when a multi-bit error occurs.

When using a high-speed CRC to make a judgment, if the target data length is large (e.g., 256 KB of target memory), the 16-bit CRC polynomial may not be able to completely detect bit errors. If the defect rate is not acceptable, use the sector CRC test using a generic CRC.

29.3.6 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.

Caution Do not evaluate this feature in on-chip debug mode. Because the monitor program uses RAM.

<Configuration>

This function has the following functions.

- SP overflow/underflow detection function
- SP overflow/underflow interrupt output function

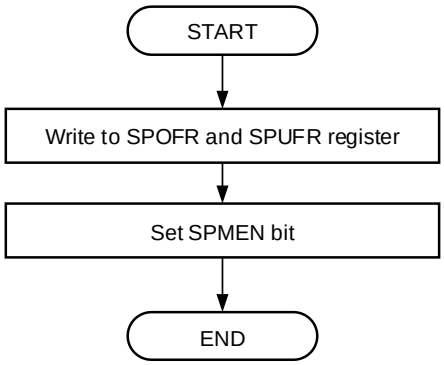
When the SPM enable bit (SPMEN) is 1, an interrupt signal (INTSPM) is generated if the monitored stack pointer value is greater than the specified SFR value (SPOFR) or smaller than the specified SFR value (SPUFR).

When the SPM enable bit (SPMEN) is 1, writing to the SPOFR and SPUFR registers is invalid.

<Register Setting Method>

Figure 29-30 shows the register setting method of this function.

Figure 29-30. 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 29-31. Format of SPM Control Register (SPMCTRL)

Address: F00D8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SPMCTRL	SPMEN	0	0	0	0	0	0	0

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 29-32. 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	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0

Bits 15 to 1	Stack pointer overflow address setting
0000H to FFFEH	Stack pointer overflow address

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 29-33. 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	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0

Bits 15 to 1	Stack pointer underflow address setting
0000H to FFFEH	Stack pointer underflow address

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.

29.3.7 Clock Monitor Function

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.

The Clock monitor module has a Clock monitor self-test mode. The clock monitor self-test function is used to check whether the Clock monitor is operating normally.

When GCSC bit of IAWCTL register is 1, writing to CLMTES register is invalid.

<Control Register>

Register Name	Description	Access Size
CLMTES	Clock monitor test register	1 bit/8 bits

(1) Configuration

Figure 29-34 shows a block diagram of the clock monitor.

Figure 29-34. Block Diagram of Clock Monitor

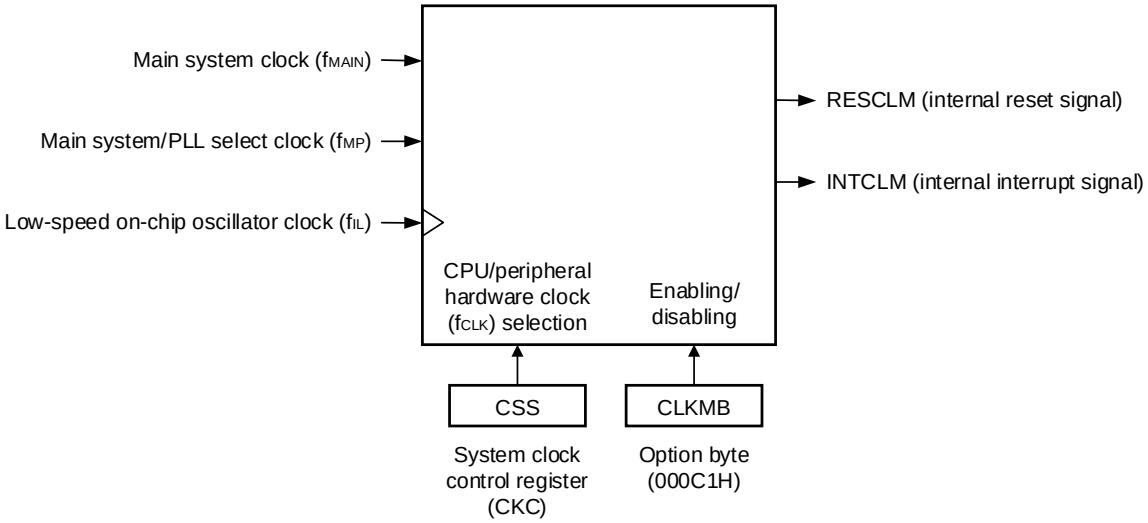


Table 29-2. Operating State of Clock Monitor

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 user option byte (000C1H/040C1H) 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 user option byte (000C1H/040C1H) 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. Do not evaluate the clock monitor function during on-chip debugging.

(4) Clock Monitor Test Register (CLMTES)

Set the CLMTES register by a 1-bit or 8-bit memory manipulation instruction.
When GCSC bit in the IAWCTL register is 1, writing to CLMTES register is invalid.

Figure 29-35. Format of Clock Monitor Test Register (CLMTES)

Address: F02CEH After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	2	<1>	<0>
CLMTES	TESEN	0	0	0	CLMTEN	0	CK2MSK	CK1MSK

TESEN	Clock monitor test function enable bit
0	Test function setting disabled
1	Test function setting enabled

CLMTEN	Clock monitor test mode enable bit
0	Clock monitor test disabled
1	Clock monitor test enabled

CK2MSK	Monitoring clock (f_{MP}) test bit
0	Stop the monitoring clock at low level
1	Does not stop the monitoring clock

CK1MSK	Monitoring clock (f_{MAIN} or f_{MP}) test bit
0	Stop the monitoring clock at low level
1	Does not stop the monitoring clock

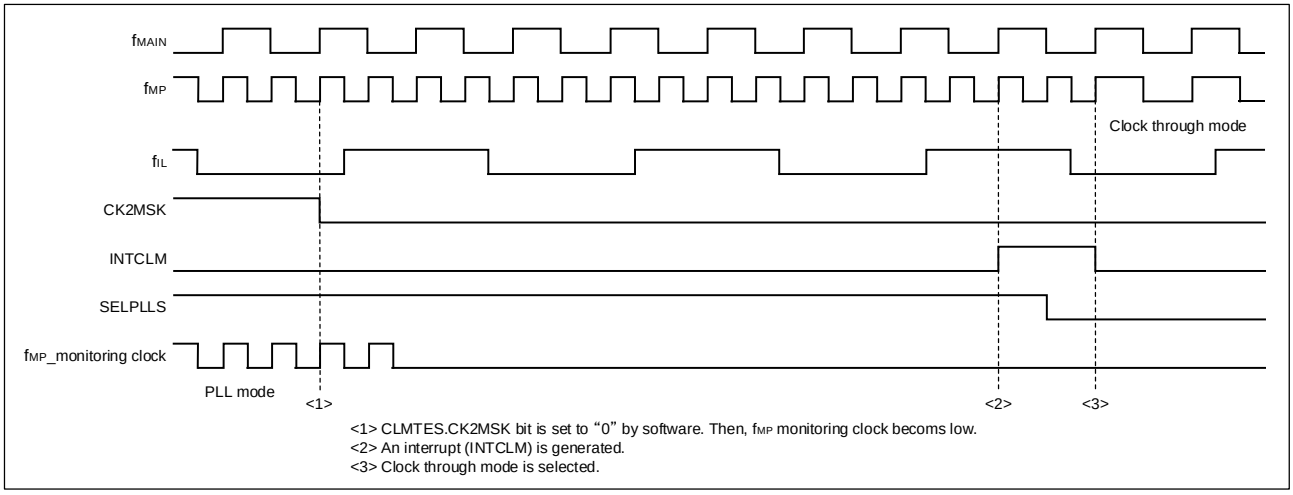
Cautions
1. **Be sure to set bits 2, 4 to 6 of the CLMTES register to 0**
2. **When TESEN bit is 0, writing to CLMTES register is invalid.**
3. **When writing "1" to the TESEN bit, write with all bits 0, 1, and 3 set to "1".**
4. **Once the device enters clock through mode with the CK2MSK bit set, reset is the only way to release clock through mode.**

Remark The CLMTES register is write protected by GCSC bit in the IAWCTL register. This register can be written only when GCSC bit is 0.

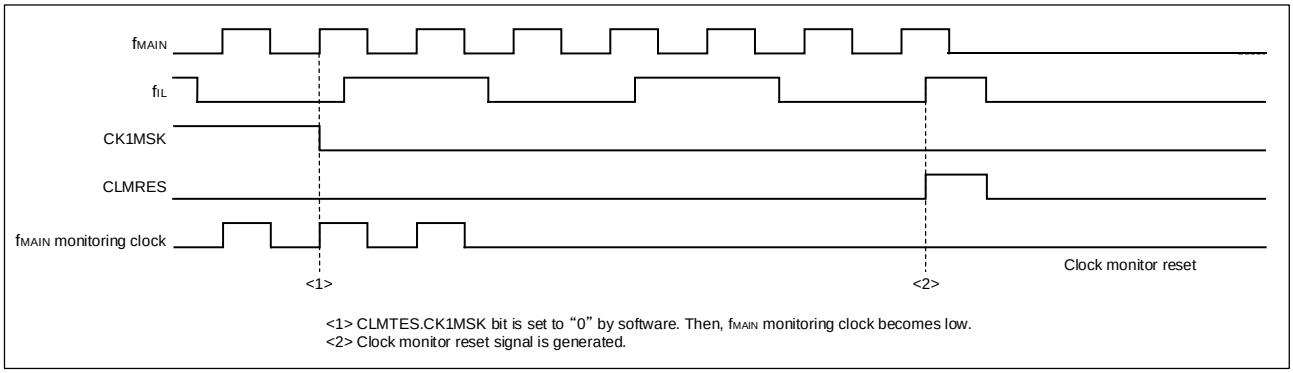
Figure 29-36 shows clock monitor self-test operation.

Figure 29-36. Operation in Clock Monitor Self-test

(a) Clock Monitor Self-Test Function (CK2MSK=0)



(b) Clock Monitor Self-Test Function (CK1MSK=0)



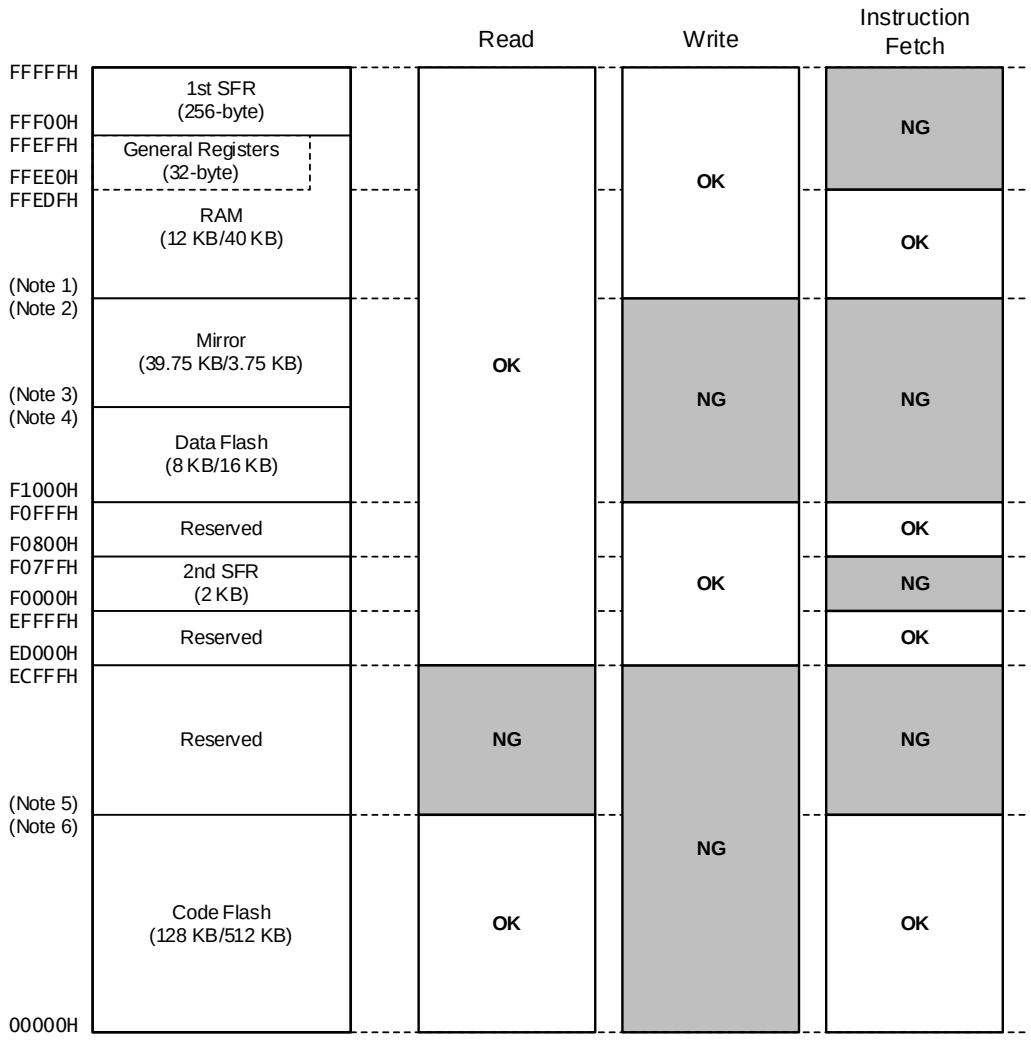
29.3.8 Invalid Memory Access Detection Function

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 29-37**.

Figure 29-37. Invalid Access Detection Area



The notes are shown in the table below.

Note	Corresponding Addresses	RL78/F25	RL78/F22
(Note 1)	RAM area start address	F5F00H	FCF00H
(Note 2)	Mirror area bottom address	F5EFFH	FCEFFH
(Note 3)	Mirror area start address	F5000H	F3000H
(Note 4)	Data flash memory area bottom address	F4FFFH	F2FFFH
(Note 5)	Code flash memory area bottom address + 1	80000H	20000H
(Note 6)	Code flash memory area bottom address	7FFFFH	1FFFFH

Remark Internal RAM size is depending on the RAMSAR register settings. For details, refer to **CHAPTER 3 CPU ARCHITECTURE**.

<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.
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 29-38. 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.

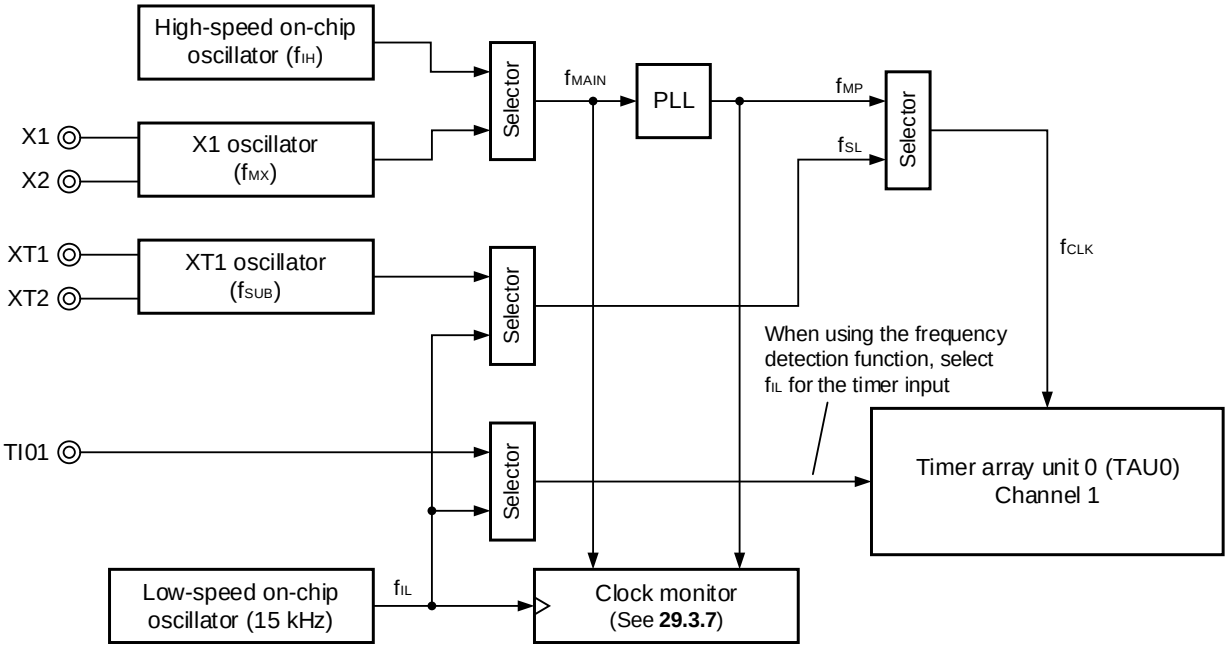
Note Only writing 1 to the IAWEN bit is valid, and writing 0 after setting the IAWEN bit to 1 is invalid.

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 32 OPTION BYTE.**)

29.3.9 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 29-39. Configuration of Frequency Detection Function



<Operational Overview>

Whether the clock frequency is correct or not correct 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>

(1) 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 29-40. 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 1	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 2

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 2

TIS04 Note 1	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 2

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 2
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 RL78/F25 products. Write "0" when writing to the timer input select register 0 (TIS0) of RL78/F22 products.
2. Provided only in RL78/F25 products. Do not set in the RL78/F22 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).**

29.3.10 A/D Test Function

12-bit A/D converter supports the following two diagnostic functions.

- Self-diagnosis of 12-bit A/D converter

  Self-diagnosis of 12-bit A/D converter is used to check whether the A/D converter is operating normally by executing A/D conversions of an internal voltage of the low-potential reference voltage, the high-potential reference voltage, and the high-potential reference voltage/2.

- Disconnection detection assist function

  This A/D converter incorporates the function to pre-charge or dis-charge sampling capacitance before start A/D conversion. This function enables disconnection detection assist for the wire connected to analog input.

<Related Register>

The registers used for A/D Test Function are shown below.

- A/D control expansion register (ADCER)
- A/D self-diagnosis data register (ADRD)
- A/D disconnection detection control register (ADDISCR)

For details of A/D Test Function, see “**12.3.8 Disconnection Detection Assist Function**” and “**12.3.9 Self-diagnosis Function**”.

29.3.11 Watchdog Timer Function

For details of watchdog timer function, see **CHAPTER 11 WATCHDOG TIMER**.

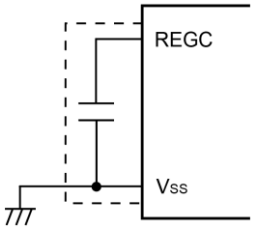
CHAPTER 30 SECURITY FUNCTIONS

Please refer to the separate volume for security functions.

CHAPTER 31 REGULATOR

31.1 Regulator Overview

The RL78/F22 and RL78/F25 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 Vss via a capacitor (0.47 to 1 μF). 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.

The regulator output voltage is 1.5 V.

CHAPTER 32 OPTION BYTE

32.1 Functions of Option Bytes

Addresses 000C0H to 000C4H of the flash memory of the RL78/F22 and RL78/F25 form an option byte area.

Option bytes consist of user option byte (000C0H to 000C2H), on-chip debug option byte (000C3H), and security option byte (000C4H).

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 000C4H are replaced by 040C0H to 040C4H. Therefore, set the same values for 040C0H to 040C4H as for 000C0H to 000C4H.

32.1.1 User Option Byte (000C0H/040C0H to 000C2H/040C2H)

(1) User Option Byte (000C0H/040C0H)

- ○ Operation of watchdog timer
  - Operation is stopped or enabled in the HALT, STOP, or SNOOZE mode.
- ○ Setting of overflow 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 040C0H when the boot swap operation is used because 000C0H is replaced by 040C0H.

(2) User Option Byte (000C1H/040C1H)

- ○ Setting of LVD0 detection level (V_{LVD0})
- ○ Operation of clock monitor
  - Operation is stopped or enabled.

Caution Set the same value as 000C1H to 040C1H when the boot swap operation is used because 000C1H is replaced by 040C1H.

(3) User Option Byte (000C2H/040C2H)

- Setting of RESOUTB output function
- Setting of the frequency of the high-speed on-chip oscillator
  - Select from 2 MHz, 4 MHz, 8 MHz, 16 MHz, 20 MHz, 32 MHz, 40 MHz, 64 MHz, and 80 MHz.

Caution Set the same value as 000C2H to 040C2H when the boot swap operation is used because 000C2H is replaced by 040C2H.

32.1.2 On-chip Debug Option Byte (000C3H/040C3H)

- Control of on-chip debug operation
  - On-chip debug operation is disabled or enabled.
- Control of flash serial programming operation
  - Flash serial programming 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 040C3H when the boot swap operation is used because 000C3H is replaced by 040C3H.

32.1.3 Security Option Byte (000C4H/040C4H)

- Control of on-chip debug and flash serial programming security ID read
  - On-chip debug and flash serial programming security ID read are disabled or enabled.

Cautions 1. Set the same value as 000C4H to 040C4H when the boot swap operation is used because 000C4H is replaced by 040C4H.

2. In case of setting IDRDEN to 0 on the security option byte (000C4H/040C4H), be sure to enable the prohibition of rewriting boot cluster 0. For details about disabling rewriting boot cluster 0 setting, see 33.8 Security Settings.

32.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 32-1. Format of User Option Byte (000C0H/040C0H)

Address: 000C0H/040C0H 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% of the overflow time + 1/2 f_{WDT} is reached.

WINDOW1	WINDOW0	Watchdog timer window open period Note 4
0	0	Setting prohibited
0	1	50%
1	0	75% Note 3
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} (474.89 ms)
1	1	0	2^{14} / f_{WDT} (949.79 ms)
1	1	1	2^{16} / f_{WDT} (3799.18 ms)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP/SNOOZE mode)
0	Counter operation stopped in HALT/STOP/SNOOZE mode Note 4
1	Counter operation enabled in HALT/STOP/SNOOZE mode

Notes 1. Set the same value as 000C0H to 040C0H when the boot swap operation is used because 000C0H is replaced by 040C0H .

2. The setting at shipment of the user option byte is FFH.

3. When using the window open period 75% setting, there is a watchdog timer counter clear prohibition period. For details, see **11.4.3 Setting window open period of watchdog timer**.

4. The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

Caution The watchdog timer continues to operate even when the flash memory is being rewritten. Set the overflow time and window open period taking into consideration the delay in clearing the watchdog timer counter.

Remarks 1. f_{WDT}: Low-speed on-chip oscillator clock frequency for WDT

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 **29.3.8 Invalid memory access detection function**.)

Figure 32-2. Format of User Option Byte (000C1H/040C1H)

Address: 000C1H/040C1H Note 1 After reset: – (user setting value Note 2)

7	6	5	4	3	2	1	0
LVD0ENB	1	1	CLKMB	1	LVD0V2	LVD0V1	LVD0V0

- LVD0 setting (reset mode)

Detection voltage		Option byte Setting Value				
V_{LVD0}		LVD0ENB	CLKMB	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge					
1.90 V Note 4	1.86 V Note 4	0	× Note 3	1	1	0
2.97 V	2.91 V		× Note 3	0	1	1
3.96 V	3.88 V		× Note 3	0	1	0
Other than above		Setting prohibited				

- LVD0 setting (LVD off)

Detection voltage		Option byte Setting Value				
V_{LVD0}		LVD0ENB	CLKMB	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge					
–	–	1	× Note 3	0	1	0
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 040C1H when the boot swap operation is used because 000C1H is replaced by 040C1H.

2. The setting at shipment of the user option byte is FFH.

3. Write the setting value of the clock monitor bit (CLKMB).

4. Can be set only for RL78/F22 products; do not set for RL78/F25 products.

Remark For details of the LVD, see **27.1 Functions of Voltage Detector**.

Figure 32-3. Format of User Option Byte (000C2H/040C2H)

Address: 000C2H/040C2H [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 • The low level is output during a reset. • The high level is automatically output upon release from the reset state. • The output latch value has no effect on the output.
1	Selects P130 as a general port pin (output only) • The low level is output during a reset. • 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	80 MHz
1	0	0	0	0	64 MHz
0	1	0	0	0	40 MHz
0	0	0	0	0	32 MHz
0	1	0	0	1	20 MHz
0	0	0	0	1	16 MHz
0	0	0	1	0	8 MHz
0	0	0	1	1	4 MHz
0	0	1	0	0	2 MHz
Other than above					Setting prohibited

Notes 1. Set the same value as 000C2H to 040C2H when the boot swap operation is used because 000C2H is replaced by 040C2H.

2. The setting at shipment of the user option byte is FFH.

32.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 32-4. Format of On-chip Debug Option Byte (000C3H/040C3H)

Address: 000C3H/040C3H [Note 1] After reset: – (user setting value [Note 3])

7	6	5	4	3	2	1	0
OCDENSET	0	FLPEN	0	0	1	HPIEN [Note 2]	OCDERSD

FLPEN	OCDENSET	HPIEN [Note 2]	OCDERSD	Control of on-chip debug operation
0	X	X	X	Disables flash serial programming and on-chip debugging operation.
1	0	0	0	• Disables on-chip debug operation.
1	1	0	0	• Enables on-chip debugging. • Disables hot plug-in operation. • Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	0	1	• Enables on-chip debugging. • Disables hot plug-in operation. • Does not erase data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	1	1	• Enables on-chip debugging. • Enables hot plug-in operation. • Does not erase data of flash memory in case of failures in authenticating on-chip debug security ID.
1	Other than the above			Setting prohibited

FLPEN	Control of flash serial programming and on-chip debugging operation
0	Disables flash serial programming and on-chip debugging operation.
1	Enables flash serial programming operation. [Note 4]

Notes 1. Set the same value as 000C3H to 040C3H when the boot swap operation is used because 000C3H is replaced by 040C3H.

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.

3. The setting at shipment of the on-chip debug option byte is FFH.

4. Set the enable/disable setting of on-chip debugging function together with the OCDENSET, HPIEN, and OCDERSD bits.

Caution Bits 7, 5, 1, and 0 (OCDENSET, FLPEN, HPIEN, and OCDERSD) can only be specified a value. Be sure to set 0 to bit 6 and 001B to bits 4 to 2.

Remark The value on bits 3 to 1 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.

32.4 Format of Security Option Byte

The format of security option byte is shown below.

Figure 32-5. Format of Security Option Byte (000C4H/040C4H)

Address: 000C4H/040C4H [Note 1] After reset: – (user setting value [Note 2])

7	6	5	4	3	2	1	0
1	1	1	1	1	IDRDEN	1	0

IDRDEN	Control of on-chip debug and flash serial programming security ID read
0	Disables read of on-chip debug and flash serial programming security ID. When areas on-chip debug security ID (000C6H to 000D5H) and flash serial programming security ID (000D6H to 000E5H) are read, all bits in this area are read as "0".
1	Enables read of on-chip debug and flash serial programming security ID.

Notes 1. Set the same value as 000C4H to 040C4H when the boot swap operation is used because 000C4H is replaced by 040C4H.

2. The setting at shipment of the security option byte is FFH.

Cautions 1. Bit 2 (IDRDEN) can only be specified a value. Be sure to set 11111B to bit 7 to bit 3 and set 10B to bit 1 to bit 0.

2. IDRDEN is not applicable to on-chip debug security ID of 040C6H-040D5H and flash serial programming security ID of 040D6H-040E5H.

3. When setting IDRDEN to 0, be sure to enable rewrite prohibition for boot cluster 0. For details on the settings for prohibiting rewriting of boot cluster 0, refer to 33.8 Security Settings.

Remark IDRDEN is set to 1 while using the on-chip debugging function.

32.5 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.

```
        .CSEG   OPT_BYTE
        .DB     0x36    ; 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     0x6B    ; Select rising edge 2.97 V, falling edge 2.91 V for VLVD0
                        ; Select reset mode as the LVD0 operation mode
                        ; Operation of clock monitor
        .DB     0xE4    ; Setting of the RESOUTB output function
                        ; Select 2 MHz as the frequency of the high-speed on-chip oscillator clock
        .DB     0xA5    ; Enables on-chip debug and flash serial programming operation,
                        ; disables hot plug-in operation,
                        ; does not erase flash memory data when security ID authorization fails
        .DB     0xFA    ; Disables read of on-chip debug and flash serial programming security ID.
```

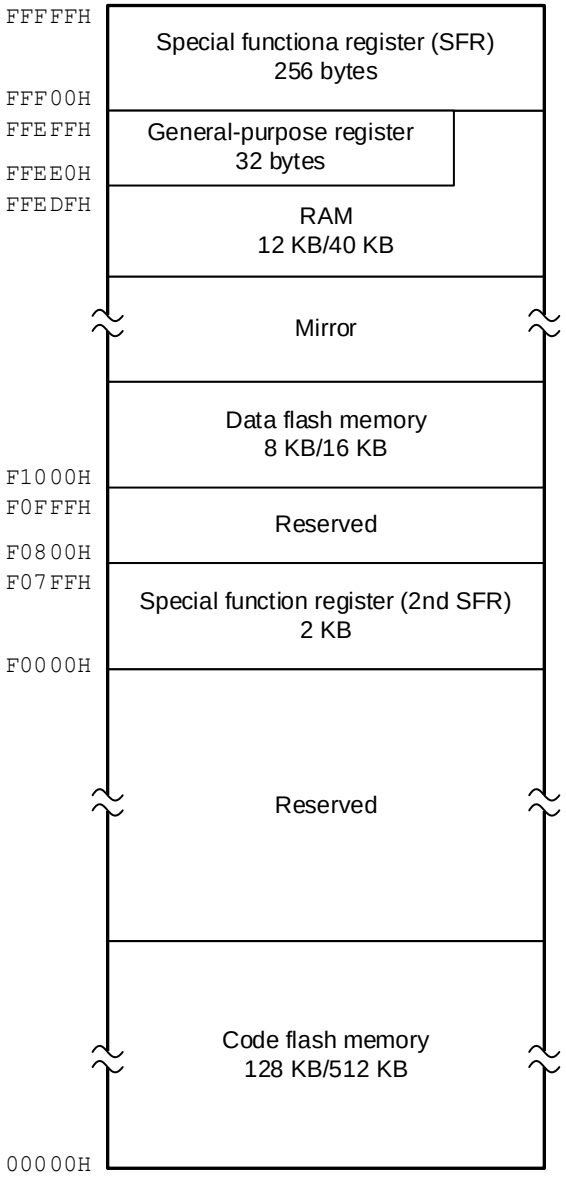
When the boot swap function is used during self programming, 000C0H to 000C4H is switched to 040C0H to 040C4H. Describe to 040C0H to 040C4H, therefore, the same values as 000C0H to 000C4H as follows.

```
OPT2    .CSEG   AT  0x040C0
        .DB     0x36    ; 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     0x6B    ; Select rising edge 2.97 V, falling edge 2.91 V for VLVD0
                        ; Select reset mode as the LVD0 operation mode
                        ; Operation of clock monitor
        .DB     0xE4    ; Setting of the RESOUTB output function
                        ; Select 2 MHz as the frequency of the high-speed on-chip oscillator clock
        .DB     0xA5    ; Enables on-chip debug and flash serial programming operation,
                        ; disables hot plug-in operation,
                        ; does not erase flash memory data when security ID authorization fails
        .DB     0xFA    ; Disables read of on-chip debug and flash serial programming security ID.
```

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 040C0H to 040C4H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.

CHAPTER 33 FLASH MEMORY

The RL78/F22 and RL78/F25 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.



Remark Memory size of code flash memory, data flash memory and RAM is shown follows.
RL78/F22: left side value of “/”
RL78/F25: right side value of “/”

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 **33.5 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 **33.2 Serial Programming Using External Device (That Incorporates UART)**.
- Self-programming
  The user application can execute self-programming of the code flash memory. See **33.7 Self-Programming** for details on code flash memory self-programming.

Caution Write data to all unused areas in the code flash memory. When the CPU reads the erased flash memory area, the code flash ECC error detection interrupt occurs. Also, it is recommended to writing data "FFH" to unused areas. If the CPU fetches the FFH code, an illegal instruction internal reset will occur.
The following shows how to write to unused areas.
- **When creating a write file (HEX, MOT), fill unused areas with data (e.g., FFH).**
- **When writing with a flash programmer, fill unused areas with data (e.g., FFH) and write.**

When using the on-chip debug function, set "Initialize unused space during flash programming" to [Yes] in the debug settings of the integrated development environment such as CS+ or e2studio. For details, refer to the User's Manual of the development environment.

The data flash memory can be written using a dedicated flash memory programmer or using serial programming by an external device. It can also be rewritten during user program execution (background operation). For details about accessing or writing to the data flash memory, see **33.7 Self-Programming** and **33.9 Data Flash Memory**.

33.1 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/F22 and RL78/F25.

- PG-FP6
- E2 or E2 Lite 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/F22 or RL78/F25 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 before the RL78/F22 or RL78/F25 is mounted on the target system.

Table 33-1. Wiring Between the RL78/F22 or RL78/F25 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin Number						
					RL78/F25				RL78/F22		
					100-pin	80-pin	64-pin	48-pin	48-pin	32-pin	24-pin
Signal Name		I/O	Pin Function		QFP	QFP	QFP	QFP	QFP	QFN	QFN
PG-FP6	E2 or E2 Lite on-chip debugging emulator										
–	TOOL0	I/O	Transmit/receive signal	TOOL0/ P40	12	9	5	3	3	3	2
SI/RXD	–	I/O	Transmit/receive signal								
–	$\overline{\text{RESET}}$	Output	Reset signal	RESET	13	10	6	4	4	4	3
/RESET	–	Output									
V_{CC}	V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	22	19	15	12	12	10	9
				AV_{DD}	73	59	47	35	35	24	18
GND		–	Ground	V_{SS}	20	17	13	11	11	9	8
				AV_{SS}	74	60	48	36	36	25	19
				EV_{SS}	21, 43	18	14	–	–	–	–
				REGC Note	19	16	12	10	10	8	7
FLMD1	EMV_{DD}	–	Driving power for TOOL0 pin	V_{DD}	–	–	–	12	12	10	9
				EV_{DD}	23, 53	20	16	–	–	–	–

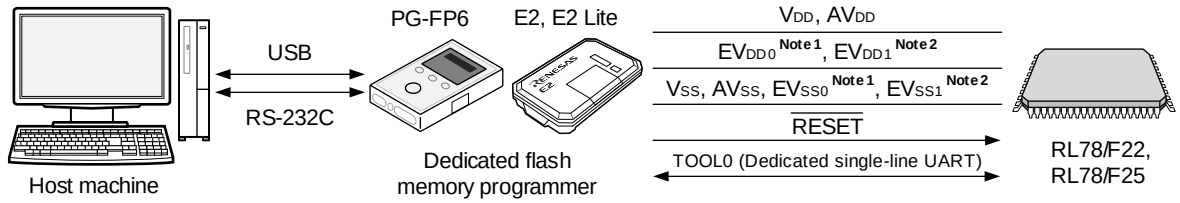
Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

33.1.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78/F22 or RL78/F25 is illustrated below.

Figure 33-1. Environment for Writing Program for 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.

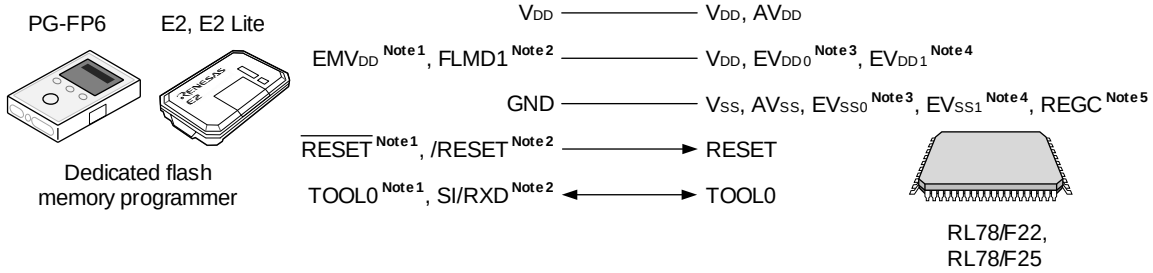
To interface between the dedicated flash memory programmer and the RL78/F22 or RL78/F25, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

33.1.2 Communication Mode

Communication between the dedicated flash memory programmer and the RL78/F22 or RL78/F25 is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78/F22 or RL78/F25.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 33-2. Communication with Dedicated Flash Memory Programmer



Notes 1. When using E2 or E2 Lite on-chip debugging emulator.
2. When using PG-FP6.
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/F22 and RL78/F25. See the manual of PG-FP6 or E2, E2 Lite on-chip debugging emulator for details.

Table 33-2. Pin Connection

Dedicated Flash Memory Programmer				RL78/F22 and RL78/F25
Signal Name		I/O	Pin Function	Pin Name
PG-FP6	E2, E2 Lite on-chip debugging emulator			
V_{CC}	V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, AV_{DD}
GND		–	Ground	V_{SS}, AV_{SS}, EV_{SS0} Note 1, EV_{SS1} Note 2, REGC Note 3
FLMD1	EMV_{DD}	–	Driving power for TOOL0 pin	V_{DD}, EV_{DD0} Note 1, EV_{DD1} Note 2
/RESET	–	Output	Reset signal	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 μF).

Caution The connection destination pins differ depending on the product. For details, see Table 33-1.

33.1.3 Usage Notes

- Be sure to set 0 to bit 0 in the Security Option Byte (000C4H/040C4H), when erasing the data flash memory by dedicated flash memory programmer.
- Do not execute erase command by selected both code flash memory and data flash memory in the block settings of the dedicated flash memory programmer. If erasing the entire code flash memory and data flash memory by the dedicated flash memory programmer, erase the data flash memory with setting 0 to bit 0 in the Security Option Byte (000C4H/040C4H) in advance, and then erase code flash memory.

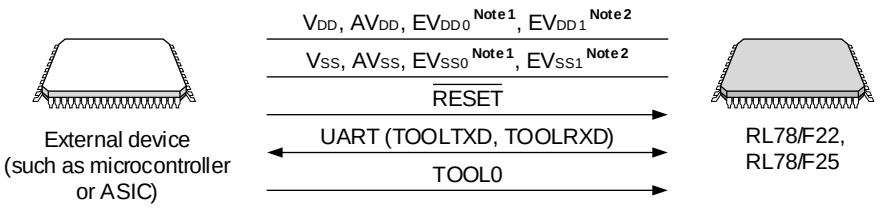
33.2 Serial Programming Using External Device (That Incorporates UART)

On-board data writing to the internal flash memory is possible by using the RL78/F22 or RL78/F25 and an external device (a microcontroller or ASIC) connected to a UART.

33.2.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78/F22 or RL78/F25 is illustrated below.

Figure 33-3. Environment for Writing Program to Flash Memory



Notes 1. 64, 80, 100-pin products only.
2. 100-pin products only.

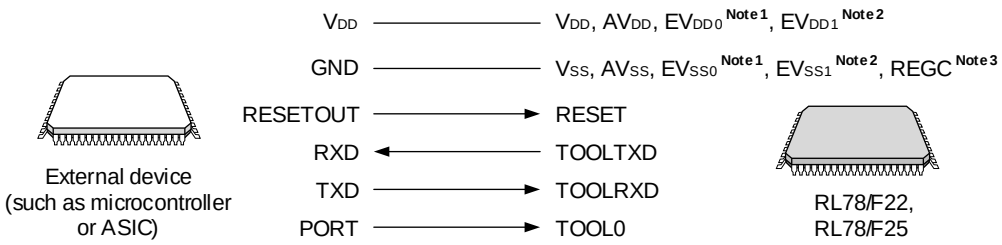
Processing to write data to or delete data from the RL78/F22 or RL78/F25 by using an external device is performed on-board. Off-board writing is not possible.

33.2.2 Communication Mode

Communication between the external device and the RL78/F22 or RL78/F25 is established by serial communication using the TOOLTXD and TOOLRXD pins via the dedicated UART of the RL78/F22 or RL78/F25.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 33-4. 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/F22 and RL78/F25.

Table 33-3. Pin Connection

External Device			RL78/F22 and RL78/F25
Signal Name	I/O	Pin Function	Pin Name
VDD	I/O	VDD voltage generation/power monitoring	VDD, AVDD, EVDD0 Note 1, EVDD1 Note 2
GND	–	Ground	VSS, AVSS, EVSS0 Note 1, EVSS1 Note 2, REGC Note 3
RESETOUT	Output	Reset signal output	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).

33.3 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 **33.5.2 Flash Memory Programming Mode**.

33.3.1 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 t_{HD} 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}: Time to hold the TOOL0 pin at the low level after releasing external reset when entering flash memory programming mode.

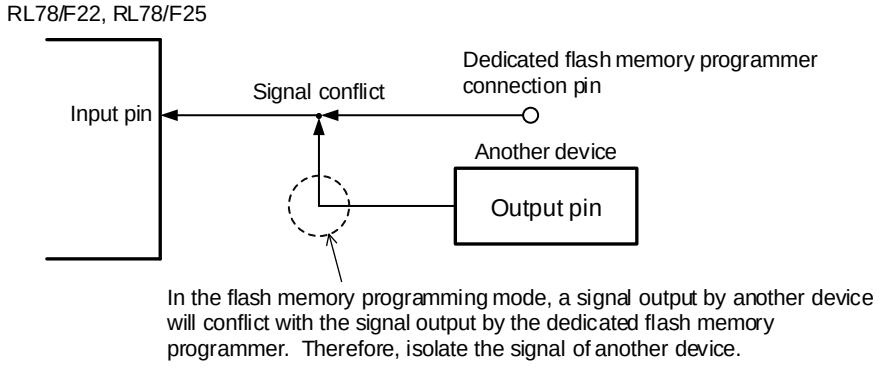
2. The SAU and IICA pins are not used for communication between the RL78/F22 or RL78/F25 and the dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

33.3.2 RESET Pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are connected to the 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 33-5. Signal Conflict (RESET Pin)



33.3.3 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} Note 1, or EV_{DD1} Note 2, 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.

33.3.4 REGC Pin

Connect the REGC pin to GND via a capacitor (0.47 to 1 μF) in the same manner as during normal operation. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

33.3.5 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.

33.3.6 Power Supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} and AV_{DD} pins to V_{DD} of the flash memory programmer, and the V_{SS} and AV_{SS} pins 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}, AV_{DD}, V_{SS}, and AV_{SS} pins to V_{DD} Note and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

Note The name of the signal for connection in the case of the PG-FP6 is V_{CC}.

33.4 Flash Serial Programming Security ID

The RL78/F22 and RL78/F25 have a flash serial programming operation control bit in the flash memory at 000C3H (see **CHAPTER 32 OPTION BYTE**) and a flash serial programming security ID setting area at 000D6H to 000E5H, to prevent third parties from programming flash memory.

When the boot swap function is used, also set a value that is the same as that of 040C3H and 040D6H to 040E5H in advance, because 000C3H, 000D6H to 000E5H and 040C3H, 040D6H to 040E5H are switched.

Table 33-4. Flash Serial Programming Security ID

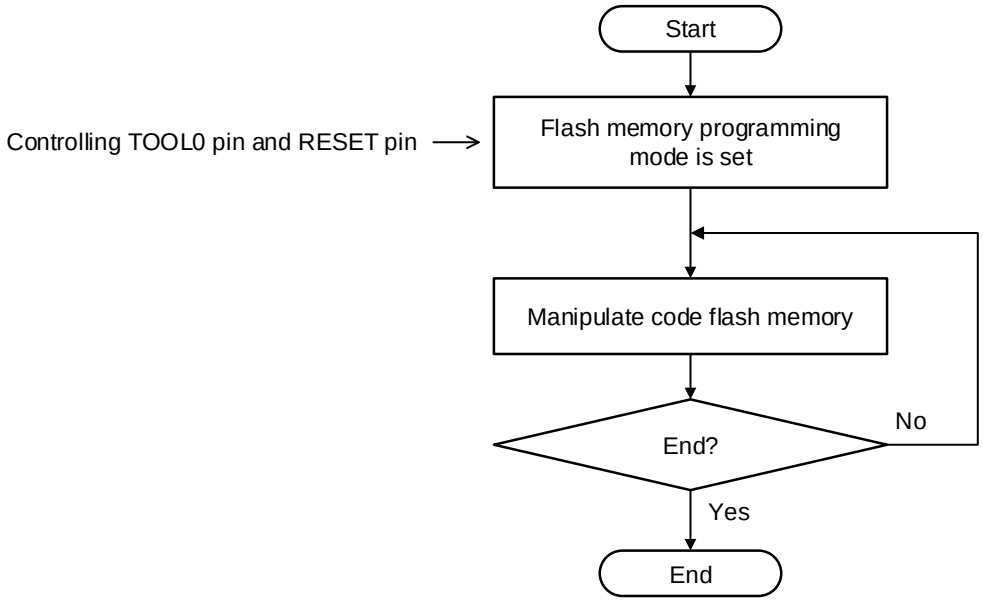
Address	Flash Serial Programming Security ID
000D6H to 000E5H	Any ID code of 16 bytes (except for all “FFH”)
040D6H to 040E5H	

33.5 Serial Programming Method

33.5.1 Serial Programming Procedure

The following figure illustrates the procedure to manipulate the flash memory.

Figure 33-6. Code Flash Memory Manipulation Procedure



33.5.2 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/F22 and RL78/F25 to the dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

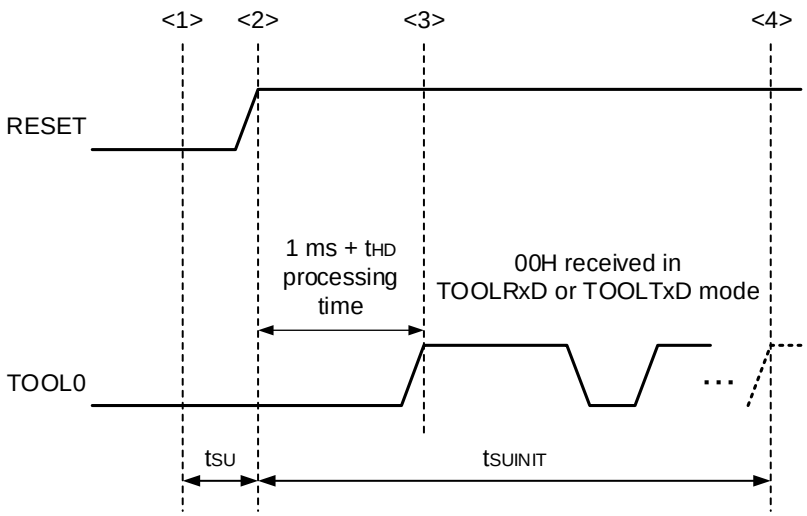
<Serial programming using an external device (UART communication)>

Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 33-5**). Then, perform steps <1> to <4> in **Figure 33-7** to enter the flash memory programming mode.

Table 33-5. 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 33-7. 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.

Remarks 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}: The time from when the TOOL0 pin places low level to when the external reset ends.

t_{HD}: Time for which the TOOL0 pin is held low level after the external and internal reset is released (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 33-6**.

Table 33-6. Voltages at Which Data can be Written, Erased, or Verified

Voltages at which data can be written, erased, or verified	Operating frequency
$2.7\ V \leq V_{DD} \leq 5.5\ V$	2 MHz to 40 MHz
$1.8\ V \leq V_{DD} \leq 2.7\ V$	2 MHz to 16 MHz

Remark For details about communication commands, see **33.5.4 Communication Commands**.

33.5.3 Selecting Communication Mode

Communications modes of the RL78/F22 and RL78/F25 are as follows.

Table 33-7. 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.

33.5.4 Communication Commands

The RL78/F22 and RL78/F25 executes serial programming through the commands listed in **Table 33-8**.

The signals sent from the dedicated flash memory programmer or external device to the RL78/F22 or RL78/F25 are called commands, and programming functions corresponding to the commands are executed.

Table 33-8. 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
	Secure Programming	Writes data to a specified area in the flash memory, and secure data is also registered at the same time.Note
Getting information	Silicon Signature	Gets information from the RL78/F22 or RL78/F25 (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 33-9 lists and describes signature data. **Table 33-10** shows examples of signature data.

Table 33-9. 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 (Sent from lower address. Example. 00000H to 1FFFFH (128 KB) → FFH, FFH, 01H)	3 bytes
Data flash memory area last address	Last address of data flash memory area (Sent from lower address. Example. F1000H to F2FFFH (8 KB) → FFH, 2FH, 0FH)	3 bytes
Firmware version	Version information of firmware for programming (Sent from upper address. Example. From Ver. 1.23 → 01H, 02H, 03H)	3 bytes

Table 33-10. Example of Signature Data

Field name	Description	Number of transmit data	Data (hexadecimal)
Device code	RL78/F22, F25	3 bytes	10 00 0C
Device name	R7F122FG	10 bytes	52 = "R" 37 = "7" 46 = "F" 31 = "1" 32 = "2" 32 = "2" 46 = "F" 47 = "G" 20 = " " 20 = " "
Code flash memory area last address	Code flash memory area 00000H to 1FFFFH (128 KB)	3 bytes	FF FF 01
Data flash memory area last address	Data flash memory area F1000H to F2FFFH (8 KB)	3 bytes	FF 2F 0F
Firmware version	Ver.1.23	3 bytes	01 02 03

33.6 Processing Time for Each Command When PG-FP6 Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP6 is used as a dedicated flash memory programmer.

Table 33-11. Processing Time for Each Command when PG-FP6 is in Use (Reference Value)

PG-FP6 Command	Code Flash	
	128 KB (RL78/F22)	512 KB (RL78/F25)
Erasing	0.7 s	1.8 s
Writing	2.3 s	8.0 s
Verification	2.0 s	6.9 s
Writing after erasing	2.5 s	8.8 s

Remark The command processing times (reference values) shown in the table are typical values under the following conditions.

Port: TOOL0 (single-line UART)
Speed: 1,000,000 bps

33.7 Self-Programming

The RL78/F22 and RL78/F25 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, 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 in the state where the IE flag is cleared (0) by the DI instruction.

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, wait at least 40 μs in software for oscillation accuracy stabilization time before executing flash self-programming.

4. To rewrite the code flash memory or extra area, place the code or values in the RAM. In code flash programming mode, reference to the code flash memory is not possible. Accordingly, in code flash programming mode, copy the user program that is to be executed from the ROM (code flash memory) and its data for reference to the RAM in advance so that the program can be executed and reference to the data in the RAM is possible.

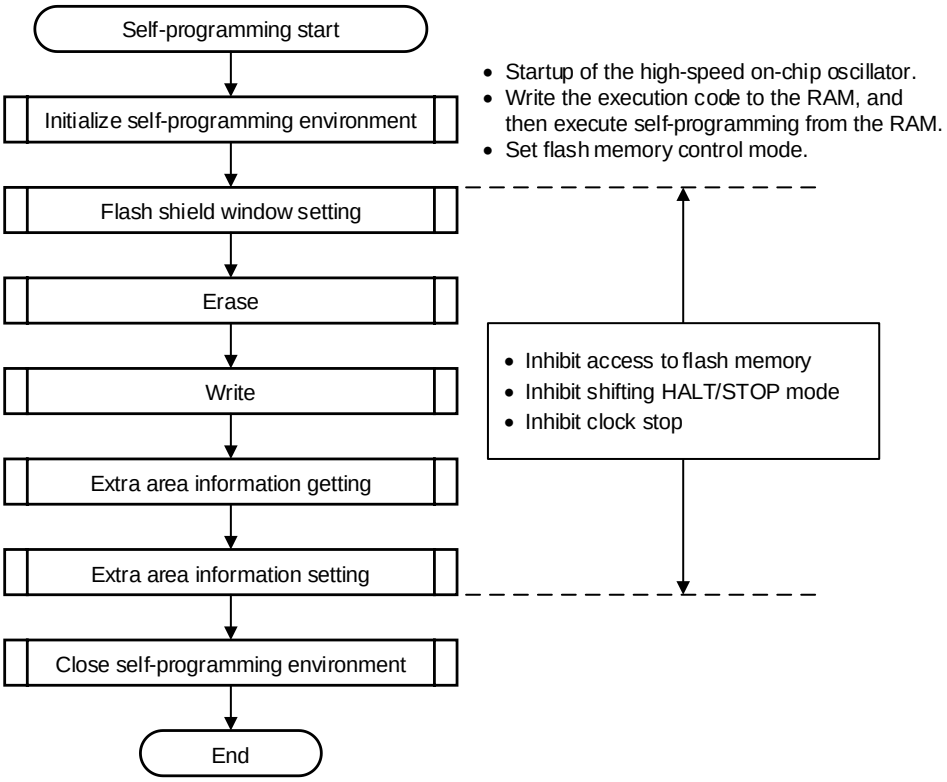
5. Interrupts should be disabled (DI state) while self-programming is in progress. When suspending self-programming and use interrupts, before enabling interrupts (EI state), move to non-programable mode and mask the code flash memory ECC interrupt.
When the CPU reads the erased flash memory area, the code flash ECC error detection interrupt occurs. Write data to all unused areas in the code flash memory. Also, it is recommended to writing data “FFH” to unused areas. If the CPU fetches the FFH code, an illegal instruction internal reset will occur.

6. The code flash memory ECC circuit must be initialized after it has been successfully written to code flash memory. Reset (external or internal reset) to initialize the code flash memory ECC circuit.

33.7.1 Self-Programming Procedure

The following figure shows the flow of rewriting code flash memory using self-programming.

Figure 33-8. Flow of Self-Programming (Rewriting Flash Memory)



33.7.2 Registers to Control the Flash Memory

Table 33-12 lists the flash memory register to control flash memory programming.

Table 33-12. Flash Memory Register Configuration

Address	Register Name	Symbol	After Reset	Access Size
F0090H	Data flash control register	DFLCTL	00H	1, 8
F00C0H	Flash programming mode control register	FLPMC	08H	8
F00C1H	Flash area selection register	FLARS	00H	1, 8
F00C2H	Flash address pointer register L	FLAPL	0000H	16
F00C4H	Flash address pointer register H	FLAPH	00H	8
F00C5H	Flash memory sequencer control register	FSSQ	00H	1, 8
F00C6H	Flash end address pointer register L	FLSEDL	0000H	16
F00C8H	Flash end address pointer register H	FLSEDH	00H	8
F00C9H	Flash registers initialization register	FLRST	00H	1, 8
F00CAH	Flash memory sequencer status register L	FSASTL	Undefined	1, 8
F00CBH	Flash memory sequencer status register H	FSASTH	00H	1, 8
F00CCH	Flash write buffer register L	FLWL	0000H	16
F00CEH	Flash write buffer register H	FLWH	0000H	16
FFFB0H	Flash security flag monitoring register	FLSEC	Undefined	16
FFFB2H	Flash FSW monitoring register S	FLFSWS	Undefined	16
FFFB4H	Flash FSW monitoring register E	FLFSWE	Undefined	16
FFFB6H	Flash memory sequencer initial setting register	FSSET	00H	8
FFFB7H	Extra area sequencer control register	FSSE	00H	1, 8
FFFC0H	Flash protect command register	PFCMD	Undefined	8
FFFC1H	Flash status register	PFS	00H	1, 8
FFFC6H	Flash ECC write buffer register	FLWE	00H	8

33.7.2.1 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 33-9. 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.

33.7.2.2 Flash Programming Mode Control Register (FLPMC)

The FLPMC register controls enable or disable programming of code and data flash memory.

This register can be set by an 8-bit memory manipulation instruction.

The FLPMC register must be set in a specific sequence using PFCMD register (See **33.7.3.1 Procedure for Executing the Specific Flash Memory Sequencer**).

Figure 33-10. Format of Flash Programming Mode Control Register

Address: F00C0H After reset: 08H R/W

Symbol	7	6	5	4	3	2	1	0
FLPMC	FLPMC[7:0]							

Code and Data flash mode	FLPMC register setting
Non-programmable mode (Read mode)	This state is the state after reset. When transitioning from Code flash programming mode and Data flash programming mode, perform in a specific sequence. (See **33.7.3.4 Procedure for Entry to the Non-Programmable Mode**).
Code flash programming mode	This mode can only be transitioned from Read mode. Set FLPMC register in a specific sequence (See **33.7.3.2 Procedure for Entry to the Code Flash Programming Mode**).
Data flash programming mode	This mode can only be transitioned from Read mode. Set FLPMC register in a specific sequence (See **33.7.3.3 Procedure for Entry to the Data Flash Programming Mode**).

33.7.2.3 Flash Area Selection Register (FLARS)

The FLARS register selects flash memory area to programming.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of the FLARS register is 00H under either of the following conditions.

Following a reset.

The value of the FLRST bit of the FLRST register is 1.

Figure 33-11. Format of Flash Area Selection Register (FLARS)

Address: F00C1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLARS	0	0	0	0	0	0	0	EXA

EXA	Selection of the area of flash memory
0	User area
1	Extra area

Caution Bits 1 to 7 are read-only bits and fixed to 0. Writing to these bits are ignored.

33.7.2.4 Flash Address Pointer Register H, L (FLAPH, FLAPL)

The FLAPH and FLAPL registers specify the address where programming the flash memory is to start.

The FLAPH register can be set by an 8-bit memory manipulation instruction, and the FLAPL register can be set by 16-bit memory manipulation instruction.

The values of the FLAPH and FLAPL registers are 00H and 0000H, respectively, under either of the following conditions.

Following a reset.

The value of the FLRST bit of the FLRST register is 1.

In the data flash memory programming case, address should be set within following address area.

RL78/F22: F1000H to F2FFFH

RL78/F25: F1000H to F4FFFH

Figure 33-12. Format of Flash Address Pointer Register H, L (FLAPH, FLAPL)

Address: F00C4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLAPH	0	0	0	0	FLAP19	FLAP18	FLAP17	FLAP16

Address: F00C2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLAPL	FLAP15, FLAP14, ..., FLAP0															

Cautions 1. Values read during the execution of a command for the extra area sequencer are undefined.

2. The settings of the FLAP1 and FLAP0 bits are meaningless during programming of the code flash memory.

3. New values cannot be written to these registers while the flash memory sequencer is operating.

4. Bits 4 to 7 of FLAPH register are read-only bits and fixed to 0. Writing to these bits are ignored.

33.7.2.5 Flash Memory Sequencer Control Register (FSSQ)

The FSSQ register controls the flash memory programming sequencer.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
The value of the FSSQ register is 00H under either of the following conditions.

Following a reset.
The value of the FLRST bit of the FLRST register is 1.

Figure 33-13. Format of Flash Memory Sequencer Control Register (FSSQ)

Address: F00C5H After reset: 00H R/W

Symbol	<7>	<6>	5	4	3	2	1	0
FSSQ	SQST	FSSTP	DCLR	0	MDCH	SQMD2	SQMD1	SQMD0

SQST	Operation control of the code/data flash memory area sequencer
0	The code/data flash memory area sequencer is stopped.
1	The code/data flash memory area sequencer is started.

FSSTP	Forcible termination control of the code/data flash memory area sequencer
0	The code/data flash memory area sequencer is not forcibly terminated.
1	The code/data flash memory area sequencer is forcibly terminated.

DCLR	Operation stop control of the ECC area sequencer
0	The ECC area sequencer is operating.
1	The ECC area sequencer is stopped.

MDCH	SQMD2	SQMD1	SQMD0	Control of the code/data flash memory sequencer
0	0	0	0	No operation
0	0	0	1	Code/data flash memory write
0	0	1	1	Code flash memory blank check
1	0	1	1	Data flash memory blank check
0	1	0	0	Code/data flash memory block erase
Other than above				Setting prohibited

Caution Be sure to set bit 4 of the FSSQ register to 0.

33.7.2.6 Flash End Address Pointer Register H, L (FLSEDH, FLSEDL)

The FLSEDH and FLSEDL registers specify the address where programming the flash memory is to end.

The FLSEDH register can be set by an 8-bit memory manipulation instruction, and the FLSEDL register can be set by 16-bit memory manipulation instruction.

The values of the FLSEDH and FLSEDL registers are 00H and 0000H, respectively, under either of the following conditions.

Following a reset.

The value of the FLRST bit of the FLRST register is 1.

In the data flash memory programming case, address should be set within following address area.

RL78/F22: F1000H to F2FFFH

RL78/F25: F1000H to F4FFFH

Figure 33-14. Format of Flash End Address Pointer Register H, L (FLSEDH, FLSEDL)

Address: F00C8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLSEDH	0	0	0	0	EWA19	EWA18	EWA17	EWA16

Address: F00C6H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLSEDL	EWA15, EWA14, ……, EWA0															

Cautions 1. Values read during the execution of a command for the extra area sequencer are undefined.

2. The settings of the EWA1 and EWA0 bits are meaningless during programming of the code flash memory.

3. New values cannot be written to these registers while the flash memory sequencer is operating.

4. Bits 4 to 7 of FLSEDH register are read-only bits and fixed to 0. Writing to these bits are ignored.

33.7.2.7 Flash Registers Initialization Register (FLRST)

The FLRST register controls initialization of flash memory registers.

This register can be set by a 1-bit or 8-bit memory manipulation instruction. When the FLRST bit is 1, write operation to the registers is ignored.

Reset signal generation sets this register to 00H.

Figure 33-15. Format of Flash Registers Initialization Register (FLRST)

Address: F00C9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLRST	0	0	0	0	0	0	0	FLRST

FLRST	Control of initializing the registers Note
0	The registers are not initialized.
1	The registers are initialized.

Note For details on how to handle the FLRST register, see **33.7.4 Clearing the Registers for Use with the Flash Memory Sequencer**.

Cautions 1. The registers below are initialized with the use of this register.

FLAPH, FLAPL, FLSEDH, FLSEDL, FLWH, FLWL, FLARS, FSSQ, and FSSE register

2. Bits 1 to 7 are read-only bits and fixed to 0. Writing to these bits are ignored.

33.7.2.8 Flash Memory Sequencer Status Register H, L (FSASTH, FSASTL)

The FSASTH and FSASTL registers specify the status of programming and erasure operation.
The FSASTH and FSASTL register can be read by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to 00H.

Figure 33-16. Format of Flash Memory Sequencer Status Register H, L (FSASTH, FSASTL)

Address: F00CBH After reset: 00H R

Symbol	<7>	<6>	5	4	3	2	1	0
FSASTH	ESQEND	SQEND	0	0	0	0 Note 1	0 Note 1	0

Address: F00CAH After reset: x000_0000B Note 2 R

Symbol	7	6	5	4	3	2	1	0
FSASTL	MBTSEL	MOPEN	ESEQER	SEQER	BLER	0	WRER	ERER

ESQEND	Status flag indicating whether the extra area sequencer is stopped
0	The extra area sequencer is operating. The ESQST bit of the FSSE register is cleared to 0 after the operation is completed.
1	The extra area sequencer has finished operating.

SQEND	Status flag indicating whether the code/data flash memory area sequencer is stopped
0	The code/data flash memory area sequencer is operating. The SQST bit of the FSSQ register is cleared to 0 after the operation is completed.
1	The code/data flash memory area sequencer has finished operating.

MBTSEL	Boot flag monitoring bit
0	FLSEC.BTFLG = 1 (the boot program area is not swapped)
1	FLSEC.BTFLG = 0 (the boot program area is swapped)

MOPEN	Code/data flash memory area sequencer operation status flag
0	The code/data flash memory area sequencer is stopped.
1	The code/data flash memory area sequencer is operating.

ESEQER	Error flag of the extra area sequencer
0	No error has occurred.
1	An error has occurred.
Clearing condition: Next activation of the extra area sequencer.	

SEQER	Error flag of the flash memory sequencer
0	No error has occurred.
1	An error has occurred.
Clearing condition: Next activation of the flash memory sequencer.	

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

BLER	Error flag of the blank check command
0	No error has occurred.
1	An error has occurred.
Clearing condition: Activation of the next command action.	

WRER	Error flag of the write command
0	No error has occurred.
1	An error has occurred.
Clearing condition: Activation of the next command action.	

ERER	Error flag of the block erase command
0	No error has occurred.
1	An error has occurred.
Clearing condition: Next activation of the flash memory sequencer. Forcible termination of the command during erasing leads to values read being undefined.	

Notes 1. When reading, read value is undefined.

2. The initial value of the MBTSEL bit is undefined because it depends on the value of the BTFLG bit (boot area switching flag) stored in the extra area.

33.7.2.9 Flash Write Buffer Register H, L (FLWH, FLWL)

The FLWH and FLWL registers hold data to be written during programming of the flash memory.
The FLWH and FLWL registers can be set by a 16-bit memory manipulation instruction.
The values of each of the FLWH and FLWL register is 0000H under any of the following conditions.

Following a reset.
The value of the FLRST bit of the FLRST register is 1.
The flash memory sequencer has finished operating.

Writing to these registers is not possible while the value of the FLRST bit is 1.
Set data to be written to the data flash memory in the 8 lower bits of the FLWL register.

Figure 33-17. Format of Flash Write Buffer Register H, L (FLWH, FLWL)

Address: F00CEH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLWH	FLW31, FLW30, ……, FLW16															

Address: F00CCH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLWL	FLW15, FLW14, ……, FLW0															

Remark The FLWH and FLWL registers can be used as flash data register for code flash ECC diagnostic self-test to inject intentional ECC error. For more detail, see **29.3.5 Code Flash Memory ECC Function**.

33.7.2.10 Flash Protect Command Register (PFCMD)

The PFCMD register is used to protect data in the protected registers of flash programming operation.

This register can be set by an 8-bit memory manipulation instruction.

Figure 33-18. Format of Flash Protect Command Register (PFCMD)

Address: FFFC0H After reset: Undefined W

Symbol	7	6	5	4	3	2	1	0
PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

To write the protected register, following sequences are required. If the sequences are incorrect, the write access is not performed and the FPRERR bit in the PFS register is set to 1.

- Step 1: Write A5H value to the PFCMD register
- Step 2: Write a data to the protected register
- Step 3: Write the data which is inverted of a data to the protected register
- Step 4: Write a data to the protected register, complete write operation

Remark Protected register is follows.
FLPMC register

33.7.2.11 Flash Status Register (PFS)

The PFS register shows the status of protection register access.

This register can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H. The FPRERR flag is cleared when execution of the specific sequence is re-started.

Figure 33-19. Format of Flash Status Register (PFS)

Address: FFFC1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
PFS	0	0	0	0	0	0	0	FPRERR

FPRERR	Protection error flag
0	No error has occurred.
1	An error has occurred.

Remark Protected register is follows.
FLPMC register

33.7.2.12 Flash ECC Write Buffer Register (FLWE)

When the DCLR bit of the FSSQ register is 0, the FLWE register stores ECC data used in flash memory programming. When the DCLR bit is 1, any ECC programing data can be written for ECC diagnosis.

The lower 7-bit data is used at code flash memory programming, and the lower 4-bit data is used at data flash memory programming.

This register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 33-20. Format of Flash ECC Write Buffer Register (FLWE)

Address: FFFC6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLWE	0	FLWE6	FLWE5	FLWE4	FLWE3	FLWE2	FLWE1	FLWE0

Caution Bits 7 are read-only bits and fixed to 0. Writing to these bits are ignored.

Remark This register can be used as flash data register for code flash ECC diagnostic self-test to inject intentional ECC error. For more detail, see **29.3.5 Code Flash Memory ECC Function**.

33.7.2.13 Flash Security Flag Monitoring Register (FLSEC)

The FLSEC register shows the monitor flag data in the extra area.

This register can be read by a 16-bit memory manipulation instruction.

Figure 33-21. Format of Flash Security Flag Monitoring Register (FLSEC)

Address: FFFB0H After reset: Undefined [Note] R

Symbol	15	14	13	12	11	10	9	8
FLSEC	TEPR	1	1	WRPR	1	SEPR	BTPR	BTFLG

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

TEPR	Test mode entry flag
0	Test mode is prohibited.
1	Test mode is enabled.

WRPR	Write-prohibited flag
0	Write is prohibited.
1	Write is enabled.

SEPR	Block erase-prohibited flag
0	Block erase is prohibited.
1	Block erase is enabled.

BTPR	Boot area rewrite-prohibited flag
0	Rewriting of the boot area is prohibited.
1	Rewriting of the boot area is enabled.

BTFLG	Boot area switching flag
0	The boot area is boot cluster 1.
1	The boot area is boot cluster 0.

Note Bits 8 to 15 value after reset depends on the initial value of the security flag setting using the extra area sequencer control register (FSSE).

Cautions 1. Bits 8 to 15 value after reset depends on the initial value of the security flag setting using the FSSE register.

2. When reading bits 11, 13 and 14, read value is undefined.

33.7.2.14 Flash FSW Monitoring Register S (FLFSWS)

The FLFSWS register shows the FSWS data in the extra area.
This register can be read by a 16-bit memory manipulation instruction.

Figure 33-22. Format of Flash FSW Monitoring Register S (FLFSWS)

Address: FFFB2H After reset: Undefined Note 1 R

Symbol	15	14	13	12	11	10	9	8
FLFSWS	0	0	0	0	0	0	0	FSWS8

7	6	5	4	3	2	1	0
FSWS7	FSWS6	FSWS5	FSWS4	FSWS3	FSWS2	FSWS1	FSWS0

FSWS[8:0]	Start block number specified for flash memory shield area
-	Start block number (start block number in the window range) Note 2

Notes 1. Bits 0 to 8 value after reset is depending on initial value of the flash shield window setting using extra area sequencer control register (FSSE).

2. For details, see **Table 33-13. Relationship between Flash Shield Window Function Setting/Change Methods and Commands**.

33.7.2.15 Flash FSW Monitoring Register E (FLFSWE)

The FLFSWE register shows the FSWE data in the extra area.
This register can be read by a 16-bit memory manipulation instruction.

Figure 33-23. Format of Flash FSW Monitoring Register E (FLFSWE)

Address: FFFB4H After reset: Undefined Note 1 R

Symbol	15	14	13	12	11	10	9	8
FLFSWE	0	0	0	0	0	0	0	FSWE8

7	6	5	4	3	2	1	0
FSWE7	FSWE6	FSWE5	FSWE4	FSWE3	FSWE2	FSWE1	FSWE0

FSWE[8:0]	End block number specified for flash memory shield area
–	End block number (end block number in the window range + 1) Note 2

Notes 1. Bits 0 to 8 value after reset is depending on initial value of the flash shield window setting using extra area sequencer control register (FSSE).

2. The setting during serial programming is different from this. For details, see **Table 33-13. Relationship between Flash Shield Window Function Setting/Change Methods and Commands**.

33.7.2.16 Flash Memory Sequencer Initial Setting Register (FSSET)

The FSSET register is used to control the flash memory programming sequencer.
This register can be set by an 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Figure 33-24. Format of Flash Memory Sequencer Initial Setting Register (FSSET)

Address: FFFB6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FSSET	TMSPMD	TMBTSEL	0	FSET4	FSET3	FSET2	FSET1	FSET0

TMSPMD	Specification for boot swapping Note 1
0	Follows the information in the extra area.
1	Follows the setting of the TMBTSEL bit.

TMBTSEL	Setting for temporary boot swapping Note 1
0	Specifies boot cluster 0 as the boot area (boot swapping does not proceed).
1	Specifies boot cluster 1 as the boot area (boot swapping proceeds).

FSET[4:0]	Setting of the operating frequency of the flash memory sequencer
–	Set the operating frequency of the flash memory sequencer. See the table below for the correspondence between the operating frequency of the flash memory sequencer and the setting of the FSET[4:0] bits.

Operating Frequency of the Flash Memory Sequencer and the Setting of the FSET[4:0] bits.

FSET[4:0]	Frequency	FSET[4:0]	Frequency	FSET[4:0]	Frequency	FSET[4:0]	Frequency
00000B	(Note 2)	01000B	9 MHz	10000B	17 MHz	11000B	26 MHz
00001B	2 MHz	01001B	10 MHz	10001B	18 MHz	11001B	28 MHz
00010B	3 MHz	01010B	11 MHz	10010B	19 MHz	11010B	30 MHz
00011B	4 MHz	01011B	12 MHz	10011B	20 MHz	11011B	32 MHz
00100B	5 MHz	01100B	13 MHz	10100B	21 MHz	11100B	34 MHz
00101B	6 MHz	01101B	14 MHz	10101B	22 MHz	11101B	36 MHz
00110B	7 MHz	01110B	15 MHz	10110B	23 MHz	11110B	38 MHz
00111B	8 MHz	01111B	16 MHz	10111B	24 MHz	11111B	40 MHz

Notes 1. If the BTPR bit in the FLSEC register is 0 (rewriting of the boot area is disabled), the TMSPMD and TMBTSEL bits cannot be set.

2. When setting the flash memory control mode, set the same frequency as the f_{CLK}. Do not set 00000B to these bits.

Caution Set the value corresponding to that obtained by rounding the CPU operating frequency up to the nearest whole number in the FSET[4:0] bits. For example, when the CPU operating frequency is 9.7 MHz, set the bits for 10 MHz.
If the value of FSET[4:0] bits are 00001B (2 MHz), 00010B (3 MHz) and 00011B (4 MHz), be sure to use the same operating frequency (f_{CLK}).

Remark f_{CLK}: CPU/peripheral hardware clock frequency.

33.7.2.17 Extra Area Sequencer Control Register (FSSE)

The FSSE register controls the flash memory extra area programming sequencer.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
The value of the FSSE register is 00H under either of the following conditions.

Following a reset.
The value of the FLRST bit of the FLRST register is 1.

Figure 33-25. Format of Extra Area Sequencer Control Register (FSSE)

Address: FFFB7H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
FSSE	ESQST	0	0	0	0	ESQMD2	ESQMD1	ESQMD0

ESQST	Operation control of the extra area sequencer
0	The extra area sequencer is stopped.
1	The extra area sequencer is started.

ESQMD2	ESQMD1	ESQMD0	Control of the extra area sequencer
0	0	0	No operation
0	0	1	Security flag setting
0	1	0	Flash shield window setting
Other than above			Setting prohibited

Cautions 1. Initialization by setting the FLRST bit to 1 is only enabled while the flash memory sequencer is stopped (both the SQEND and ESQEND bits of the FSASTH register have the setting 0).

2. To write the extra area, set the EXA bit of the FLARS register to 1 and set the data to be written in the FLWH and FLWL registers before activating the extra area sequencer.

3. Values read from the FLAPL, FLWH, FLWL, and FSSQ registers are undefined after activation of the extra area sequencer.

4. Bits 3 to 6 are read-only bits and fixed to 0. Writing to these bits are ignored.

33.7.3 Setting the Flash Memory Control Mode

Execution of the specific sequence for use with the flash memory sequencer enables setting the flash memory control mode to the states where the code or data flash memory area or neither of them can be rewritten.

- State where the code flash memory (and extra area) can be rewritten: Code flash programming mode
- State where the data flash memory can be rewritten: Data flash programming mode
- State where the flash memory (and extra area) cannot be rewritten: Non-programmable mode

Cautions 1. For handling of the data flash memory area, follow the procedure while access to the data flash memory is enabled (the value of the DFLEN bit of the DFLCTL register is 1).
2. The high-speed on-chip oscillator should be kept operating during self-programming.
3. Set the operating frequency of the flash memory sequencer in the FSSET register.

33.7.3.1 Procedure for Executing the Specific Flash Memory Sequencer

Writing the required values to the flash programming mode control register (FLPMC register) by following steps 1 to 4 below enables the transitions to each of the flash memory control modes.

<1> Write A5H to the PFCMD register.
<2> Write the value to be set to the FLPMC register.
<3> Write the inverse of the value to be set to the FLPMC register.
<4> Write the value to be set to the FLPMC register.

- The specific sequence can only be executed while the value of the FLRST bit of the FLRST register is 0 and the flash memory sequencer is stopped.
- If writing to any other memory area or register is attempted in the intervals between steps 1 to 4 during execution of the specific sequence, a protection error occurs, writing to the specified register does not proceed, and the FPRERR flag of the flash status register (PFS) is set to 1. The FPRERR flag is cleared following a reset or when execution of the specific sequence is re-started.

33.7.3.2 Procedure for Entry to the Code Flash Programming Mode

The procedure for entry to the code flash programming mode is shown below.

<1> Write A5H to the PFCMD register.
<2> Write 02H to the FLPMC register.
<3> Write FDH to the FLPMC register (inverse of 02H).
<4> Write 02H to the FLPMC register.

33.7.3.3 Procedure for Entry to the Data Flash Programming Mode

The procedure for entry to the data flash programming mode is shown below.

<1> Write A5H to the PFCMD register.
<2> Write 10H to the FLPMC register.
<3> Write EFH to the FLPMC register (inverse of 10H).
<4> Write 10H to the FLPMC register.

33.7.3.4 Procedure for Entry to the Non-Programmable Mode

The procedure for transitioning from code flash memory programming mode or data flash memory programming mode to non-programmable mode is shown below.

<1> Write A5H to the PFCMD register.
<2> Write 08H to the FLPMC register.
<3> Write F7H to the FLPMC register (inverse of 08H).
<4> Write 08H to the FLPMC register.
<5> Waiting for mode setup time **Note**

Note The mode setup time is as follows.

- When transitioning from code flash memory programming mode: 15 μs
- When transitioning from data flash memory programming mode: 10 μs

33.7.4 Clearing the Registers for Use with the Flash Memory Sequencer

The following registers can be cleared by setting the FLRST bit of the flash registers initialization register (FLRST) to 1.

Target registers: FLAPH, FLAPL, FLSEDH, FLSEDL, FLWH, FLWL, FLARS, FSSQ, FSSE

The procedure for clearing the target registers is shown below.

<1> Set the FLRST bit.

<2> Use software code to wait for at least one cycle of the CPU operating clock.

<3> Clear the FLRST bit.

33.7.5 Setting the Operating Frequency of the Flash Memory Sequencer

Set the value corresponding to the operating frequency of the CPU (2 MHz to 40 MHz) in the FSET[4:0] bits of the flash memory sequencer initial setting register (FSSET). When making the setting, round the CPU operating frequency value up to the nearest whole number.

(Example: When the CPU operating frequency is 9.7 MHz, set the bits for 10 MHz.)

How to set the operating frequency of the flash memory sequencer is described below.

<1> Enter the code flash programming mode or data flash programming mode. For the procedures for entry to each of these flash programming modes, see **33.7.3.1 Procedure for Executing the Specific Flash Memory Sequencer**, **33.7.3.2 Procedure for Entry to the Code Flash Programming Mode**, and **33.7.3.3 Procedure for Entry to the Data Flash Programming Mode**.

<2> After reading from the flash memory sequencer initial setting register (FSSET), set the TMSPMD and TMBTSEL bits to the same values as those read from the FSSET register and the FSET[4:0] bits to the value corresponding to the CPU operating frequency, respectively.

Caution When using the flash memory sequencer to rewrite the code or data flash memory or the extra area, set the value corresponding to the CPU operating frequency in the FSET[4:0] bits of the FSSET register before proceeding.
Note that if rewriting of any of these areas is attempted while the value corresponding to the CPU operating frequency is not correct, operation is undefined and written data are not guaranteed. Even if the values in the flash memory are as expected immediately after rewriting, retaining the values for any specified period is not guaranteed.

33.7.6 Commands for Use with the Flash Memory Sequencer in the Respective Areas

33.7.6.1 Overview

The flash memory sequencer serves as a code/data flash memory area sequencer or an extra area sequencer. In the former role, it is used to rewrite the code flash memory area or data flash memory area, while in the latter role it is used to rewrite the extra area. To rewrite a given area, execute the corresponding commands for use with the flash memory sequencer.

33.7.6.2 Selecting the Area to be Rewritten

Use the flash area selection register (FLARS) to select the user area or extra area for rewriting to the code/data flash memory area or extra area, respectively.

33.7.6.3 Commands for Use with the Code/Data Flash Memory Area Sequencer

Use the commands that are exclusively for use with the code/data flash memory area sequencer to rewrite the code or data flash memory area. Issuing a command requires entering the target command number in the bits SQMD[2:0] and MDCH of the flash memory sequencer control register (FSSQ) and setting the SQST bit after or at the same time as that. Do not set the SQST bit before setting the SQMD[2:0] and MDCH bits.

For the commands exclusively for use with the code/data flash memory area sequencer, see **33.7.2.5 Flash Memory Sequencer Control Register (FSSQ)**.

33.7.6.4 Operations for Rewriting the Code Flash Memory Area

To rewrite the code flash memory area, enter the code flash programming mode and then execute commands for use with the code/data flash memory area sequencer. Before starting to execute a command, set the data required for execution, such as specifying an address, addresses, or data, in the corresponding registers.

Units for block erasure and for writing in rewriting of the code flash memory area:

- Unit for block to be erased: 2 Kbyte
- Unit for writing: 1 word (4 bytes)

<Handling the commands>

The commands to used are block erase, write and blank check of the code flash memory area.

- Enter the code flash programming mode. For the procedure for entry to the code flash programming mode, see **33.7.3.1 Procedure for Executing the Specific Flash Memory Sequencer** and **33.7.3.2 Procedure for Entry to the Code Flash Programming Mode**.
- FLARS register = 00H (EXA bit = 0): Setting to select the user (not extra) area
- Set the specified data in the corresponding registers before executing the individual commands.

(1) Block erasure:

FLAPH and FLAPL registers: Block start address in the code flash memory (example: 0x002000)
FLSEDH and FLSEDL registers: Block end address in the code flash memory (example: 0x0027FF)

(2) Writing: As writing proceeds in one-word (four-byte) units, set the lower 2 bits in the FLAPL register to 00B.

FLAPH and FLAPL registers: Start address in the target flash memory area (example: 0x002000)
FLSEDH and FLSEDL registers: Set all bits to 0 or do not set them (example: 0x000000).
FLWH and FLWL registers: One-word (four-byte) values to be written.

(3) Blank checking: As blank checking proceeds in one-word (four-byte) units, set the lower 2 bits in the FLAPL register to 00B. Also, set the lower 2 bits in the FLSEDL register to 11B.

FLAPH and FLAPL registers: Start address in the target flash memory area (example: 0x002000)
FLSEDH and FLSEDL registers: End address in the target flash memory area (example: 0x0027FF)

Note. When blank checking is only to be applied to one word (four bytes), set the FLSEDH and FLSEDL registers to the same values as those in the FLAPH and FLAPL registers.

- After issuing a command for use with the code/data flash memory area sequencer, wait for the completion of its execution. For details on the procedure for waiting for the completion of command execution, see the section titled "Procedure for checking the completion of commands for use with the code/data flash memory area sequencer" in **33.7.6.9 Procedure for Checking Completion of the Commands for Use with the Flash Memory Sequencer in the Respective Areas**.
- Processing after executing a command
  When command processing is to continue:
  The same command with the target registers set to updated values or a rewrite command for any other area in the code flash memory can be executed with the state remaining in code flash programming mode.
  When command processing has been completed:
  Switch to the non-programmable mode. For the procedure for switching to the non-programmable mode, see **33.7.3.1 Procedure for Executing the Specific Flash Memory Sequencer** and **33.7.3.4 Procedure for Entry to the Non-Programmable Mode**.

33.7.6.5 Operations for Rewriting the Data Flash Memory Area

To rewrite the data flash memory area, enter the data flash programming mode and then execute commands for use with the code/data flash memory area sequencer. Before starting to execute a command, set the data required for execution, such as specifying an address, addresses, or data, in the corresponding registers.

Units for block erasure and for writing in rewriting of the data flash memory area:

- Unit for block to be erased: 1 Kbyte
- Unit for writing: 1 word (1 byte)

<Handling the commands>

The commands to used are block erase, write and blank check of the data flash memory area.

- Enter the data flash programming mode. For the procedure for entry to the data flash programming mode, see **33.7.3.1 Procedure for Executing the Specific Flash Memory Sequencer** and **33.7.3.3 Procedure for Entry to the Data Flash Programming Mode**.
- FLARS register = 00H (EXA bit = 0): Setting to select the user (not extra) area
- Set the specified data in the corresponding registers before executing the individual commands.

(1) Block erasure:

FLAPH and FLAPL registers: Block start address in the data flash memory (example: 0x0F1000)
FLSEDH and FLSEDL registers: Block end address in the data flash memory (example: 0x0F13FF)

(2) Writing: 1 byte:

FLAPH and FLAPL registers: Start address in the target flash memory area (example: 0x0F1101)
FLSEDH and FLSEDL registers: Set all bits to 0 or do not set them (example: 0x000000).
FLWH and FLWL registers: Set a value to be written in the range from 0x00000000 to 0x000000FF, since only the FLW7 to FLW0 bits are valid.

(3) Blank checking:

FLAPH and FLAPL registers: Start address in the target flash memory area (example: 0x0F1000)
FLSEDH and FLSEDL registers: End address in the target flash memory area (example: 0x0F13FF)

Note. When blank checking is only to be applied to one byte, set the FLSEDH and FLSEDL registers to the same values as those in the FLAPH and FLAPL registers.

- After issuing a command for use with the code/data flash memory area sequencer, wait for the completion of its execution. For details on the procedure for waiting for the completion of command execution, see the section titled "Procedure for checking the completion of commands for use with the code/data flash memory area sequencer" in **33.7.6.9 Procedure for Checking Completion of the Commands for Use with the Flash Memory Sequencer in the Respective Areas**.
- Processing after executing a command
  When command processing is to continue:
  The same command with the target registers set to updated values or a rewrite command for any other area in the data flash memory can be executed with the state remaining in data flash programming mode.
  When command processing has been completed:
  Switch to the non-programmable mode. For the procedure for switching to the non-programmable mode, see **33.7.3.1 Procedure for Executing the Specific Flash Memory Sequencer** and **33.7.3.4 Procedure for Entry to the Non-Programmable Mode**.

33.7.6.6 Commands for Use with the Extra Area Sequencer

To rewrite the extra area, enter the code flash programming mode and then use commands that are exclusively for use with the extra area sequencer. Issuing a command requires entering the target command number in the ESQMD[2:0] bits of the extra area sequencer control register (FSSE) and setting the ESQST bit after or at the same time as that. Do not set the ESQST bit before the ESQMD[2:0] bits.

33.7.6.7 Operations for Rewriting the Extra Area

To rewrite the extra area, enter the code flash programming mode and then execute commands for use with the extra area sequencer. Before starting to execute a command, set the data required for execution in the corresponding registers.

Unit for writing in rewriting of the extra area:

- Unit for writing: 1 word (4 bytes)

<Handling the commands>

The target commands are for writing data to the extra area.

- Enter the code flash programming mode. For the procedure for entry to the code flash programming mode, see **33.7.3.1 Procedure for Executing the Specific Flash Memory Sequencer** and **33.7.3.2 Procedure for Entry to the Code Flash Programming Mode**.
- FLARS register = 01H (EXA bit = 1): Setting to select the extra (not user) area
- Before executing a command, set a one-word (four-byte) value in the FLWH and FLWL registers. Specifically, set the value to be written to the target extra area data EX bits 31 to 0 in the FLW[31:0] bits of the FLWH and FLWL registers.
- Specifying a command determines the area to which data are to be written. Enter the target command number in the ESQMD[2:0] bits of the FSSE register and also set the ESQST bit of the same register to 1.

(1) For programming of the security flags and boot area switching flag: 81H
(2) For programming of the FSW-related data: 82H

- After issuing a command for use with the extra area sequencer, wait for the completion of its execution. For details on the procedure for waiting for the completion of command execution, see **33.7.6.9 Procedure for Checking Completion of the Commands for Use with the Flash Memory Sequencer in the Respective Areas**.
- Processing after executing a command
  When command processing is to continue:
  The same command with the target registers set to updated values or a rewrite command for any other area in the extra area can be executed with the state remaining in code flash programming mode.
  When command processing has been completed:
  Switch to the non-programmable mode. For the procedure for switching to the non-programmable mode, see **33.7.3.1 Procedure for Executing the Specific Flash Memory Sequencer** and **33.7.3.4 Procedure for Entry to the Non-Programmable Mode**.

33.7.6.8 Data to be Set for the Commands for Use with the Extra Area Sequencer

Writing to the extra area proceeds per word (four bytes), including values that are not to be changed.

Before executing a command, set the value to be set in the extra area data EX bits 31 to 0 for each target command to be executed in the FLW[31:0] bits of the FLWH and FLWL registers.

(1) Programming of the Security flags and Boot Area Switching Flag

Set the value to be set in the extra area data EX bits 31 to 0 shown below in the FLW31 to FLW0 bits of the FLWH and FLWL registers.

EX bit 31	EX bit 30	EX bit 29	EX bit 28	EX bit 27	EX bit 26	EX bit 25	EX bit 24
1	1	1	1	1	1	1	1

EX bit 23	EX bit 22	EX bit 21	EX bit 20	EX bit 19	EX bit 18	EX bit 17	EX bit 16
1	1	1	1	1	1	1	1

EX bit 15	EX bit 14	EX bit 13	EX bit 12	EX bit 11	EX bit 10	EX bit 9	EX bit 8
TEPR	1	1	WRPR	1	SEPR	BTPR	BTFLG

EX bit 7	EX bit 6	EX bit 5	EX bit 4	EX bit 3	EX bit 2	EX bit 1	EX bit 0
1	1	1	1	1	1	1	1

- The value to be set in the TEPR bit (bit 15) controls the test mode entry.
  TEPR = 0: Test mode entry is disabled.
  TEPR = 1: Test mode entry is enabled.
- The value to be set in the WRPR bit (bit 12) controls the prohibition of writing in serial programming mode.
  WRPR = 0/1 (at shipment): Writing in serial programming mode is prohibited/writing in serial programming mode is enabled.
- The value to be set in the SEPR bit (bit 10) controls the prohibition of block erasure in serial programming mode.
  SEPR = 0/1 (at shipment): Block erasure in serial programming mode is prohibited/block erasure in serial programming mode is enabled.
- The value to be set in the BTPR bit (bit 9) controls the prohibition of rewriting in the boot area through serial programming and self-programming.
  BTPR = 0/1 (at shipment): Rewriting in the boot area is prohibited/rewriting in the boot area is enabled.
- The value to be set in the BTFLG bit (bit 8) is for control of the boot cluster to be set as the boot area when TMSPMD = 0; that is, boot swapping follows the information in the extra area (the BTFLG setting).
  BTFLG = 0/1 (at shipment): The boot area is boot cluster 1/the boot area is boot cluster 0.

Cautions 1. When changing the value of the BTFLG bit, set all other bits to 1.

2. When changing the value of security flags other BTFLG bit to 0, read the register first and set the BTFLG bit to the same value as was read, and set the other bits to 1.

3. When setting the WRPR bit to 0, the WRPR bit can only be set to 1 by executing the chip erase command in serial programmer mode.
Note that if either of the prohibition settings listed below is made, executing the chip erase command in serial programming mode is not possible.
- **SEPR = 0 (Block erasure is prohibited.)**
- **BTPR = 0 (Rewriting of the boot area is prohibited.)**

(2) Programming of the flash shield window data

Set the value to be set in the extra area data EX bits 31 to 0 shown below in the FLW31 to FLW0 bits of the FLWH and FLWL registers.

EX bit 31	EX bit 30	EX bit 29	EX bit 28	EX bit 27	EX bit 26	EX bit 25	EX bit 24
0	0	0	0	0	0	0	FSWE8

EX bit 23	EX bit 22	EX bit 21	EX bit 20	EX bit 19	EX bit 18	EX bit 17	EX bit 16
FSWE7	FSWE6	FSWE5	FSWE4	FSWE3	FSWE2	FSWE1	FSWE0

EX bit 15	EX bit 14	EX bit 13	EX bit 12	EX bit 11	EX bit 10	EX bit 9	EX bit 8
0	0	0	0	0	0	0	FSWS8

EX bit 7	EX bit 6	EX bit 5	EX bit 4	EX bit 3	EX bit 2	EX bit 1	EX bit 0
FSWS7	FSWS6	FSWS5	FSWS4	FSWS3	FSWS2	FSWS1	FSWS0

- The value to be set in the FSWE8 to FSWE0 bits (bits 24 to 16) is the end block number (end block number in the window range + 1).
- The value to be set in FSWS8 to FSWS0 bits (bits 8 to 0) is the start block number (start block number in the window range).

33.7.6.9 Procedure for Checking Completion of the Commands for Use with the Flash Memory Sequencer in the Respective Areas

Completion of an activated command for use with the flash memory sequencer requires the execution of the specific procedure for checking completion described below.

- Procedure for checking the completion of commands for use with the code/data flash memory area sequencer:
  <Procedure for checking completion>

(1) Wait until the SQEND flag of the FSASTH register is set after activating a command for use with the code/data flash memory area.
(2) Set the FSSQ register to 00H after confirming that the value of the SQEND flag is 1. For the processing time of each command, refer to the Electrical Specifications in this document.
(3) Wait until the SQEND flag is cleared, at which point the procedure is completed. It takes a few clocks (2 CPU clocks) for the SQEND flag to be cleared.

- Procedure for checking the completion of commands for use with the extra area sequencer:
  <Procedure for checking completion>

(1) Wait until the ESQEND flag of the FSASTH register is set after activating a command for use with the extra area sequencer.
(2) Set the FSSE register to 00H after confirming that the value of the ESQEND flag is 1. After the command is executed, the time until the ESQEND flag becomes 1 is TYP. 13.4 ms, MAX. 440.0 ms (condition: f_{CLK} = 8 to 40 MHz).
(3) Wait until the ESQEND flag is cleared, at which point the procedure is completed. It takes a few clocks (2 CPU clocks) for the ESQEND flag to be cleared.

33.7.6.10 Procedure for Forcibly Terminating a Command for Use with the Code/Data Flash Memory Area Sequencer

If an abnormal state arises during the execution of a command for use with the code/data flash memory area sequencer, forcibly terminating the command is possible.

Note that forcibly terminating a command for use with the extra area sequencer while it is in progress is not possible.

<Procedure for forcible termination>

(1) Set the FSSTP bit of the FSSQ register to 1 between activating a command in step 1 and clearing the SQST bit of the FSSQ register in step 2 in the section titled “Procedure for checking the completion of commands for use with the code/data flash memory area sequencer” in **33.7.6.9 Procedure for Checking Completion of the Commands for Use with the Flash Memory Sequencer in the Respective Areas**. This forcibly terminates the command for use with the code/data flash memory area sequencer which was activated.
(2) Clear the SQST and FSSTP bits of the FSSQ register after confirming that the value of the SQEND flag of the FSASTH register is 1. The time from when the FSSTP bit is set to 1 until the SQEND flag becomes 1 is maximum 28.0 μs.
(3) Wait until the SQEND flag is automatically cleared, at which point the procedure is completed. It takes a few clocks (2 CPU clocks) for the SQEND flag to be cleared.

33.7.7 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.

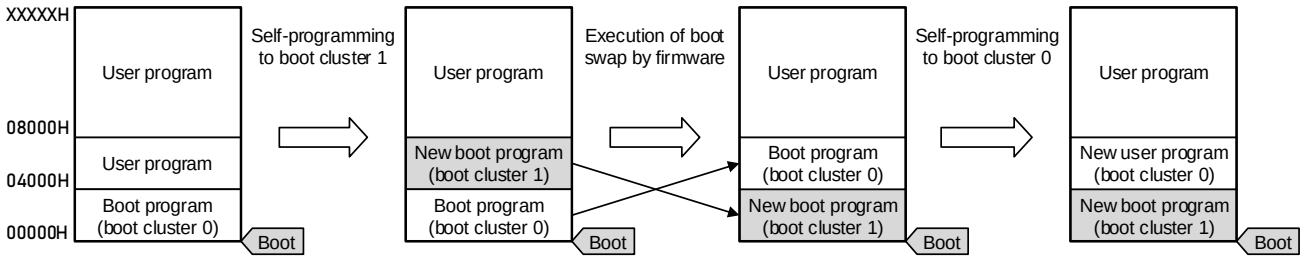
The boot swap function is used to avoid this problem.

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, the boot area changes from boot cluster 0 to boot cluster 1 through self-programming. 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 a 16 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

Figure 33-26. Boot Swap Function

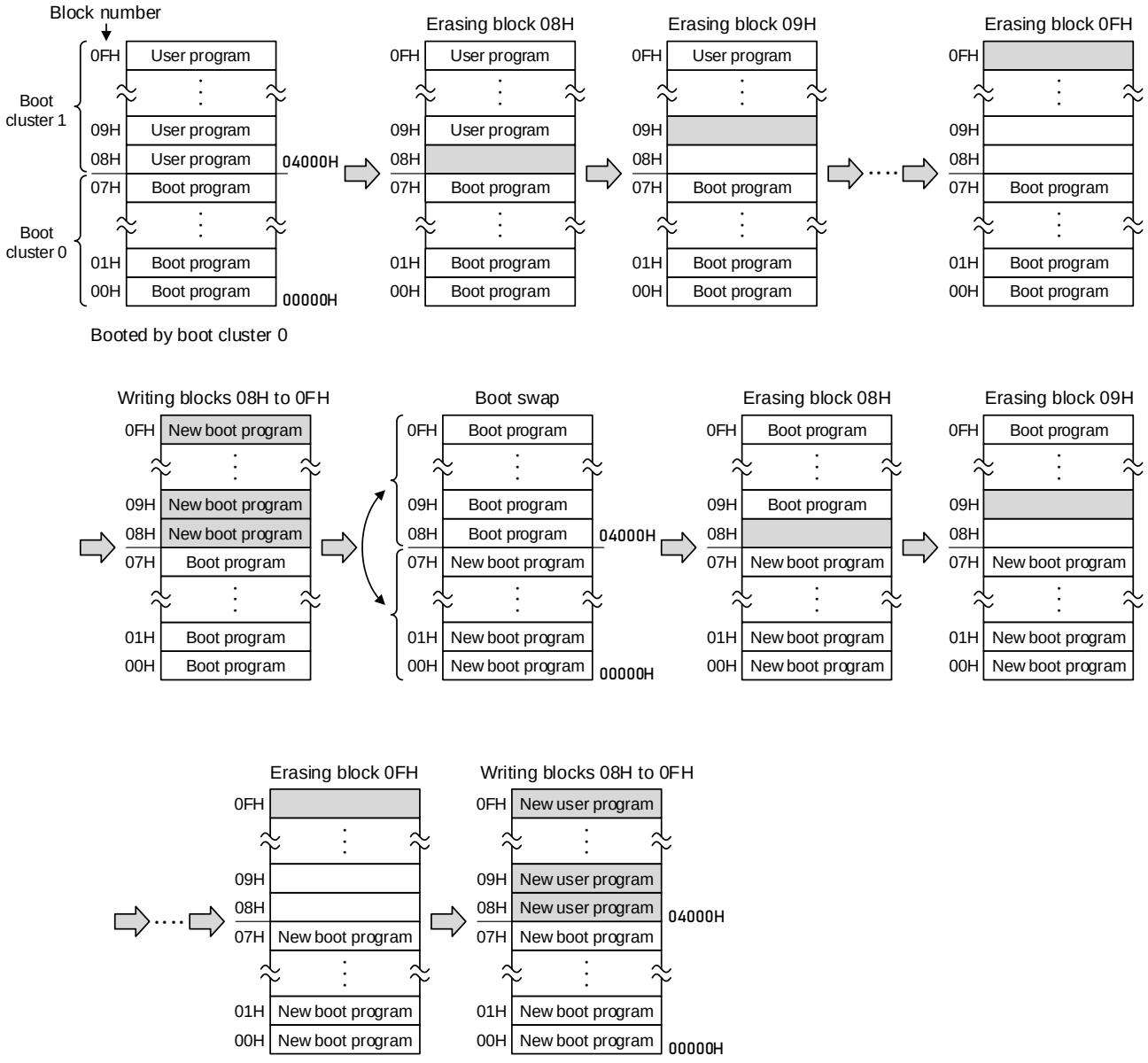


In an example of above figure, it is as follows.

Boot cluster 0: Boot program area before boot swap

Boot cluster 1: Boot program area after boot swap

Figure 33-27. Example of Executing Boot Swapping



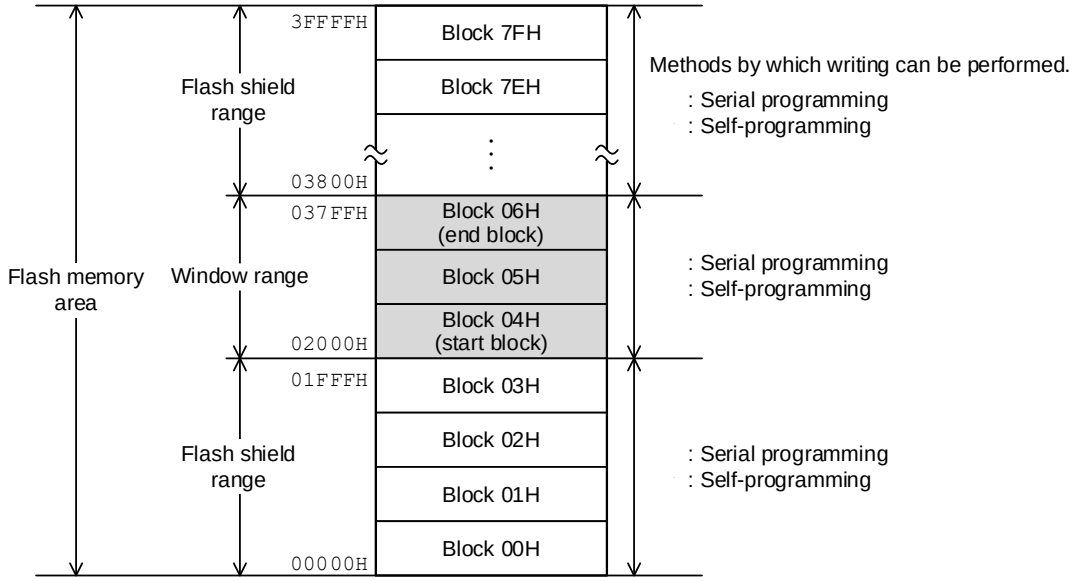
33.7.8 Flash Shield Window Function

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 33-28. Flash Shield Window Setting Example (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 33-13. 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.	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.

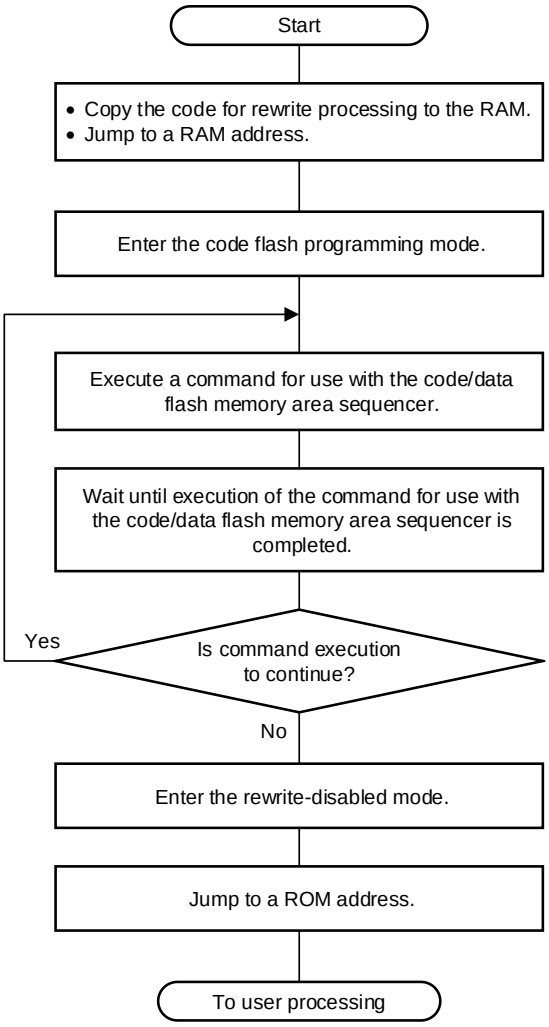
Remark See **33.8 Security Settings** to prohibit writing or erasing during serial programming.

33.7.9 Example of Executing the Commands to Rewrite the Flash Memory Area

(1) Example of Executing the Commands to Rewrite the Code Flash Memory Area

Figure 33-29 shows the flow of executing the commands to rewrite the code flash memory area.

Figure 33-29. Flow of Executing the Commands to Rewrite the Code Flash Memory Area



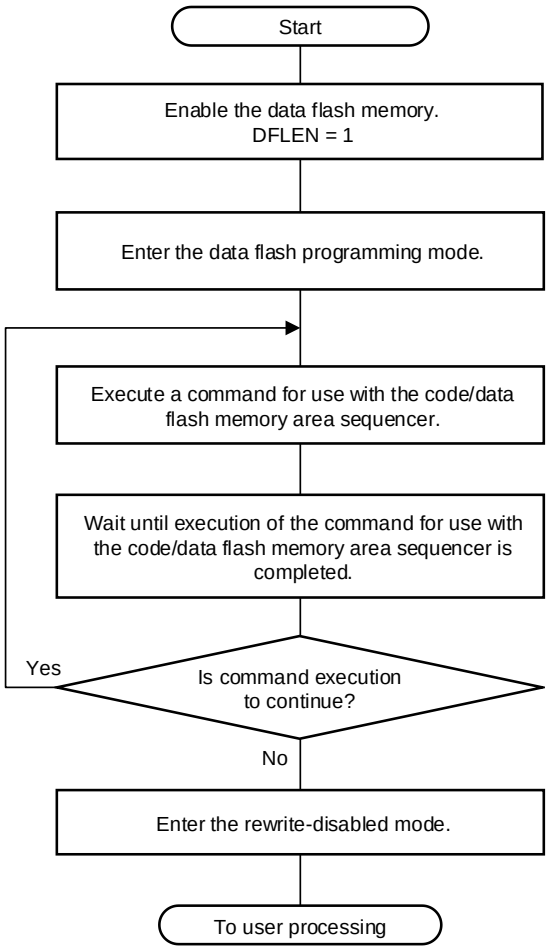
Reference section

- **33.7.3.2 Procedure for Entry to the Code Flash Programming Mode**
- **33.7.5 Setting the Operating Frequency of the Flash Memory Sequencer**
- **33.7.6.4 Operations for Rewriting the Code Flash Memory Area**
- **33.7.6.9 Procedure for Checking Completion of the Commands for Use with the Flash Memory Sequencer in the Respective Areas**
- **33.7.3.4 Procedure for Entry to the Non-Programmable Mode**

(2) Example of Executing the Commands to Rewrite the Data Flash Memory Area

Figure 33-30 shows the flow of executing the commands to rewrite the data flash memory area.

Figure 33-30. Flow of Executing the Commands to Rewrite the Data Flash Memory Area



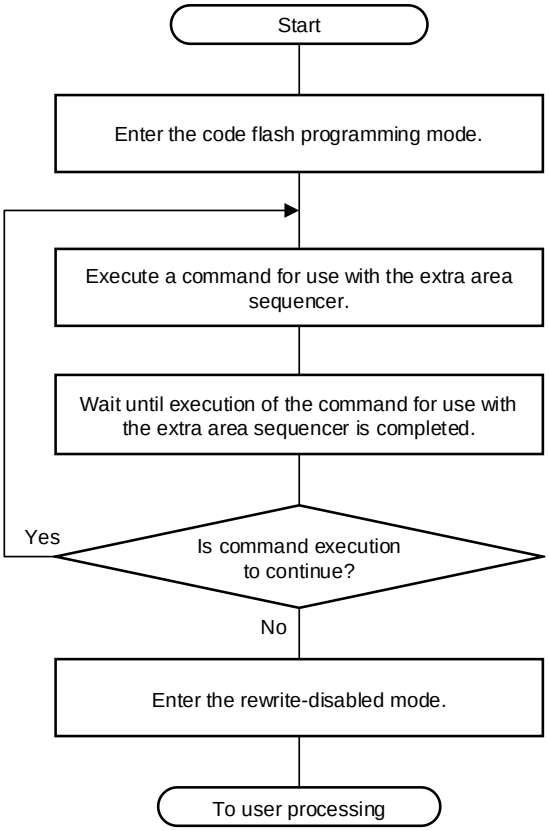
Reference section

- **33.7.3.3 Procedure for Entry to the Data Flash Programming Mode**
- **33.7.5 Setting the Operating Frequency of the Flash Memory Sequencer**
- **33.7.6.5 Operations for Rewriting the Data Flash Memory Area**
- **33.7.6.9 Procedure for Checking Completion of the Commands for Use with the Flash Memory Sequencer in the Respective Areas**
- **33.7.3.4 Procedure for Entry to the Non-Programmable Mode**

(3) Example of Executing the Commands to Rewrite the Extra Area

Figure 33-31 shows the flow of executing the commands to rewrite the extra area.

Figure 33-31. Flow of Executing the Commands to Rewrite the Extra Area



Reference section

- **33.7.3.2 Procedure for Entry to the Code Flash Programming Mode**
- **33.7.5 Setting the Operating Frequency of the Flash Memory Sequencer**
- **33.7.6.7 Operations for Rewriting the Extra Area**
- **33.7.6.9 Procedure for Checking Completion of the Commands for Use with the Flash Memory Sequencer in the Respective Areas**
- **33.7.3.4 Procedure for Entry to the Non-Programmable Mode**

33.7.10 Notes on Self-Programming

(1) Rewriting the code flash memory or extra area

To rewrite the code flash memory or extra area, place the re-programming code in the RAM.

(2) Erasing the code flash memory area

Interrupts should be disabled (DI state) while self-programming is in progress. When suspending self-programming and use interrupts, before enabling interrupts (EI state), move to non-programable mode and mask the code flash memory ECC interrupt.
When the CPU reads the erased flash memory area, the code flash ECC error detection interrupt occurs. Write data to all unused areas in the code flash memory. Also, it is recommended to writing data “FFH” to unused areas. If the CPU fetches the FFH code, an illegal instruction internal reset will occur.

(3) Precondition for manipulating the data flash memory area

Before manipulating the data flash memory area, set the DFLEN bit of the data flash control register (DFLCTL) to 1 (enabling access to the data flash memory).

(4) Execution of programs during rewriting of the flash memory

The flash memory sequencer is used to control rewriting of the flash memory during self-programming.
In the flash memory control modes where rewriting of the flash memory is enabled, reference to the flash memory to be manipulated is not possible.

- In code flash programming mode, reference to the code flash memory is not possible. Accordingly, in code flash programming mode, copy the user program that is to be executed from the ROM (code flash memory) and its data for reference to the RAM in advance so that the program can be executed and reference to the data in the RAM is possible.
- In data flash programming mode, reference to the data flash memory is not possible. Accordingly, in data flash programming mode, copy data that are for reference to the RAM in advance so that reference to the data in the RAM is possible.

(5) The code flash memory ECC circuit must be initialized after it has been successfully written to code flash memory. Reset (external or internal reset) to initialize the code flash memory ECC circuit.

(6) Notes when using on-chip debugger

When using on-chip debugger to break during flash self-programming. Set “Using the flash self programming” to [Yes] in the debug settings of the integrated development environment such as CS+ or e^{2}studio. For details, refer to the User’s Manual of the development environment.

33.8 Security Settings

The RL78/F22 and RL78/F25 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 by serial programming or self-programming.

- 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 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 03FFFH) in the code flash memory is prohibited by this setting.
- Disabling test mode
  Test mode entry can be prohibited. If test mode is disabled, Renesas Electronics will not be able to analyze flash memory using the tester. It can be released using flash memory programmer or self-programming for extra area.

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 33-14 shows the relationship between the erase and write commands when the RL78/F22 or RL78/F25 security function is enabled.

Caution The security function 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 **33.7.8 Flash Shield Window Function** for detail).

Table 33-14. 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.
Prohibition of test mode	Blocks can be erased.	Can be performed.

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.
Prohibition of test mode	Blocks can be erased.	Can be performed.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **33.7.8 Flash Shield Window Function** for detail).

Table 33-15. 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.
Prohibition of test mode		Set via GUI of dedicated flash memory programmer, etc.

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.	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.
Prohibition of test mode		Disable by using self programming.

33.9 Data Flash Memory

33.9.1 Overview of the Data Flash Memory

An overview of the data flash memory is provided below.

- The data flash memory can be rewritten by using self-programming with a user program.
- 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 (background 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 HALT/STOP status is prohibited while rewriting the data flash memory.
- The data flash memory can be programmed 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, wait at least 40 μs in software for oscillation accuracy stabilization time before executing the data flash self-programming.

Remark For details about rewriting the code flash memory by using a user program, see **33.7 Self-Programming**.

33.9.2 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 it can be read/written by a user program using self-programming.

<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: 250 ns
<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). Then, wait at least 40 μs in software for oscillation accuracy stabilization time before executing the data flash self-programming.

CHAPTER 34 ON-CHIP DEBUG FUNCTION

34.1 Overview of On-chip Debug Function

The RL78/F22 and RL78/F25 have enhanced on-chip debugging functions compared to previous RL78 family microcontrollers. The enhanced on-chip debugging functions are the following three functions, the same as those of the RL78/F23, F24. This chapter provides an overview of these functions.

- Hot plug-in
- Real-time RAM monitor (RRM) and dynamic memory modification (DMM) by the DTC
- On-chip trace

For points requiring cautions in using these functions, refer to the following manuals.

- **E2 Emulator RTE0T00020KCE00000R User's Manual (R20UT3538)**
- **E2 Emulator Lite RTE0T0002LKCE00000R User's Manual (R20UT3240)**

34.1.1 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.

34.1.2 Real-time RAM Monitor (RRM) and Dynamic Memory Modification (DMM) by DTC

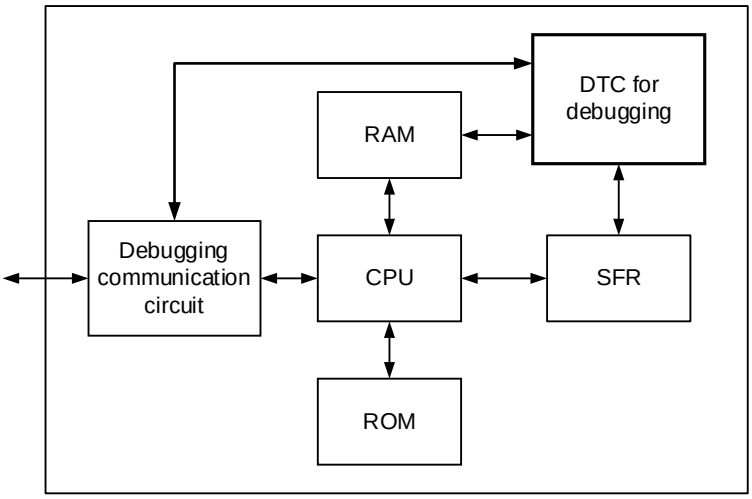
These functions are for accessing the MCU memory during the execution of the user program after the connection between the MCU and emulator.

The CPU handles all access to memory in the conventional RL78 family microcontrollers. The RL78/F22 and RL78/F25 allow access to memory without using the CPU because they have a DTC for debugging.

These functions use RAM.

Figure 34-1 shows the configuration of RRM and DMM by the DTC.

Figure 34-1. Configuration of RRM and DMM by DTC



34.1.3 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.

Table 34-1 shows the RAM area used and the number of branches retained by on-chip trace.

Table 34-1. RAM Area Used and Number of Branches Retained by On-chip Trace

Series Name	RAM	RAM Area Used	Number of Branches
RL78/F22 Note 1	12 KB	• FD500H to FD52FH (Hot plug-in/RRM and DDM by DTC) • FD300H to FD4FFH (on-chip trace)	128
RL78/F25 Note 2	40 KB	• F6500H to F652FH (Hot plug-in/RRM and DDM by DTC) • F6300H to F64FFH (on-chip trace)	

Notes 1. When using the hot plug-in function, RRM, DMM function, or on-chip debugger trace function, be sure to set CFH in the RAMSAR register.

2. When using the hot plug-in function, RRM, DMM function, or on-chip debugger trace function, be sure to set 5FH in the RAMSAR register.

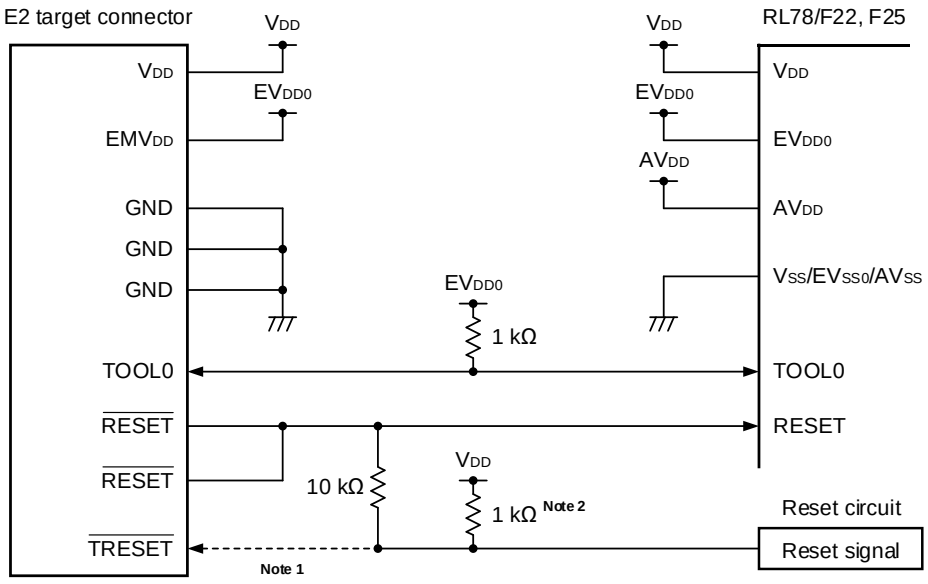
34.2 Connecting E2/E2 Lite On-chip Debugging Emulator to RL78/F22 or RL78/F25

The RL78/F22 and RL78/F25 use the V_{DD}, EV_{DD0}, RESET, TOOL0, and V_{SS} pins to communicate with the host machine via an E2/E2 Lite on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

The RL78/F22 and RL78/F25 are provided with the hot plug-in detection function.

Caution The RL78/F22 and RL78/F25 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 34-2. Connection Example of E2 On-chip Debugging Emulator and RL78/F22 or RL78/F25



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)

34.3 On-chip Debug Security ID

The RL78/F22 and RL78/F25 have an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 32 OPTION BYTE**) and an on-chip debug security ID setting area at 000C6H to 000D5H, to prevent outsiders from reading memory content.

When the boot swap function is used, also set a value that is the same as that of 040C3H and 040C6H to 040D5H in advance, because 000C3H and 000C6H to 000D5H are switched to 040C3H and 040C6H to 040D5H.

Table 34-2. On-chip Debug Security ID

Address	On-chip Debug Security ID
000C6H to 000D5H	Any ID code of 16 bytes (except for all FFH)
040C6H to 040D5H	

34.4 Securing of User Resources

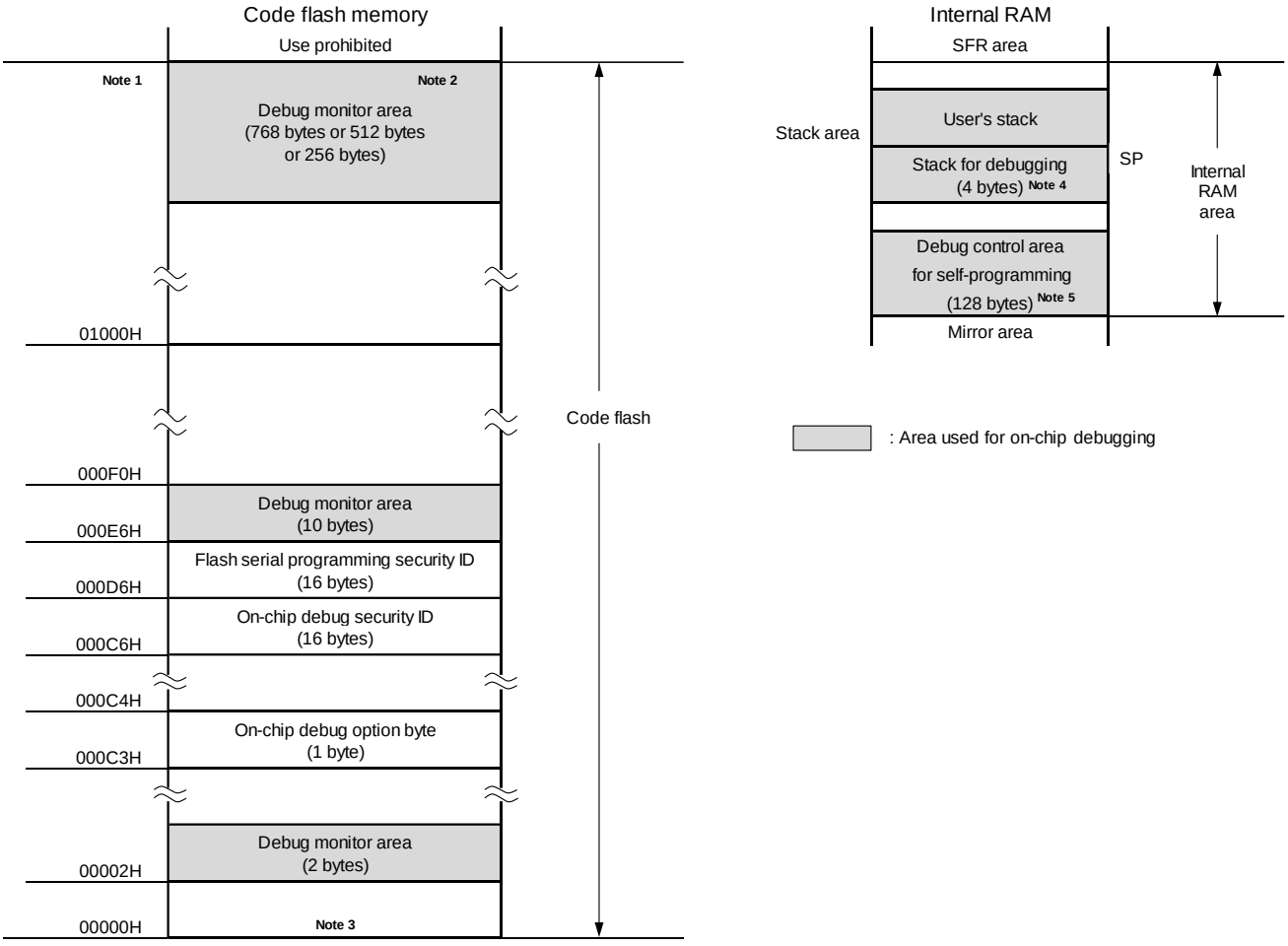
To perform communication between the RL78/F22 or RL78/F25 and E2/E2 Lite on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using linker options.

34.4.1 Securement of Memory Space

The shaded portions in **Figure 34-3** 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 34-3. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on product series as follows.

Product series (code flash memory capacity)	Address of **Note 1**
RL78/F22 (128 KB)	1FFFFH
RL78/F25 (512 KB)	7FFFFH

2. Real-time RAM monitor (RRM) function, Dynamic Memory Modification (DMM) function, Start/Stop function (E2/E2 Lite only supported), transit point (E2 extension function, E2 only supported), area varies depending on usage status.

256 bytes: RRM/DMM function, Start/Stop function, and passing point are all unused
512 bytes: Use any one of RRM/DMM function, Start/Stop function, and passing point
768 bytes: Use both RRM/DMM function and Start/Stop function
(The RRM/DMM function or Start/Stop function cannot be used together with the passing point.)

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.

5. When performing self-programming during on-chip debugging, the on-chip debugger uses the following RAM areas.

RL78/F22: FCF00H to FCF7FH (128 bytes)
RL78/F25: F5F00H to F5F7FH (128 bytes)

CHAPTER 35 BCD CORRECTION CIRCUIT

35.1 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).

35.2 Registers Used by BCD Correction Circuit

The following register is used to control the BCD correction circuit function.

Table 35-1. BCD Correction Circuit Register

Address	Register Name	Symbol	After Reset	Access Size
F00FEH	BCD correction result register	BCDADJ	Undefined	8

(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 35-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

35.3 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 36 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**.

36.1 Conventions Used in Operation List

36.1.1 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 36-1. Operand Identifiers and Specification Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol (SFR symbol) FFF00H to FFFFFH
sfrp	Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only Note) FFF00H to FFFFFH
saddr	FFE20H to FFF1FH Immediate data or labels
saddrp	FFE20H to FFF1FH Immediate data or labels (even addresses only Note)
addr20	00000H to FFFFFH Immediate data or labels
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions Note)
addr5	0080H to 00BFH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	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. The extended special function registers can be described to operand !addr16 as symbols. For details on the SFR and 2nd SFR symbols, refer to each function chapter.

36.1.2 Description of Operation Column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 36-2. 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 X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 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)

36.1.3 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 36-3. 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

36.1.4 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 36-4. 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.

36.1.5 Multiply & Accumulate Instruction

This product supports multiply-accumulate instructions. It can be executed by the instruction of MACHU (unsigned multiply-accumulate) or MACH (signed multiply-accumulate). **Table 36-5** shows the registers used in multiply-accumulate instructions.

Table 36-5. Registers Used in Multiply & Accumulate Instructions

Address	Register Name	Symbol	After Reset	Access Size
FFFF0H FFFF1H	Multiply and accumulation register (L)	MACRL	0000H	16
FFFF2H FFFF3H	Multiply and accumulation register (H)	MACRH	0000H	16

- **Multiply and Accumulation Register (MACRH, MACRL)**

The MACRH register and the MACRL register are used in combination as a 32-bit length register (MACR). The MACR register is used in MACHU or MACH instructions.

Address: FFFF2H (MACRH) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACRH																

Address: FFFF0H (MACRL) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACRL																

MACR (MACRH, MACRL)	Multiply and accumulation register
00000000H to FFFFFFFFH	• Unsigned multiply-accumulate operation Operate using MACHU instruction. MACR ← MACR + AX × BC (unsigned) • Signed multiply-accumulate operation Operate using MACH instruction. MACR ← MACR + AX × BC (signed)

36.2 Operation List

Table 36-6. Operation List (1/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (2/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (3/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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) 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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (4/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (5/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (6/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (7/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (8/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (9/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (10/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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	×	0	0
		X	1	1	–	X − 00H	×	0	0
		B	1	1	–	B − 00H	×	0	0
		C	1	1	–	C − 00H	×	0	0
		!addr16	3	1	4	(addr16) − 00H	×	0	0
		ES:!addr16	4	2	5	(ES:addr16) − 00H	×	0	0
		saddr	2	1	–	(saddr) − 00H	×	0	0
	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (11/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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)	×	×	×

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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. 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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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.
For C source, depending on the C compiler, the NOP instruction is automatically added immediately after the DIVHU or DIVWU instruction is output during the build process. Check the specifications of the C compiler to be used.

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 36-6. Operation List (13/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (14/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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	–	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			×

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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (15/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (16/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (17/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 36-6. Operation List (18/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	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 code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

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 37 ELECTRICAL SPECIFICATIONS (RL78/F25)

Cautions 1. RL78/F25 has an on-chip debug function 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".

37.1 Absolute Maximum Ratings

(1/3)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}, AV_{DD}	$V_{DD} = AV_{DD}$	−0.5 to +6.5	V
	EV_{DD0}, EV_{DD1}	$EV_{DD0} = EV_{DD1} = V_{DD}$	−0.5 to +6.5	V
	V_{SS}, AV_{SS}	$V_{SS} = AV_{SS}$	−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 + 2.1 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, 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}	P80 to P87, P90 to P97, P100 to P105	−0.3 to AV_{DD} + 0.3 Note 2	V
	V_{I3}	P121 to P124, P137, 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, 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}	P80 to P87, P90 to P97, P100 to P105	−0.3 to AV_{DD} + 0.3	V
Analog input voltage	V_{AI1}	ANI22 to ANI28	−0.3 to EV_{DD0} + 0.3 and −0.3 to AV_{DD} + 0.3 Notes 2, 3	V
	V_{AI2}	ANI0 to ANI21	−0.3 to AV_{DD} + 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_{DD} + 0.3.

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.

(2/3)

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, P106, P107, P120, P125 to P127, P130, P140, P150 to P157	−40	mA
		Total of all pins −170 mA	P01, P02, P40 to P47, 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	P80 to P87, P90 to P97, 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, P106, P107, P120, P125 to P127, P130, P140, P150 to P157	40	mA
		Total of all pins 170 mA	P01, P02, P40 to P47, 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	P80 to P87, P90 to P97, P100 to P105	1	mA
		Total of all pins		5	mA

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.

(3/3)

Parameter	Symbol	Conditions		Ratings	Unit
Positive injected current ($V_I > V_{DD}$) Note	I_{INJP}	Per pin	P00 to P03, P10, P11, P13 to P17, P30 to P32, P41 to P47, P50 to P57, P60 to P67, P75 to P77, P106, P107, P126, P127, P140, P150 to P157	5	mA
			P70 to P74, P80 to P87, P90 to P97, P100 to P105, P120, P125	2	mA
Negative injected current ($V_I < V_{SS}$) Note	I_{INJN}	Per pin	P00 to P03, P10, P11, P13 to P17, P30 to P32, P41 to P47, P50 to P57, P60 to P67, P75 to P77, P106, P107, P126, P127, P140, P150 to P157	−5	mA
			P70 to P74, P80 to P87, P90 to P97, P100 to P105, P120, P125	−0.5	mA
Sum of all positive injected currents Note	ΣI_{INJP}	Total of all pins	P00 to P03, P10, P11, P13 to P17, P30 to P32, P41 to P47, P50 to P57, P60 to P67, P75 to P77, P106, P107, P126, P127, P140, P150 to P157	40	mA
			P70 to P74, P80 to P87, P90 to P97, P100 to P105, P120, P125	10	mA
Sum of all negative injected currents Note	ΣI_{INJN}	Total of all pins	P00 to P03, P10, P11, P13 to P17, P30 to P32, P41 to P47, P50 to P57, P60 to P67, P75 to P77, P106, P107, P126, P127, P140, P150 to P157	−40	mA
			P70 to P74, P80 to P87, P90 to P97, P100 to P105, P120, P125	−2	mA
Total of all injected currents Note	$\Sigma\lvert I_{INJP}\rvert + \Sigma\lvert I_{INJN}\rvert$	Total of all pins	P00 to P03, P10, P11, P13 to P17, P30 to P32, P41 to P47, P50 to P57, P60 to P67, P75 to P77, P106, P107, P126, P127, P140, P150 to P157	40	mA
			P70 to P74, P80 to P87, P90 to P97, P100 to P105, P120, P125	10	mA
Operating ambient temperature	T_A	In normal operation mode	Grade 3 products	−40 to +105	°C
			Grade 4 products	−40 to +125	
		In flash memory programming mode	Grade 3 products	−40 to +105	°C
			Grade 4 products	−40 to +125	
Operating temperature	T_{OPR}		Grade 3 products	−40 to +105	°C
			Grade 4 products	−40 to +125	
Storage temperature	T_{stg}			−65 to +150	°C

Note Conditions: 2.7 V ≤ EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1} = 0 V

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.

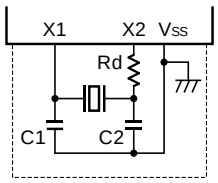
Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

2. V_I: Input voltage level to the port pins.

37.2 Oscillator Characteristics

37.2.1 Main System Clock Oscillator Characteristics

($T_A = T_{OPR}$, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 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	2.0		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.

37.2.2 On-chip Oscillator Characteristics

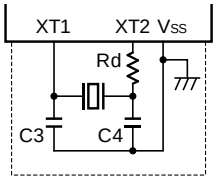
($T_A = T_{OPR}$, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Note	f_{IH}		2		80	MHz
High-speed on-chip oscillator clock frequency accuracy	–		−1.0		+1.0	%
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/040C2H) and bits 0 to 2 of the HOCODIV register.

37.2.3 Subsystem Clock Oscillator Characteristics

($T_A = T_{OPR}$, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 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.

37.2.4 PLL Circuit Characteristics

($T_A = T_{OPR}$, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Resonator	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
PLL input enable clock frequency Note 1	f_{PLLI}	f_{MAIN}: 4.0 MHz	FMAINDIV[1:0] = 00B	3.92	4.0	4.08	MHz
		f_{MAIN}: 8.0 MHz	FMAINDIV[1:0] = 00B	7.84	8.0	8.16	MHz
		f_{MAIN}: 16.0 MHz	FMAINDIV[1:0] = 10B	7.84	8.0	8.16	MHz
		f_{MAIN}: 20.0 MHz	FMAINDIV[1:0] = 11B	4.90	5.0	5.10	MHz
PLL output frequency (center value)	f_{PLL}	f_{MAIN}: 20 MHz, PLLMULA = 0, PLLMUL = 1	PLLDIV0 = 0, FPLLDIV = 0, PLLDIV1 = 0	f_{PLLI} × 16/2			MHz
			PLLDIV0 = 0, FPLLDIV = 1, PLLDIV1 = 1	f_{PLLI} × 16			MHz
		f_{MAIN}: 4 MHz, PLLMULA = 1, PLLMUL = 1	PLLDIV0 = 0, FPLLDIV = 0, PLLDIV1 = 0	f_{PLLI} × 20/2			MHz
			PLLDIV0 = 0, FPLLDIV = 1, PLLDIV1 = 1	f_{PLLI} × 20			MHz
		f_{MAIN}: 8 MHz or 16 MHz, PLLMULA = 0, PLLMUL = 0	PLLDIV0 = 1, FPLLDIV = 0, PLLDIV1 = 0	f_{PLLI} × 12/4			MHz
			PLLDIV0 = 0, FPLLDIV = 0, PLLDIV1 = 1	f_{PLLI} × 12/2			MHz
		f_{MAIN}: 8 MHz or 16 MHz, PLLMULA = 0, PLLMUL = 1	PLLDIV0 = 1, FPLLDIV = 0, PLLDIV1 = 0	f_{PLLI} × 16/4			MHz
			PLLDIV0 = 0, FPLLDIV = 0, PLLDIV1 = 1	f_{PLLI} × 16/2			MHz
		f_{MAIN}: 8 MHz or 16 MHz, PLLMULA = 1, PLLMUL = 0	PLLDIV0 = 0, FPLLDIV = 0, PLLDIV1 = 0	f_{PLLI} × 10/2			MHz
			PLLDIV0 = 0, FPLLDIV = 1, PLLDIV1 = 1	f_{PLLI} × 10			MHz
Long-term jitter Note 2	t_{LJ}	term = 1 μs		−1		+1	ns
		term = 10 μs		−1		+1	ns
		term = 20 μs		−2		+2	ns

Notes 1. X1 clock or external main system clock can be selected as the PLL input clock.

2. Guaranteed by design, but not tested before shipment.

Remark f_{MAIN}: Main system clock frequency.

37.3 DC Characteristics

37.3.1 Pin Characteristics

For the relationship between the port pins shown in the following tables and the products, refer to **CHAPTER 4 PORT FUNCTIONS**.

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V) **(1/6)**

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, P106, P107, P120, P125 to P127, P130, P140, P150 to P157	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, 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)	Grade 3 products: 4.0 V ≤ EV_{DD0} ≤ 5.5 V			−50.0	mA
			Grade 3 products: 2.7 V ≤ EV_{DD0} < 4.0 V			−29.0	mA
			Grade 4 products: 4.0 V ≤ EV_{DD0} ≤ 5.5 V			−42.0	mA
			Grade 4 products: 2.7 V ≤ EV_{DD0} < 4.0 V			−29.0	mA
	I_{OH2}	Per pin for: P80 to P87, P90 to P97, P100 to P105	2.7 V ≤ AV_{DD} ≤ 5.5 V			−0.1	mA
		Total of all pins (for duty factors ≤ 70% Note 2)	2.7 V ≤ AV_{DD} ≤ 5.5 V			−2.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from EV_{DD0}, EV_{DD1}, AV_{DD}, and V_{DD} pins to an output pin.

2. These output current values are obtained under the condition that the duty factor is 70% or less.
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 $(I_{OH} \times 0.7) / (n \times 0.01)$

<Example> Where I_{OH} = −10.0 mA and n = 80%
Total output current of pins = $(-10.0 \times 0.7) / (80 \times 0.01) \approx -8.7$ 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.

Caution P10 to P17, P32, 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.

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V) **(2/6)**

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, P106, P107, P120, P125 to P127, P130, P140, P150 to P157	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, 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: P80 to P87, P90 to P97, P100 to P105	2.7 V ≤ AV_{DD} ≤ 5.5 V			0.4	mA
		Total of all pins (for duty factors ≤ 70% Note 2)	2.7 V ≤ AV_{DD} ≤ 5.5 V			5.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the EV_{SS0}, EV_{SS1}, AV_{SS}, and V_{SS} pins.

2. These output current values are obtained under the condition that the duty factor is 70% or less.
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 $(I_{OL} \times 0.7) / (n \times 0.01)$
  <Example> Where I_{OL} = 10.0 mA and n = 80%
  Total output current of pins = $(10.0 \times 0.7) / (80 \times 0.01) \approx 8.7$ 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.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V) (3/6)

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	V
			2.7 V ≤ EV_{DD0} < 4.0 V	0.7 EV_{DD0}		EV_{DD0} Note	V
	V_{IH2}	P10, P11, P13, P14, P16, P17, P30, P41, P43, P50, P52 to P54, P60 to P63, P70, P71, P73, P75 to P77, P107, P120, P125, P150, P152, P153, P154 (Schmitt 3 mode)	4.0 V ≤ EV_{DD0} ≤ 5.5 V	0.8 EV_{DD0}		EV_{DD0} Note	V
			2.7 V ≤ EV_{DD0} < 4.0 V	0.85 EV_{DD0}		EV_{DD0} Note	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	V
			2.7 V ≤ EV_{DD0} < 4.0 V	2.0		EV_{DD0} Note	V
	V_{IH4}	P80 to P87, P90 to P97, P100 to P105 (fixed to Schmitt 3 mode)	4.0 V ≤ AV_{DD} ≤ 5.5 V	0.8 AV_{DD}		AV_{DD}	V
			2.7 V ≤ AV_{DD} < 4.0 V	0.85 AV_{DD}		AV_{DD}	V
	V_{IH5}	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
	V_{IH7}	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

Note The maximum value of V_{IH} of the pins P10 to P17, P32, P60 to P63, P70 to P72, and P120 is EV_{DD0}, even in N-ch open-drain mode and set to high-level output.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V) **(4/6)**

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, P41, P43, P50, P52 to P54, P60 to P63, P70, P71, P73, P75 to P77, P107, P120, P125, P150, P152, P153, P154 (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}	P80 to P87, P90 to P97, P100 to P105 (fixed to Schmitt 3 mode)	4.0 V ≤ AV_{DD} ≤ 5.5 V	0		0.5 AV_{DD}	V
			2.7 V ≤ AV_{DD} < 4.0 V	0		0.4 AV_{DD}	V
	V_{IL5}	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
	V_{IL7}	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

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V) **(5/6)**

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, 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}	P80 to P87, P90 to P97, P100 to P105	2.7 V ≤ AV_{DD} ≤ 5.5 V I_{OH2} = −100 µA	AV_{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, 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}	P80 to P87, P90 to P97, P100 to P105	2.7 V ≤ AV_{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

Caution P10 to P17, P32, 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.

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V) **(6/6)**

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, P106, P107, P120, P125 to P127, P140, P150 to P157	$V_I = EV_{DD0}$				1	μA
	I_{LIH2}	P80 to P87, P90 to P97, P100 to P105	$V_I = AV_{DD}$				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
	I_{LIH4}	P137, RESET	$V_I = V_{DD}$				1	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, P106, P107, P120, P125 to P127, P140, P150 to P157	$V_I = EV_{SS0}$				−1	μA
	I_{LIL2}	P80 to P87, P90 to P97, P100 to P105	$V_I = AV_{SS}$				−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
	I_{LIL4}	P137, RESET	$V_I = V_{SS}$				−1	μA
Positive injected current Note	$I_{INJPRMS}$	P00 to P03, P10, P11, P13 to P17, P30 to P32, P41 to P47, P50 to P57, P60 to P67, P75 to P77, P106, P107, P126, P127, P140, P150 to P157	Per pin, $V_I > EV_{DD0}$				0.4	mA
			Total of all pins, $V_I > EV_{DD0}$				4	mA
		P70 to P74, P80 to P87, P90 to P97, P100 to P105, P120, P125	Per pin, $V_I > V_{DD} = AV_{DD}$				0.15	mA
			Total of all pins, $V_I > V_{DD} = AV_{DD}$				1	mA
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, P106, P107, P120, P125 to P127, P140, P150 to P157	$V_I = EV_{SS0}$, in input port		10	20	100	kΩ

Note Guaranteed by design, but not tested before shipment.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

2. V_I: Input voltage level to the port pins.

37.3.2 Supply Current Characteristics

(1) RL78/F25 (Grade 3 Products)

(T_A = −40 to +105°C, 2.7 V ≤ EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (1/2)

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} = 80 MHz	f_{CLK} = 40 MHz Notes 3, 4		6.8	15.0	mA
					f_{IH} = 40 MHz	f_{CLK} = f_{IH} Notes 3, 4		6.5	13.5	mA
					f_{IH} = 2 MHz	f_{CLK} = f_{IH} Notes 3, 4		1.0	2.3	mA
				Resonator operation	f_{MX} = 20 MHz	f_{CLK} = f_{MX} Notes 3, 5		3.7	7.5	mA
					f_{MX} = 2 MHz	f_{CLK} = f_{MX} Notes 3, 5		0.9	2.1	mA
				Resonator operation (PLL operation) (PLL input clock = f_{MX})	f_{PLL} = 80 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Notes 3, 6		6.8	15.0	mA
					f_{PLL} = 40 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Notes 3, 6		6.6	13.5	mA
					f_{PLL} = 40 MHz, f_{MX} = 4 MHz	f_{CLK} = 40 MHz Notes 3, 6		6.4	13.0	mA
				Subsystem clock operation	f_{SUB} = 32.768 kHz	f_{CLK} = f_{SUB} Note 7		12.3	550	µA
				Low-speed on-chip oscillator clock operation	f_{IL} = 15 kHz	f_{CLK} = f_{IL} Note 8		9.8	550	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, and with setting of ADCEN = 0 or ADSLP = 1.

8. When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped, and with setting of ADCEN = 0 or ADSLP = 1.

Remark f_{MX}: High-speed system clock frequency
f_{SUB}: Subsystem clock frequency
f_{PLL}: PLL clock frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency

(T_A = −40 to +105°C, 2.7 V ≤ EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (2/2)

Items	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Notes 1, 3	I_{DD2}	HALT mode Note 2	High-speed on-chip oscillator clock operation	f_{IH} = 80 MHz	f_{CLK} = 40 MHz Note 5		2.0	10.0	mA
				f_{IH} = 40 MHz	f_{CLK} = f_{IH} Note 5		1.7	8.5	mA
				f_{IH} = 2 MHz	f_{CLK} = f_{IH} Note 5		0.5	1.8	mA
			Resonator operation	f_{MX} = 20 MHz	f_{CLK} = f_{MX} Note 6		1.0	5.0	mA
				f_{MX} = 2 MHz	f_{CLK} = f_{MX} Note 6		0.3	1.8	mA
			Resonator operation (PLL operation) (PLL input clock = f_{MX})	f_{PLL} = 80 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Note 7		2.0	10.0	mA
				f_{PLL} = 40 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Note 7		1.9	8.5	mA
				f_{PLL} = 40 MHz, f_{MX} = 4 MHz	f_{CLK} = 40 MHz Note 7		1.7	8.0	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz	f_{CLK} = f_{SUB} Note 8		7.1	160	µA
			Low-speed on-chip oscillator clock operation	f_{IL} = 15 kHz	f_{CLK} = f_{IL} Note 9		7.1	160	µA
	I_{DD3}	STOP mode Note 4	T_A = +25°C				0.6		µA
			T_A = +50°C					10.5	
			T_A = +70°C					27	
			T_A = +105°C					125	
	I_{SNOZ}	SNOOZE mode	DTC operation				6.0		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. 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. When high-speed system clock, subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped. When the RAMSDMD bit and RAMSDS bit of the PSMCR register are set to 11B (shutdown mode).

5. When high-speed system clock, subsystem clock, PLL clock, and low-speed on-chip oscillator clock are stopped.

6. When subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

7. When subsystem clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

8. When high-speed system clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped, and with setting of ADCEN = 0 or ADSLP = 1.

9. When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped, and with setting of ADCEN = 0 or ADSLP = 1.

Remark f_{MX}: High-speed system clock frequency
f_{SUB}: Subsystem clock frequency
f_{PLL}: PLL clock frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency

(T_A = −40 to +105°C, 2.7 V ≤ EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{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_{WDT} = 15 kHz			0.3		µA
A/D converter operating current	I_{ADC} Note 3	When conversion at maximum speed	AV_{DD} = V_{DD} = 5.0 V		1.3	1.6	mA
		When internal reference voltage is selected Note 4			128		µA
Sample-and-hold circuit operating current	I_{ADSH} Note 5				0.8	1.2	mA
LVD operating current	I_{LVD} Note 6				0.045		µA
D/A converter operating current	I_{DAC}				0.4	0.5	mA
Comparator operating current	I_{CMP}				100		µA
BGO operating current	I_{BGO} Note 7				2.5	12.2	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 15 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. Operating current that increases when the internal reference voltage is selected. This current flows even when conversion is stopped.

5. Operating current that increases when the sample-and-hold circuit is used. This current flows for each analog input channel.

6. 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.

7. Current increased by the BGO operation. The current value is the sum of I_{DD1} and I_{BGO} when the BGO operates in operation mode.

(2) RL78/F25 (Grade 4 Products)

(T_A = −40 to +125°C, 2.7 V ≤ EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1} = 0 V) **(1/2)**

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} = 80 MHz	f_{CLK} = 40 MHz Notes 3, 4		6.8	15.0	mA
					f_{IH} = 40 MHz	f_{CLK} = f_{IH} Notes 3, 4		6.5	13.5	mA
					f_{IH} = 2 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		3.7	7.5	mA
					f_{MX} = 2 MHz	f_{CLK} = f_{MX} Notes 3, 5		0.9	2.4	mA
				Resonator operation (PLL operation) (PLL input clock = f_{MX})	f_{PLL} = 80 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Notes 3, 6		6.8	15.0	mA
					f_{PLL} = 40 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Notes 3, 6		6.6	13.5	mA
					f_{PLL} = 40 MHz, f_{MX} = 4 MHz	f_{CLK} = 40 MHz Notes 3, 6		6.4	13.0	mA
				Subsystem clock operation	f_{SUB} = 32.768 kHz	f_{CLK} = f_{SUB} Note 7		12.3	800	µA
				Low-speed on-chip oscillator clock operation	f_{IL} = 15 kHz	f_{CLK} = f_{IL} Note 8		9.8	800	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 and A/D converter 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, and with setting of ADCEN = 0 or ADSLP = 1.

8. When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped, and with setting of ADCEN = 0 or ADSLP = 1.

Remark f_{MX}: High-speed system clock frequency
f_{SUB}: Subsystem clock frequency
f_{PLL}: PLL clock frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency

(T_A = −40 to +125°C, 2.7 V ≤ EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (2/2)

Items	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Notes 1, 3	I_{DD2}	HALT mode Note 2	High-speed on-chip oscillator clock operation	f_{IH} = 80 MHz	f_{CLK} = 40 MHz Note 5		2.0	10.0	mA
				f_{IH} = 40 MHz	f_{CLK} = f_{IH} Note 5		1.7	8.5	mA
				f_{IH} = 2 MHz	f_{CLK} = f_{IH} Note 5		0.5	2.0	mA
			Resonator operation	f_{MX} = 20 MHz	f_{CLK} = f_{MX} Note 6		1.0	5.0	mA
				f_{MX} = 2 MHz	f_{CLK} = f_{MX} Note 6		0.3	2.0	mA
			Resonator operation (PLL operation) (PLL input clock = f_{MX})	f_{PLL} = 80 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Note 7		2.0	10.0	mA
				f_{PLL} = 40 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Note 7		1.9	8.5	mA
				f_{PLL} = 40 MHz, f_{MX} = 4 MHz	f_{CLK} = 40 MHz Note 7		1.7	8.0	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz	f_{CLK} = f_{SUB} Note 8		7.1	330	µA
			Low-speed on-chip oscillator clock operation	f_{IL} = 15 kHz	f_{CLK} = f_{IL} Note 9		7.1	330	µA
	I_{DD3}	STOP mode Note 4	T_A = +25°C				0.6		µA
			T_A = +50°C					10.5	
			T_A = +70°C					27	
			T_A = +105°C					125	
			T_A = +125°C					290	
	I_{SNOZ}	SNOOZE mode	DTC operation				6.0		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. 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, and A/D converter are stopped.

4. When high-speed system clock, subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped. When the RAMSDMD bit and RAMSDS bit of the PSMCR register are set to 11B (shutdown mode).

5. When high-speed system clock, subsystem clock, PLL clock, and low-speed on-chip oscillator clock are stopped.

6. When subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

7. When subsystem clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

8. When high-speed system clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped, and with setting of ADCEN = 0 or ADSLP = 1.

9. When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped, and with setting of ADCEN = 0 or ADSLP = 1.

Remark f_{MX}: High-speed system clock frequency
f_{SUB}: Subsystem clock frequency
f_{PLL}: PLL clock frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency

(T_A = −40 to +125°C, 2.7 V ≤ EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{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_{WDT} = 15 kHz			0.3		μA
A/D converter operating current	I_{ADC} Note 3	When conversion at maximum speed	AV_{DD} = V_{DD} = 5.0 V		1.3	1.6	mA
		When internal reference voltage is selected Note 4			128		μA
Sample-and-hold circuit operating current	I_{ADSH} Note 5				0.8	1.2	mA
LVD operating current	I_{LVD} Note 6				0.045		μA
D/A converter operating current	I_{DAC}				0.4	0.5	mA
Comparator operating current	I_{CMP}				100		μA
BGO operating current	I_{BGO} Note 7				2.5	12.2	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 15 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. Operating current that increases when the internal reference voltage is selected. This current flows even when conversion is stopped.

5. Operating current that increases when the sample-and-hold circuit is used. This current flows for each analog input channel.

6. 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.

7. Current increased by the BGO operation. The current value is the sum of I_{DD1} and I_{BGO} when the BGO operates in operation mode.

37.4 AC Characteristics

37.4.1 Basic Operation

($T_A = T_{OPR}$, 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		0.025		0.5	μs
		High-speed system clock operation		0.05		0.5	μs
		PLL clock operation		0.025		0.5	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		0.025		0.5	μs
CPU/peripheral hardware clock frequency	f_{CLK}			0.025		66.6	μs
External system clock frequency	f_{EX}			2.0		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, TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1, TRJIO0, TRJO0 output frequency	f_{TO}	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, TRDIOC0, TRDIOD0, TRDIOD1, TRJO0 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_{TJIH}, t_{TJIL}	TRJIO0		40			ns
Timer RDe input high-level, low-level width	t_{TDIH}, t_{TDIL}	TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1, TRDCLK0, TRD0RES, TRD1RES		$3/f_{TRD}$			ns
Timer RDe pulse output forced cutoff signal low-level width	t_{TDSIL}	P137/INTP0	2 MHz < f_{CLK} ≤ 40 MHz	1			μs
			f_{CLK} ≤ 2 MHz	$1/f_{CLK}$ + 1			μs

Caution Excluding the error in oscillation frequency accuracy.

Remark f_{MCK}: Timer array unit operation clock frequency
f_{TRD}: Timer RDe operation clock frequency

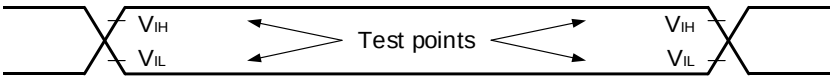
($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = V_{DD}$ ≤ 5.5 V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP13 Note 1		1			μs
KR0 to KR7 key interrupt input low-level width	t_{KR}			250			ns
RESET low-level width	t_{RSL}	Note 1		10			μs
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, P106, P107, P120, P125 to P127, P130, P140, P150 to P157 (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 2	60	ns
			2.7 V ≤ EV_{DD0} < 4.0 V			100	ns

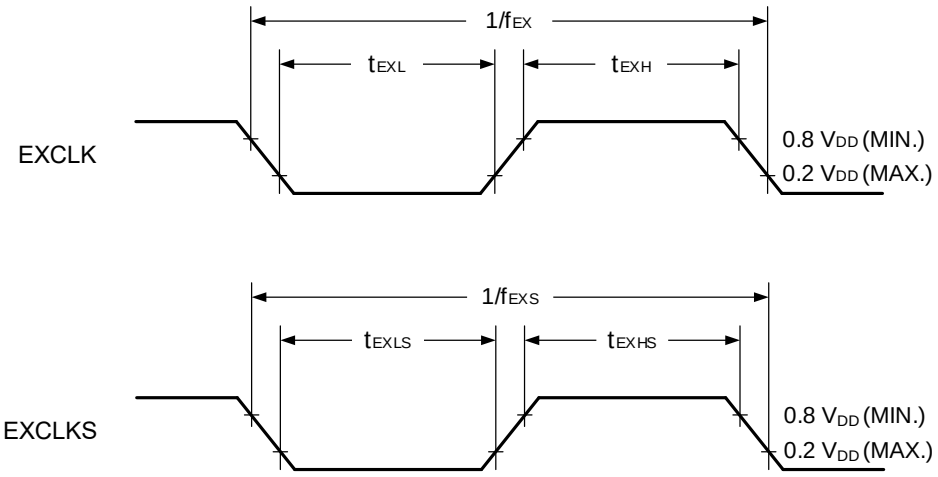
Notes 1. Pins RESET, INTP0 to INTP3, INTP12, and INTP13 have noise filters for transient levels lasting less than 100 ns.

2. T_A = +25°C, E_{VDD0} = 5.0 V

AC Timing Test Points

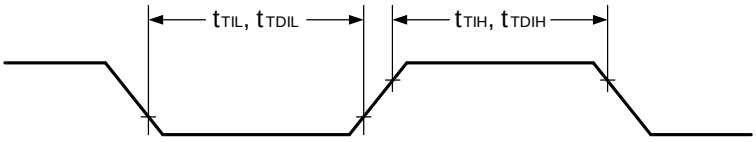


External System Clock Timing

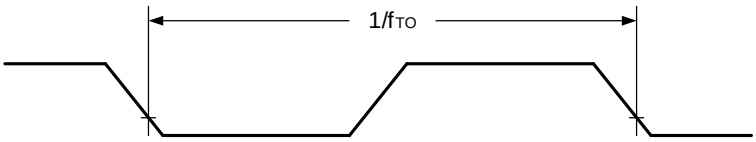


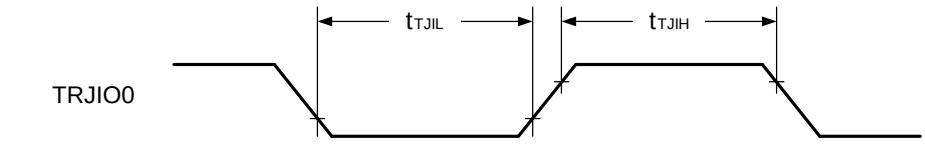
TI/TO Timing

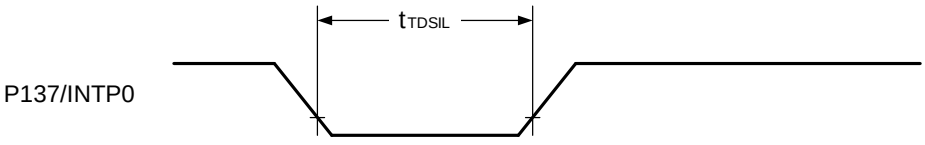
TI00 to TI07, TI10 to TI17, TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1, TRDCLK0, TRD0RES, TRD1RES



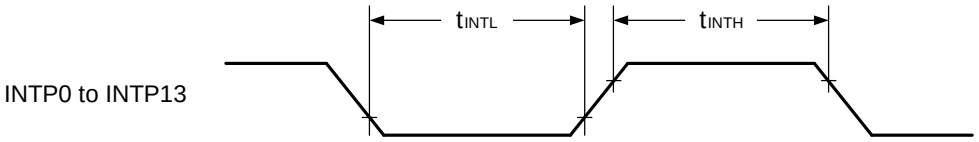
TO00 to TO07, TO10 to TO17, TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1, TRJIO0, TRJO0



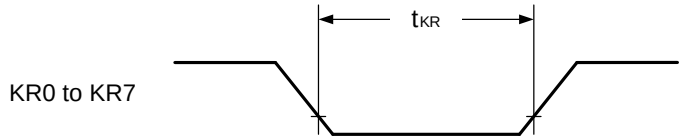




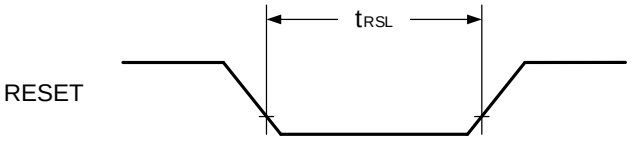
Interrupt Request Input Timing



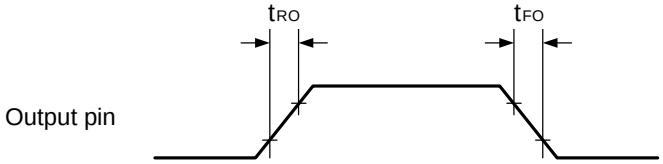
Key Interrupt Input Timing



RESET Input Timing



Output Rising and Falling Timing



37.5 Peripheral Functions Characteristics

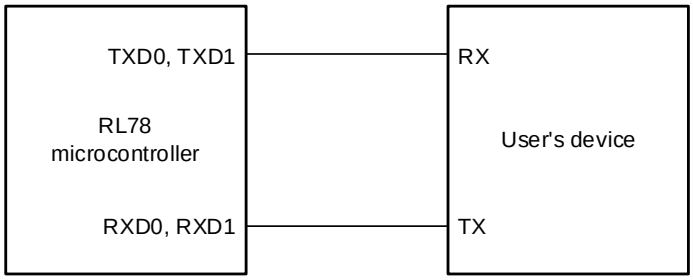
37.5.1 Serial Array Unit

(1) During Communication at Same Potential (UART mode) (Dedicated Baud Rate Generator Output)

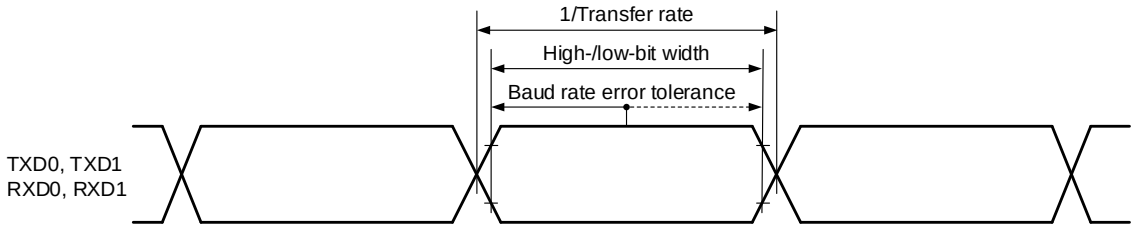
($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = V_{DD}$ ≤ 5.5 V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Transfer rate	–					f_{MCK}/6	bps
		f_{CLK} = 40 MHz, $f_{MCK} = f_{CLK}$	Normal slew rate			6.6	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 normal input buffer for the RXD0 and RXD1 pins, and select normal output mode for the TXD0 and TXD1 pins.

Remark f_{MCK}: Serial array unit operation clock frequency

(2) During Communication at Same Potential (CSI mode) (Master Mode, SCKp … Internal Clock Output, Normal Slew Rate)

($T_A = T_{OPR}$, $2.7\ V \le EV_{DD0} = EV_{DD1} = V_{DD} \le 5.5\ V$, $V_{SS} = EV_{SS0} = EV_{SS1} = 0\ V$)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY1}	Grade 3 products		100 Note 5			ns
		Grade 4 products		150 Note 5			ns
SCKp high-level width, low-level width	t_{KH1}, t_{KL1}	$4.0\ V \le EV_{DD0} \le 5.5\ V$		$t_{KCY1}/2 - 12$			ns
		$2.7\ V \le EV_{DD0} < 4.0\ V$		$t_{KCY1}/2 - 18$			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	Grade 3 products	$4.0\ V \le EV_{DD0} \le 5.5\ V$	33			ns
			$2.7\ V \le EV_{DD0} < 4.0\ V$	44			ns
		Grade 4 products	$4.0\ V \le EV_{DD0} \le 5.5\ V$	44			ns
			$2.7\ V \le EV_{DD0} < 4.0\ V$	55			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}			30			ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 30 pF Note 4				30	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.
The SIp setup time becomes “to 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 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 SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

5. $t_{KCY1} \ge 4/f_{MCK}$ must also be satisfied.

Caution Select normal input buffer for the SIp pin, and select normal output mode for the SOp and SCKp pins.

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, SCKp … Internal Clock Output, Special Slew Rate)

($T_A = T_{OPR}$, 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
SCKp cycle time	t_{KCY1}		500 Note 5			ns
SCKp high-level width, low-level width	t_{KH1}, t_{KL1}		t_{KCY1}/2 − 60			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}		120			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}		80			ns
Delay time from 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 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 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 SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

5. $t_{KCY1} \geq 4/f_{MCK}$ must also be satisfied.

Caution Select normal input buffer for the SIp pin, and select normal output mode and special slew rate for the SOp and SCKp pins.

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, SCKp … External Clock Input, Normal Slew Rate)

($T_A = T_{OPR}$, 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
SCKp cycle time	t_{KCY2}	f_{MCK} > 32 MHz		10/f_{MCK}			ns
		f_{MCK} ≤ 32 MHz		8/f_{MCK}			ns
SCKp high-level width, low-level width	t_{KH2}, t_{KL2}			t_{KCY2}/2			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}			1/f_{MCK} + 20			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}			1/f_{MCK} + 31			ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	4.0 V ≤ EV_{DD0} ≤ 5.5 V			2/f_{MCK} + 44	ns
			2.7 V ≤ EV_{DD0} < 4.0 V			2/f_{MCK} + 57	ns
SSIp setup time	t_{SSIK}	DAP = 0		120			ns
		DAP = 1		1/f_{MCK} + 120			ns
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 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 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 SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select normal input buffer for the SIp, SCKp and SSIp pins, and select 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, SCKp … External Clock Input, Special Slew Rate)

($T_A = T_{OPR}$, 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
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
SCKp high-level width, low-level width	t_{KH2}, t_{KL2}		t_{KCY2}/2			ns
SIp setup time (to SCKp↑) Note1	t_{SIK2}		1/f_{MCK} + 50			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}		1/f_{MCK} + 50			ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4			2/f_{MCK} + 80	ns
SSIp setup time	t_{SSIK}	DAP = 0	120			ns
		DAP = 1	1/f_{MCK} + 120			ns
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 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 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 SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

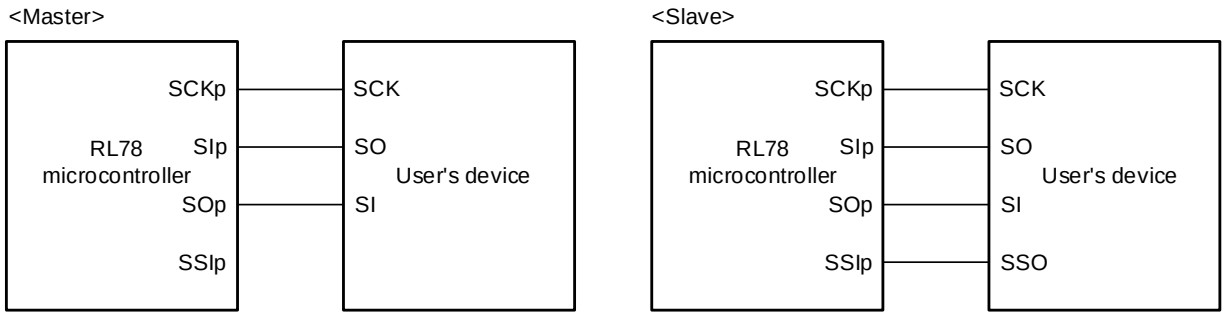
4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select normal input buffer for the SIp, SCKp and SSIp pins, and select normal output mode and special slew rate for the SOp pin.

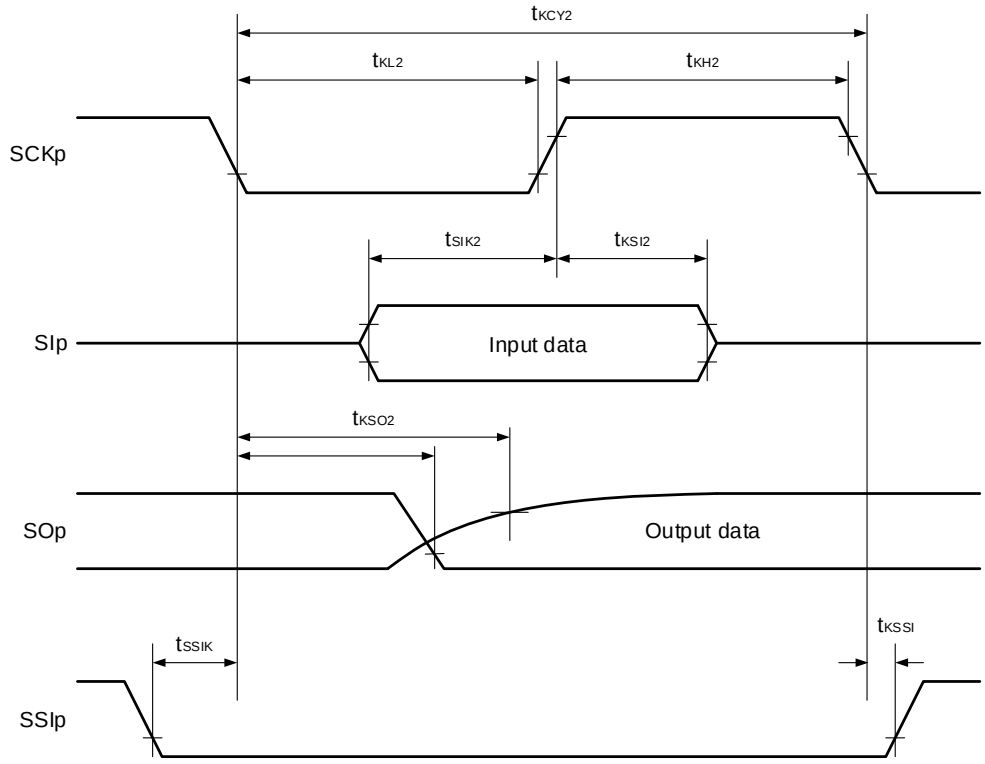
Remarks 1. p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

2. f_{MCK}: Serial array unit operation clock frequency

CSI Mode Connection Diagram (During Communication at Same Potential)

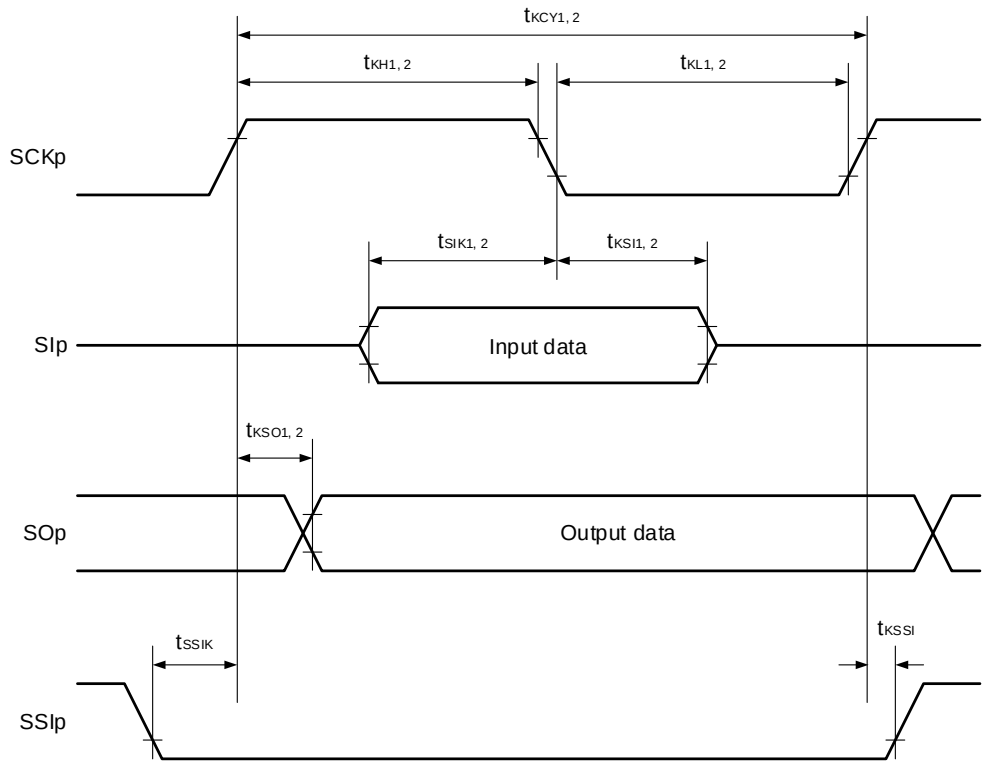


CSI Mode Serial Transfer Timing (During Communication at Same Potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



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)

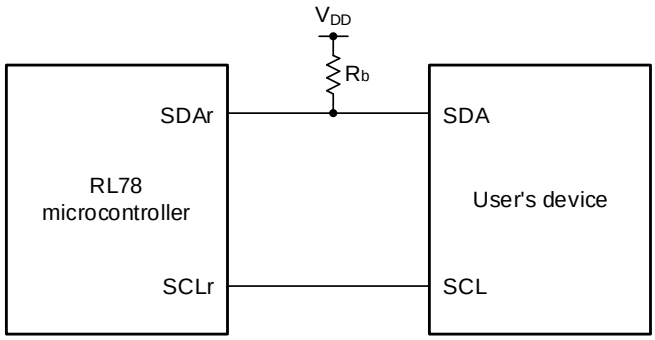
(6) During Communication at Same Potential (Simplified I^2C Mode) (SDAr: N-ch Open-drain Output (EV_{DD} Tolerance) Mode, SCLr: Normal Output Mode)

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = V_{DD}$ ≤ 5.5 V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

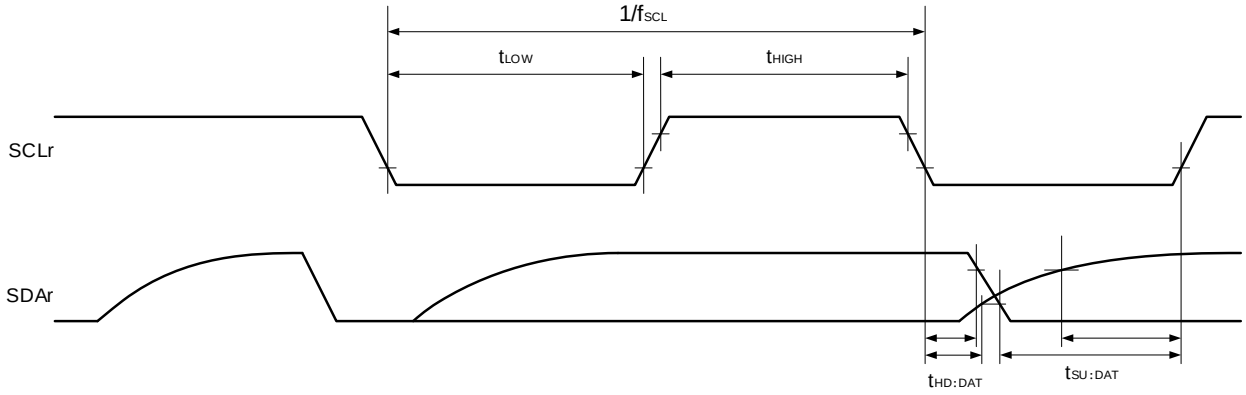
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SCLr clock frequency	f_{SCL}				1000 Note	kHz
Hold time when SCLr = "L"	t_{LOW}		475			ns
Hold time when SCLr = "H"	t_{HIGH}		475			ns
Data setup time (reception)	$t_{SU:DAT}$		1/f_{MCK} + 85			ns
Data hold time (transmission)	$t_{HD:DAT}$	C_b = 50 pF, R_b = 2.7 kΩ	0		305	ns

Note $f_{SCL} \leq f_{MCK}/4$ 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 normal input buffer and N-ch open-drain output mode for the SDAr pin, and select 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

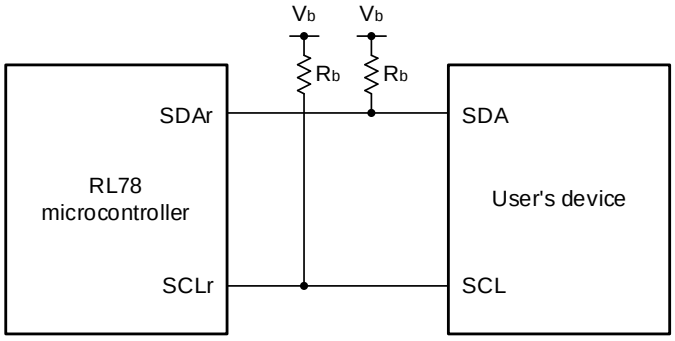
(7) During Communication at Same Potential (Simplified I^2C Mode) (SDAr and SCLr: N-ch Open-drain Output (EV_{DD} Tolerance) Mode)

($T_A = T_{OPR}$, 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 ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	1300		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, C_b = 100 pF, R_b = 2.7 kΩ			
Hold time when SCLr = "H"	t_{HIGH}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	600		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, C_b = 100 pF, R_b = 2.7 kΩ			
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	1/f_{MCK} + 120		ns
		2.7 V ≤ EV_{DD0} < 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 ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 1.7 kΩ	0	300	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, C_b = 100 pF, R_b = 2.7 kΩ			

Note $f_{SCL} \le f_{MCK}/4$ must also be satisfied.

Simplified I^2C Mode Connection Diagram (During Communication at Same Potential)



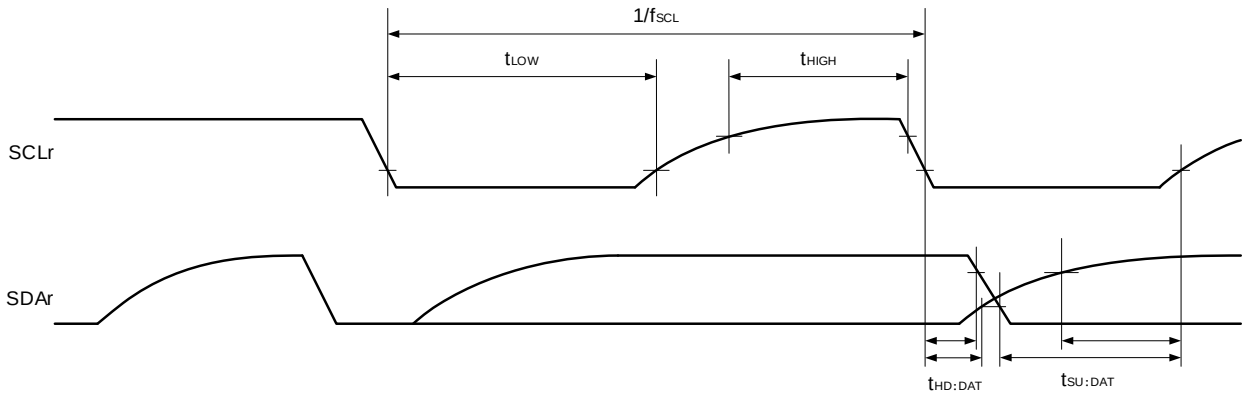
Caution Select normal input buffer and N-ch open-drain output mode for the SDAr and SCLr pins.

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 I²C Mode Serial Transfer Timing (During Communication at Same Potential)



Remark r: IICr (r = 00, 01, 10, 11)

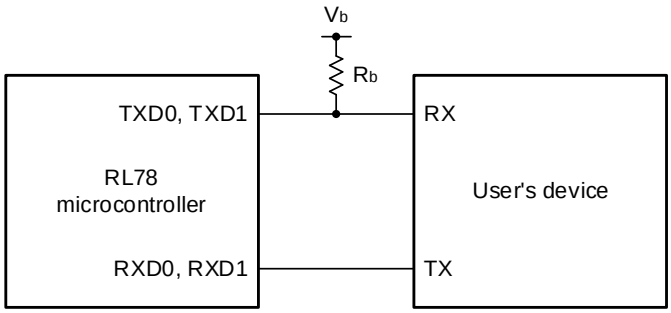
(8) Communication at Different Potential (UART mode) (TXD Output Buffer: N-ch Open-drain, RXD Input Buffer: TTL)

($T_A = T_{OPR}$, 4.0 V ≤ $EV_{DD0} = EV_{DD1} = V_{DD}$ ≤ 5.5 V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

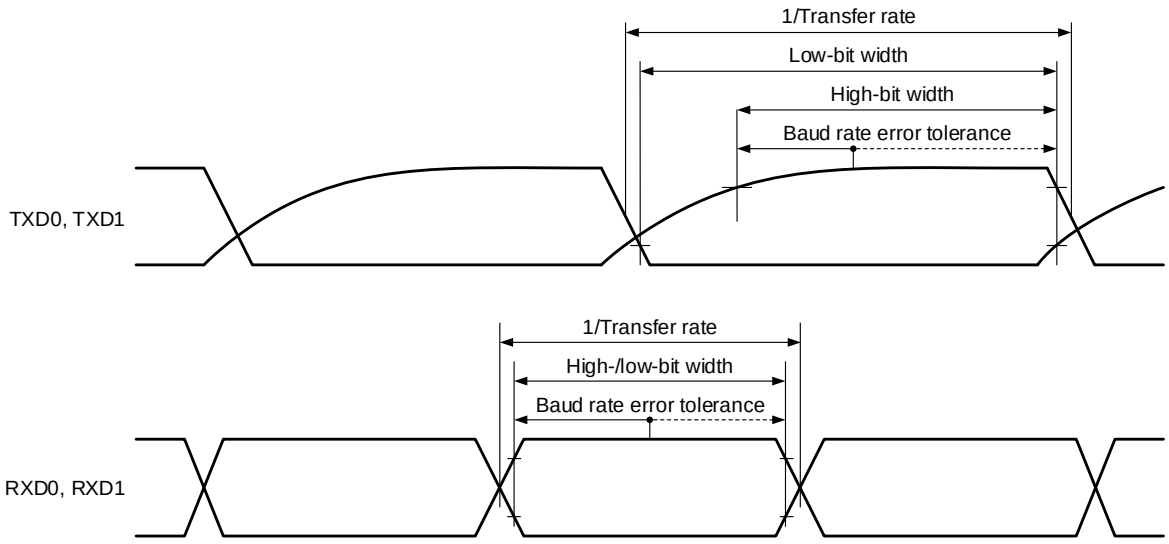
Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Transfer rate	–	Reception	2.7 V ≤ V_b ≤ EV_{DD0}, V_{IH} = 2.2 V, V_{IL} = 0.8 V					f_{MCK}/6	bps
				Theoretical value of the maximum transfer rate Note (C_b = 30 pF)	Grade 3 products			5.3	Mbps
					Grade 4 products			4.0	
		Transmission	2.7 V ≤ V_b ≤ 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	Grade 3 products			5.3	Mbps
					Grade 4 products			4.0	

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 TTL input buffer for the RXD0 and RXD1 pins, and select N-ch open-drain output mode for the TXD0 and TXD1 pins.

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, SCKp … Internal Clock Output, Normal Slew Rate)

($T_A = T_{OPR}$, 4.0 V ≤ $EV_{DD0} = EV_{DD1} = V_{DD}$ ≤ 5.5 V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

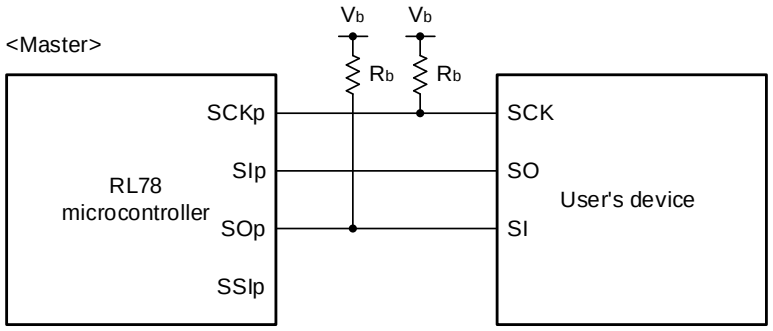
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY1}	2.7 V ≤ V_b ≤ EV_{DD0}, C_b = 30 pF, R_b = 1.4 kΩ	400 Note3			ns
SCKp high-level width	t_{KH1}	2.7 V ≤ V_b ≤ EV_{DD0}, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}/2 − 75			ns
SCKp low-level width	t_{KL1}	2.7 V ≤ V_b ≤ EV_{DD0}, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}/2 − 20			ns
SIp setup time (to 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 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 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 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 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 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. $t_{KCY1} \geq 4/f_{MCK}$ must also be satisfied.

CSI Mode Connection Diagram (During Communication at Different Potential)



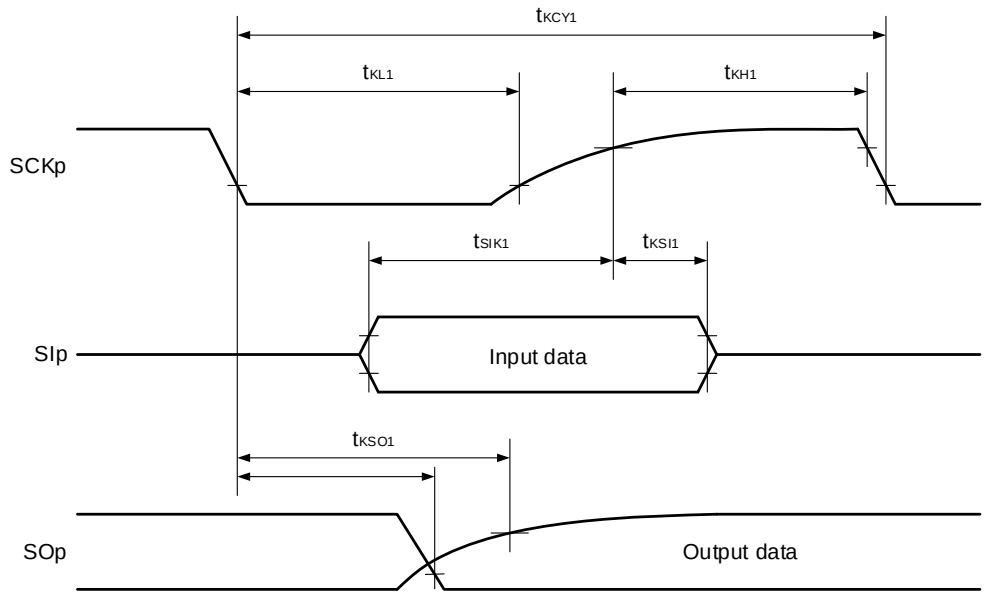
Caution Select TTL input buffer for the SIp pin, and select N-ch open-drain output mode for the SOp and SCKp pins.

Remarks 1. R_b [Ω]: Communication line (SCKp, SOp) pull-up resistance,
C_b [F]: Communication line (SOp, SCKp) load capacitance,
V_b [V]: Communication line voltage

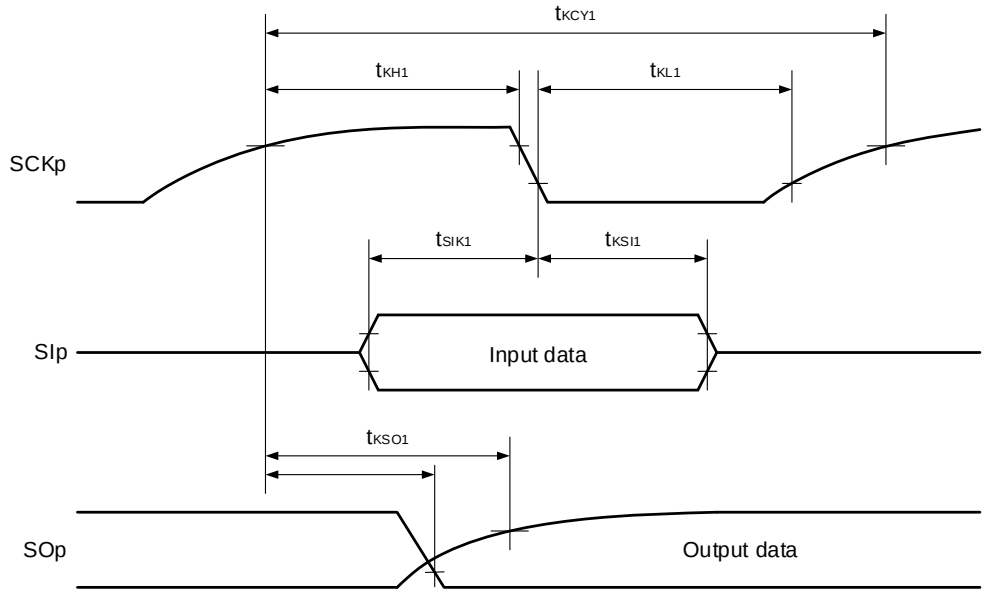
2. p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

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)



Remark p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

(10) During Communication at Different Potential (3 V Supply System) (CSI Mode) (Slave Mode, SCKp … External Clock Input, Normal Slew Rate)

($T_A = T_{OPR}$, 4.0 V ≤ $EV_{DD0} = EV_{DD1} = V_{DD}$ ≤ 5.5 V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

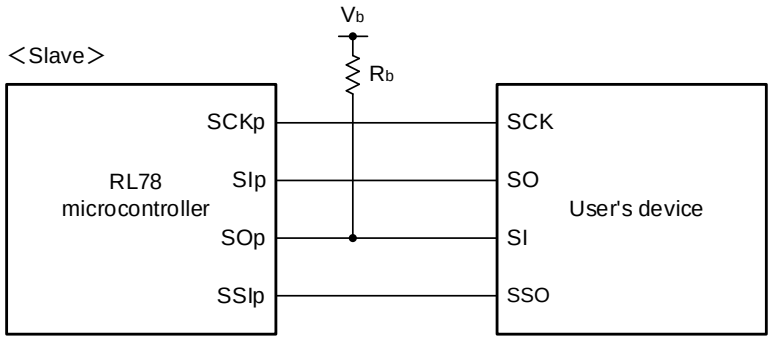
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY2}	2.7 V ≤ V_b ≤ EV_{DD0}	32 MHz < f_{MCK}				
			Grade 3 products	18/f_{MCK}			ns
			Grade 4 products	20/f_{MCK}			ns
			24 MHz < f_{MCK} ≤ 32 MHz				
			Grade 3 products	14/f_{MCK}			ns
			Grade 4 products	16/f_{MCK}			ns
			20 MHz < f_{MCK} ≤ 24 MHz				
			Grade 3 products	12/f_{MCK}			ns
			Grade 4 products	12/f_{MCK}			ns
			8 MHz < f_{MCK} ≤ 20 MHz				
			Grade 3 products	10/f_{MCK}			ns
			Grade 4 products	10/f_{MCK}			ns
			4 MHz < f_{MCK} ≤ 8 MHz				
			Grade 3 products	8/f_{MCK}			ns
			Grade 4 products	8/f_{MCK}			ns
			f_{MCK} ≤ 4 MHz				
			Grade 3 products	6/f_{MCK}			ns
			Grade 4 products	6/f_{MCK}			ns
SCKp high-level width, low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_b ≤ EV_{DD0}		t_{KCY2}/2 − 20			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}			90			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}			1/f_{MCK} + 50			ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	2.7 V ≤ V_b ≤ EV_{DD0}, C_b = 30 pF, R_b = 1.4 kΩ				2/f_{MCK} + 120	ns
SSIp setup time	t_{SSIK}	DAP = 0		120			ns
		DAP = 1		1/f_{MCK} + 120			ns
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 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 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 SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

CSI Mode Connection Diagram (During Communication at Different Potential)



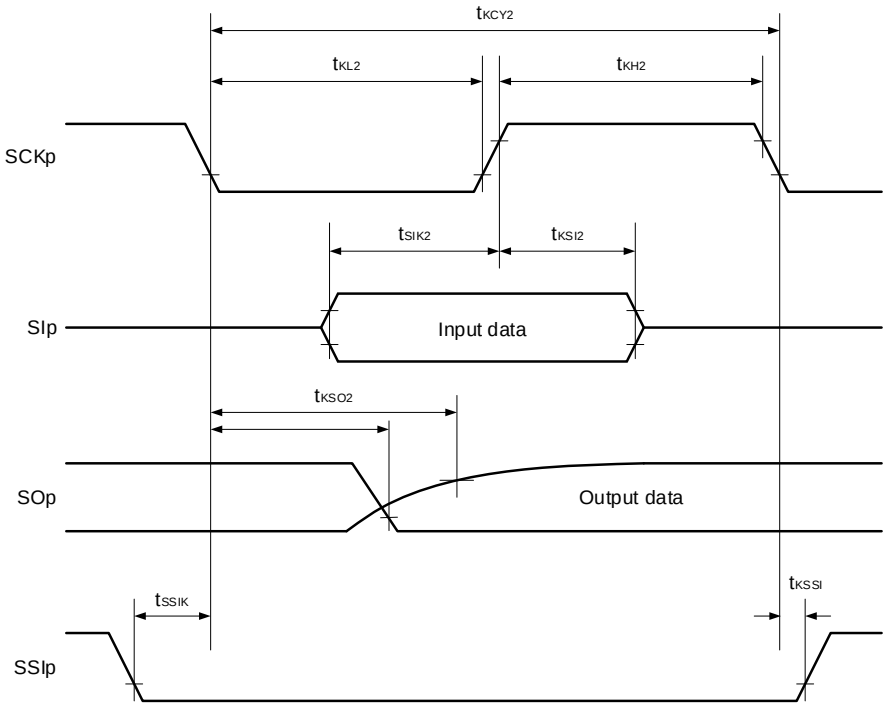
Caution Select TTL input buffer for the SIp, SCKp and SSIp pins, and select N-ch open-drain output mode for the SOp pin.

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

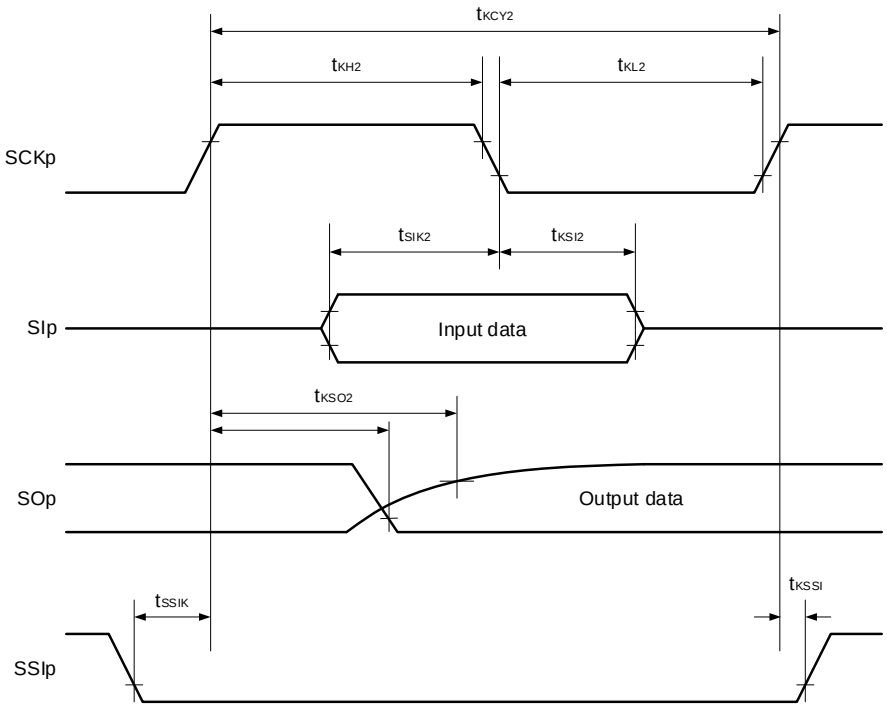
2. p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

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)



Remark p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

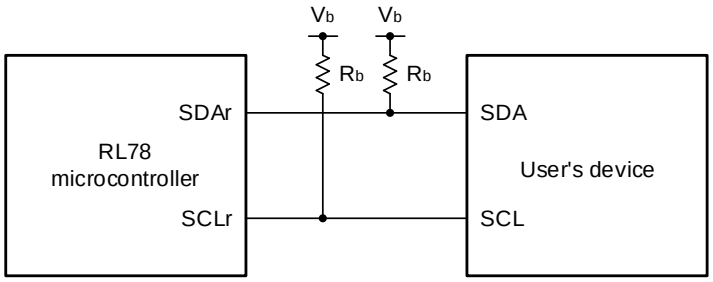
(11) During Communication at Different Potential (3 V Supply System) (Simplified I²C Mode) (SDAr: TTL Input Buffer Mode and N-ch Open-drain Output (EV_{DD} tolerance) Mode, SCLr: N-ch Open-drain Output (EV_{DD} Tolerance) Mode)

($T_A = T_{OPR}$, 4.0 V ≤ $EV_{DD0} = EV_{DD1} = V_{DD}$ ≤ 5.5 V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

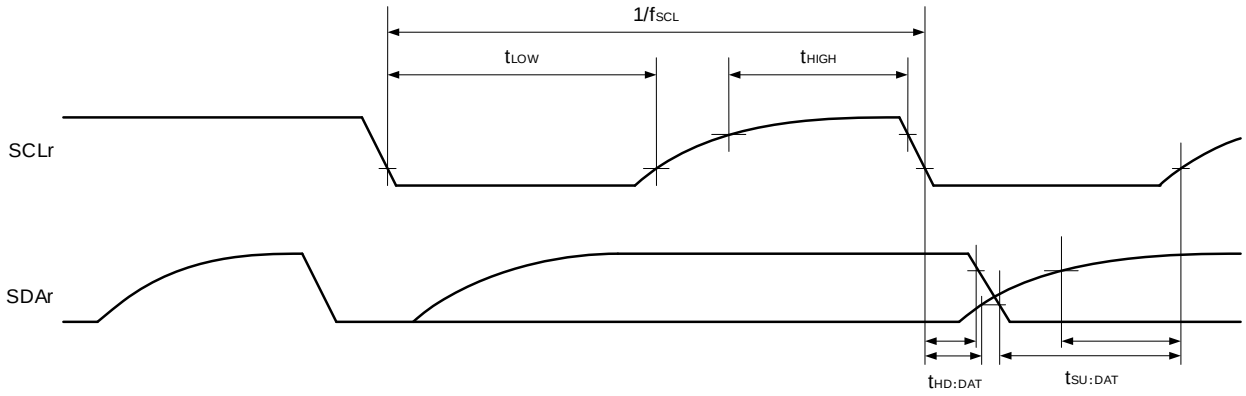
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ		400 Note	kHz
Hold time when SCLr = "L"	t_{LOW}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	1200		ns
Hold time when SCLr = "H"	t_{HIGH}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 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 $f_{SCL} \le f_{MCK}/4$ 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 TTL input buffer and N-ch open-drain output mode for the SDAr pin, and select 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

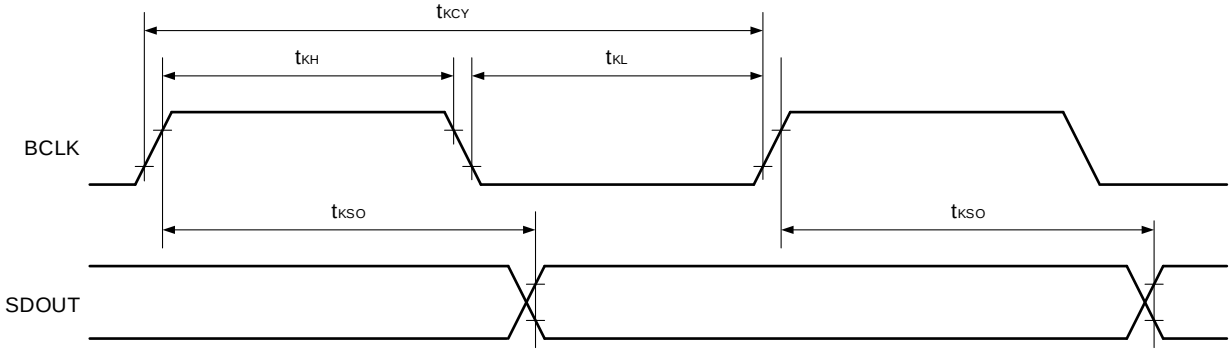
2. f_{MCK}: Serial array unit operation clock frequency

(12) During Communication at Same Potential (Simplified I²S Master Transmission) (CSI Mode, Slave Transmission Mode)

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = V_{DD}$ ≤ 5.5 V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

Parameter	Symbol	Conditions Note	MIN.	TYP.	MAX.	Unit
BCLK cycle time	t_{KCY}	Data rate = 2.5 MHz		400		ns
BCLK high-level width, low-level width	t_{KH}, t_{KL}		Value greater than 0.35 t_{KCY}			ns
SDOUT output delay time	t_{KSO}				Value less than 0.80 t_{KCY}	ns

Note The CPU/peripheral hardware clock frequency (f_{CLK}) must satisfy the condition of 8 MHz to 40 MHz.



Caution Set the I2SEN bit in the I2SER register to 1 (enable simple I2S communication function) and the TO121 and TO120 bits in the PWMDLY2 register to 01B (TO12 output is delayed by one cycle of f_{CLK}).

Remarks 1. BCLK: TO12 output of timer array unit 1, channel 2
SDOUT: SO10 output of serial array unit 1, channel 0
LRCLK: TO13 output of timer array unit 1, channel 3

2. Serial array unit 1, channel 0: CSI mode, slave transmission mode
Timer array unit 1, channel 2: Interval timer mode
Timer array unit 1, channel 3: External event counter mode

37.5.2 Serial Interface IICA

($T_A = T_{OPR}$, 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_{MCK}					0	1000	kHz
		Fast mode: 3.5 MHz ≤ f_{MCK}			0	400			kHz
		Normal mode: 1 MHz ≤ f_{MCK}	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

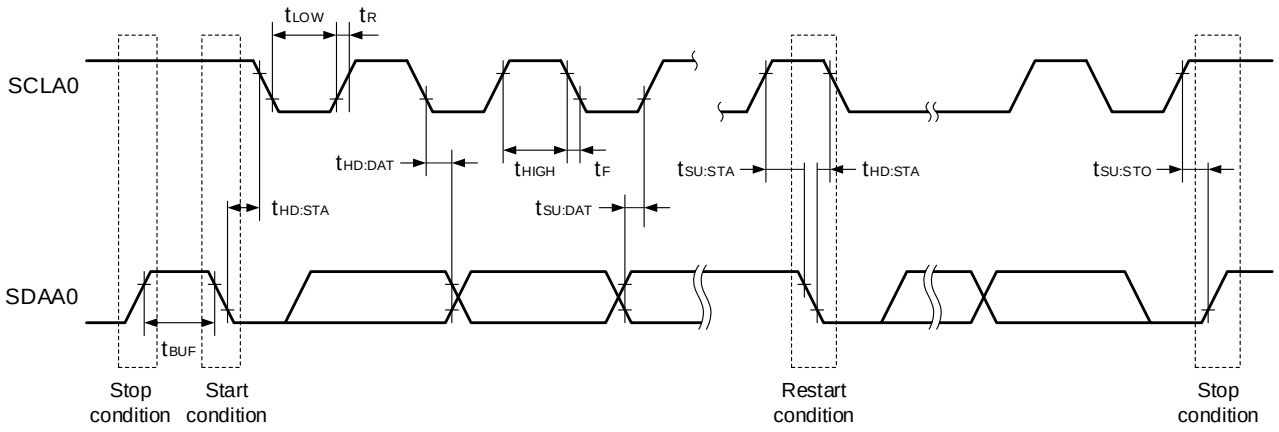
Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a wait state is inserted in the 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



37.5.3 On-chip Debug (UART)

($T_A = T_{OPR}$, 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

37.5.4 LIN/UART Module (RLIN3) UART Mode

($T_A = T_{OPR}$, 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 MHz to 40 MHz			4000	kbps
		SNOOZE mode	LIN communication clock source (f_{CLK}): 2 MHz to 40 MHz			9.6	

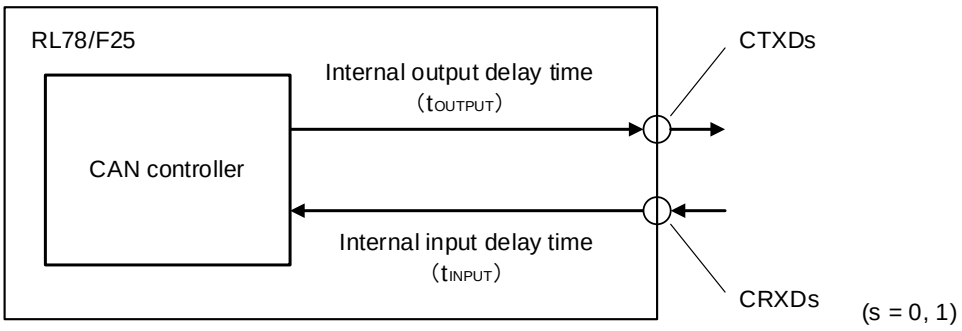
37.5.5 CAN-FD Communication Interface (RS-CANFD lite) Timing

($T_A = T_{OPR}$, 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	–	Classical CAN mode				1	Mbps
		CAN-FD mode	Data bit rate			5	Mbps
		CAN-FD mode	Nominal bit rate			1	Mbps
Internal delay time Note	t_{NODE}					50	ns

Note t_{NODE} = Internal input delay time (t_{INPUT}) + Internal output delay time (t_{OUTPUT})

Image of Internal Delay



37.6 Analog Characteristics

37.6.1 A/D Converter Characteristics

- When Reference voltage (+) = AV_{REFP}, Reference voltage (−) = AV_{REFM} = 0 V.
- Target ANI pin: ANI0 to ANI28, Internal reference voltage (+).
- Conversion mode: High-speed conversion mode

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V, Reference Voltage (+) = AV_{REFP}, Reference Voltage (−) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES				12	bit
Overall error Note 1	ABS	ANI0 to ANI21 Note 2, [4.5 V ≤ AV_{REFP}/AV_{DD} ≤ 5.5 V]			±5.0	LSB
		ANI0 to ANI21 Note 2, [2.7 V ≤ AV_{REFP}/AV_{DD} < 4.5 V]			±5.0	LSB
		ANI1, ANI2 Note 3, [4.5 V ≤ AV_{REFP}/AV_{DD} ≤ 5.5 V] [0.25 V ≤ V_{AIN} ≤ AV_{DD} − 0.25 V]			±6.0	LSB
		ANI1, ANI2 Note 3, [2.7 V ≤ AV_{REFP}/AV_{DD} < 4.5 V] [0.25 V ≤ V_{AIN} ≤ AV_{DD} − 0.25 V]			±8.0	LSB
		ANI22 to ANI28, [4.5 V ≤ AV_{REFP}/AV_{DD} ≤ 5.5 V]			±11	LSB
		ANI22 to ANI28, [2.7 V ≤ AV_{REFP}/AV_{DD} < 4.5 V]			±13	LSB
Integral linearity error Note 1	INL	ANI0 to ANI21			±3.0	LSB
		ANI22 to ANI28			±7.0	LSB
Differential linearity error Note 1	DNL	ANI0 to ANI21			±1.5	LSB
		ANI22 to ANI28			±3.5	LSB
Zero-scale error Note 1	ZSE	ANI0 to ANI21 Note 2			±4.5	LSB
		ANI22 to ANI28			±8.5	LSB
Full-scale error Note 1	FSE	ANI0 to ANI21 Note 2			±4.5	LSB
		ANI22 to ANI28			±8.5	LSB
Reference voltage (+)	AV_{REFP}		2.7		V_{DD}	V
Analog input voltage	V_{AIN}	ANI0 to ANI28	0		AV_{REFP}	V
Internal reference voltage (+)	V_{BGR}	2.7 V ≤ V_{DD} ≤ 5.5 V	1.42	1.48	1.54	V
Analog input slew rate	SR				0.4	V/µs
Operation clock	f_{AD}		2		40	MHz
Conversion time Note 4 (per 1 channel)	t_{CONV}	ADCLK = 40 MHz, input impedance ≤ 0.5 kΩ				
		ANI0 to ANI15 Note 2	1.125			µs
		ANI16 to ANI28	1.8			µs
		ANI1, ANI2 Note 3	1.85			µs

Notes 1. Excludes quantization error (±1/2 LSB).

2. In case that dedicated sample & hold circuit is not used.

3. In case that dedicated sample & hold circuit is used.

4. The A/D conversion processing time (t_{CONV}) consists of sampling time and time for conversion by successive approximation.

37.6.2 D/A Converter Characteristics

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					8	bit
Overall error	AINL	Rload = 4 MΩ	2.7 V ≤ AV_{DD} ≤ 5.5 V			±2.5	LSB
		Rload = 8 MΩ	2.7 V ≤ AV_{DD} ≤ 5.5 V			±2.5	LSB
Settling time	t_{SET}	Cload = 20 pF	2.7 V ≤ AV_{DD} ≤ 5.5 V			3	µs

37.6.3 Comparator Characteristics

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = AV_{DD} = V_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IOCMP}			±5	±40	mV
Input voltage range	V_{IVCMP}	IVCMP00 to IVCMP03	0		AV_{DD}	V
	V_{IVREF}	IVREF0	0		AV_{DD}	V
Input voltage slope Note 4	S_{IVCMP}	IVCMP00 to IVCMP03			±3	V/µs
Response time	t_{CR}, t_{CF}	Input amplitude ±100 mV		50	100	ns
Stabilization wait time during input channel switching Note 1	t_{WAIT}	Input amplitude ±100 mV	300			ns
Operation stabilization wait time Note 2, 3	t_{CMP}	2.7 V ≤ AV_{DD} ≤ 5.5 V	1			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.

3. When the D/A converter output is selected as the reference voltage, this is the time it takes for the comparator to satisfy the DC/AC characteristics after the D/A converter output value is changed.

4. Input voltage slope without noise filter.

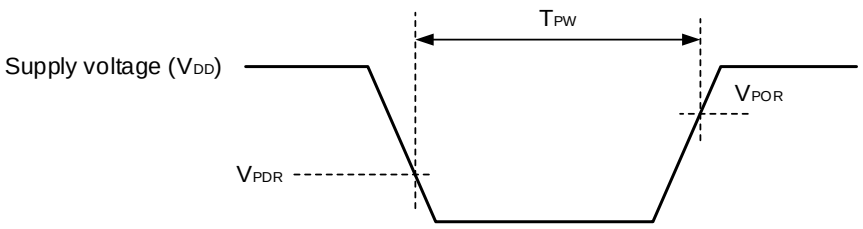
37.6.4 POR Circuit Characteristics

($T_A = T_{OPR}$, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage Note 1	V_{POR}, V_{PDR}		1.43	1.50	1.57	V
Minimum pulse width Note 2	T_{PW}		300			µs
Detection delay time	T_{PD}				350	µs

Notes 1. 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).

2. Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}.



37.6.5 LVD Circuit Characteristics

(1) LVD0 Detection Voltage of Reset Mode

($T_A = T_{OPR}$, $V_{PDR} \le EV_{DD0} = EV_{DD1} = V_{DD} \le 5.5$ V, $V_{SS} = EV_{SS0} = EV_{SS1} = 0$ V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD00}	Power supply rise time	3.84	3.96	4.08	V
			Power supply fall time	3.76	3.88	4.00	V
		V_{LVD01}	Power supply rise time	2.88	2.97	3.06	V
			Power supply fall time	2.82	2.91	3.00	V
Minimum pulse width		t_{LW}		500			μs
Detection delay time		t_{LD}				500	μs

(2) LVD1 Detection Voltage of Interrupt and Reset Mode

($T_A = T_{OPR}$, $V_{PDR} \le EV_{DD0} = EV_{DD1} = V_{DD} \le 5.5$ V, $V_{SS} = EV_{SS0} = EV_{SS1} = 0$ V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD10}	Power supply rise time	4.08	4.16	4.24	V
			Power supply fall time	4.00	4.08	4.16	V
		V_{LVD11}	Power supply rise time	3.88	3.96	4.04	V
			Power supply fall time	3.80	3.88	3.96	V
		V_{LVD12}	Power supply rise time	3.68	3.75	3.82	V
			Power supply fall time	3.60	3.67	3.74	V
		V_{LVD13}	Power supply rise time	3.48	3.55	3.62	V
			Power supply fall time	3.40	3.47	3.54	V
		V_{LVD14}	Power supply rise time	3.28	3.35	3.42	V
			Power supply fall time	3.20	3.27	3.34	V
		V_{LVD15}	Power supply rise time	3.07	3.13	3.19	V
			Power supply fall time	3.00	3.06	3.12	V
		V_{LVD16}	Power supply rise time	2.91	2.97	3.03	V
			Power supply fall time	2.85	2.91	2.97	V
		V_{LVD17}	Power supply rise time	2.76	2.82	2.87	V
			Power supply fall time	2.70	2.76	2.81	V
Minimum pulse width		t_{LW}		500			μs
Detection delay time		t_{LD}				500	μs

37.7 Power Supply Voltage Rising Time

($T_A = T_{OPR}$, $V_{SS} = EV_{SS0} = EV_{SS1} = 0$ V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum power supply voltage rising slope Note 1	S_{VRMAX}	0 V → V_{DD} (LVD0ENB = 0 or 1 Note 2)			50 Note 3	V/ms

Notes 1. Maintain the reset state by the voltage detection circuit (LVD0) or external reset up to the operating voltage range.

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.

37.8 Regulator Output Voltage Characteristics

($T_A = T_{OPR}$, $V_{SS} = EV_{SS0} = EV_{SS1} = 0$ V)

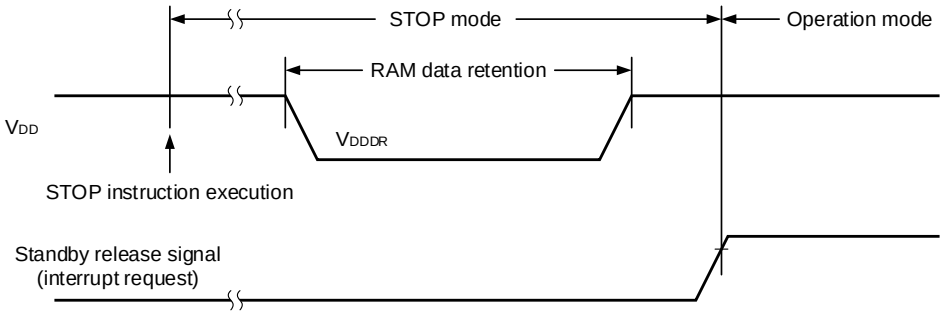
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
REGC output voltage	V_{OREGC}	C = 0.47 to 1 µF	1.45	1.50	1.56	V

37.9 RAM Data Retention Characteristics

($T_A = T_{OPR}$, $V_{SS} = EV_{SS0} = EV_{SS1} = 0$ V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.47 Note		5.5	V

Note This depends on the POR detection voltage. For a falling voltage, data in RAM are retained until the voltage reaches the level that triggers a POR reset but not once it reaches the level at which a POR reset is generated.



37.10 Flash Memory Programming Characteristics

($T_A = T_{OPR}$, 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}		2		40	MHz
Number of code flash rewrites Notes 1, 2, 3	Cerwr	Retained for 20 years, T_A = +85°C Note 4	1,000			Times
Number of data flash rewrites Notes 1, 2, 3		Retained for 20 years, T_A = +85°C Note 4	10,000			
		Retained for 5 years, T_A = +85°C Note 4	100,000			
Erase time	Terasa	Block erase	5			ms
Write time	Twrwa	1 word write	10			μs

Notes 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The starting point of the retaining years are after the erase.

2. When using flash memory programmer and Renesas Electronics self programming code.

3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

4. The average temperature for data retention.

(1) Code Flash Memory Processing Time

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = V_{DD}$ ≤ 5.5 V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

Item		f_{CLK} = 2 MHz		f_{CLK} = 4 MHz		f_{CLK} = 8 MHz		f_{CLK} = 16 MHz		Unit
	Conditions	TYP.	MAX.	TYP.	MAX.	TYP.	MAX.	TYP.	MAX.	
Programming time	4 bytes	52.0	470.0	43.0	400.0	38.0	360.0	36.0	340.0	μs
Erasure time	2 KB	7.8	265.0	6.5	240.0	5.9	225.0	5.8	220.0	ms
Blank checking time	4 bytes	–	19.2	–	13.2	–	10.3	–	9.1	μs
	2 KB	–	1310.0	–	660.0	–	335.0	–	270.0	μs

Item		f_{CLK} = 20 MHz		f_{CLK} = 32 MHz		f_{CLK} = 40 MHz		Unit
	Conditions	TYP.	MAX.	TYP.	MAX.	TYP.	MAX.	
Programming time	4 bytes	36.0	340.0	35.0	330.0	35.0	330.0	μs
Erasure time	2 KB	5.8	220.0	5.7	220.0	5.7	220.0	ms
Blank checking time	4 bytes	–	8.9	–	8.5	–	8.5	μs
	2 KB	–	270.0	–	245.0	–	245.0	μs

Caution The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

(2) Data Flash Memory Processing Time

($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = V_{DD}$ ≤ 5.5 V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

Item		f_{CLK} = 2 MHz		f_{CLK} = 4 MHz		f_{CLK} = 8 MHz		f_{CLK} = 16 MHz		Unit
	Conditions	TYP.	MAX.	TYP.	MAX.	TYP.	MAX.	TYP.	MAX.	
Programming time	1 byte	52.0	470.0	43.0	400.0	38.0	360.0	36.0	340.0	µs
Erasure time	1 KB	10.4	320.0	7.8	265.0	6.5	240.0	6.3	235.0	ms
Blank checking time	1 byte	–	19.2	–	13.2	–	10.3	–	9.1	µs
	1 KB	–	2605.0	–	1305.0	–	660.0	–	530.0	µs

Item		f_{CLK} = 20 MHz		f_{CLK} = 32 MHz		f_{CLK} = 40 MHz		Unit
	Conditions	TYP.	MAX.	TYP.	MAX.	TYP.	MAX.	
Programming time	1 byte	36.0	340.0	35.0	330.0	35.0	330.0	µs
Erasure time	1 KB	6.3	235.0	6.2	230.0	6.2	230.0	ms
Blank checking time	1 byte	–	8.9	–	8.5	–	8.5	µs
	1 KB	–	530.0	–	475.0	–	475.0	µs

Caution The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

37.11 Dedicated Flash Memory Programmer Communication (UART)

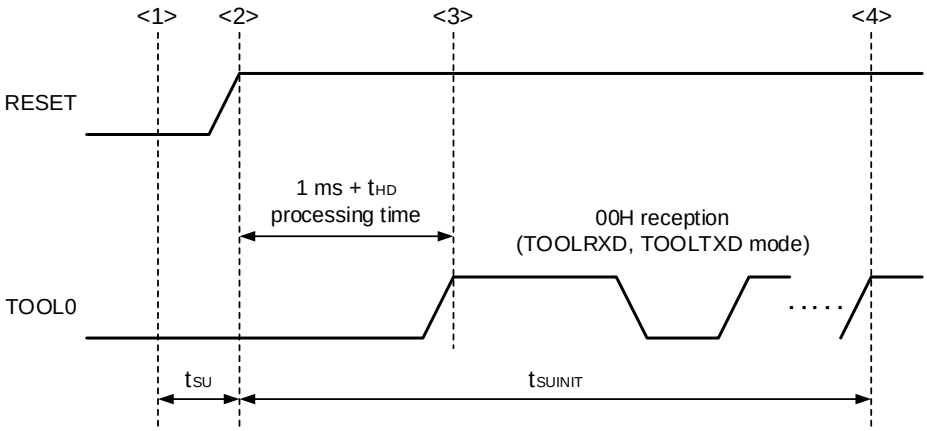
($T_A = T_{OPR}$, 2.7 V ≤ $EV_{DD0} = EV_{DD1} = V_{DD}$ ≤ 5.5 V, $V_{SS} = EV_{SS0} = EV_{SS1}$ = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate	–	During serial programming	115.2 k		1 M	bps

37.12 Timing of Entry to Flash Memory Programming Modes

($T_A = T_{OPR}$, 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
Time to complete the communication for the initial setting after the external reset is released	t_{SUINIT}	POR and LVD reset must be released before the external reset is released.			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	t_{SU}	POR and LVD reset must be released before the external reset is released.	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset must be released before the external reset is released.	1			ms



<1> The low level is input to the TOOL0 pin.

<2> The external reset is released (POR and LVD reset must be released before the external reset is released.).

<3> The TOOL0 pin is set to the high level.

<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: Communication for the initial setting must be completed within 100 ms after the external reset is released during this period

t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level

t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)

CHAPTER 38 ELECTRICAL SPECIFICATIONS (RL78/F22)

Cautions 1. RL78/F22 has an on-chip debug function 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. The pins mounted depending on the product. For details, refer to "1.5 Pin Configurations" and "2.1 Pin Function List".

3. The power supply voltage condition of 1.8 V ≤ AV_{DD} = V_{DD} < 2.7 V in each item applies to RL78/F22 Grade 3 products. In that case, the maximum operating clock (f_{CLK}) is 16 MHz.

38.1 Absolute Maximum Ratings

(1/3)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}, AV_{DD}	$V_{DD} = AV_{DD}$	−0.5 to +6.5	V
	V_{SS}, AV_{SS}	$V_{SS} = AV_{SS}$	−0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC	−0.3 to + 2.1 and −0.3 to V_{DD} + 0.3 Note 1	V
Input voltage	V_{I1}	P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P140	−0.3 to V_{DD} + 0.3 Note 2	V
	V_{I2}	P80 to P87, P90 to P92	−0.3 to AV_{DD} + 0.3 Note 2	V
	V_{I3}	P121 to P124, P137, RESET	−0.3 to V_{DD} + 0.3 Note 2	V
Output voltage	V_{O1}	P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P130, P140	−0.3 to V_{DD} + 0.3 Note 2	V
	V_{O2}	P80 to P87, P90 to P92	−0.3 to AV_{DD} + 0.3	V
Analog input voltage	V_{AI1}	ANI22 to ANI27	−0.3 to V_{DD} + 0.3 and −0.3 to AV_{DD} + 0.3 Notes 2, 3	V
	V_{AI2}	ANI0 to ANI10	−0.3 to AV_{DD} + 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_{DD} + 0.3.

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.

(2/3)

Parameter	Symbol	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P130, P140	−40	mA
		Total of all pins −170 mA	P40, P41, P120, P125	−70	mA
			P00, P10 to P17, P30 to P32, P60 to P63, P70 to P73, P130, P140	−100	mA
	I_{OH2}	Per pin	P80 to P87, P90 to P92	−0.5	mA
		Total of all pins		−2	mA
Output current, low	I_{OL1}	Per pin	P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P130, P140	40	mA
		Total of all pins 170 mA	P40, P41, P120, P125	70	mA
			P00, P10 to P17, P30 to P32, P60 to P63, P70 to P73, P130, P140	100	mA
	I_{OL2}	Per pin	P80 to P87, P90 to P92	1	mA
		Total of all pins		5	mA

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.

(3/3)

Parameter	Symbol	Conditions		Ratings	Unit
Positive injected current ($V_I > V_{DD}$) Note	I_{INJP}	Per pin	P00, P10, P11, P13 to P17, P30 to P32, P41, P60 to P63, P140	5	mA
			P70 to P73, P80 to P87, P90 to P92, P120, P125	2	mA
Negative injected current ($V_I < V_{SS}$) Note	I_{INJN}	Per pin	P00, P10, P11, P13 to P17, P30 to P32, P41, P60 to P63, P140	−5	mA
			P70 to P73, P80 to P87, P90 to P92, P120, P125	−0.5	mA
Sum of all positive injected currents Note	ΣI_{INJP}	Total of all pins	P00, P10, P11, P13 to P17, P30 to P32, P41, P60 to P63, P140	40	mA
			P70 to P73, P80 to P87, P90 to P92, P120, P125	10	mA
Sum of all negative injected currents Note	ΣI_{INJN}	Total of all pins	P00, P10, P11, P13 to P17, P30 to P32, P41, P60 to P63, P140	−40	mA
			P70 to P73, P80 to P87, P90 to P92, P120, P125	−2	mA
Total of all injected currents Note	$\Sigma\|I_{INJP}\| + \Sigma\|I_{INJN}\|$	Total of all pins	P00, P10, P11, P13 to P17, P30 to P32, P41, P60 to P63, P140	40	mA
			P70 to P73, P80 to P87, P90 to P92, P120, P125	10	mA
Operating ambient temperature	T_A	In normal operation mode	Grade 3 products	−40 to +105	°C
			Grade 4 products	−40 to +125	
		In flash memory programming mode	Grade 3 products	−40 to +105	°C
			Grade 4 products	−40 to +125	
Operating temperature	T_{OPR}		Grade 3 products	−40 to +105	°C
			Grade 4 products	−40 to +125	
Storage temperature	T_{stg}			−65 to +150	°C

Note Conditions: 1.8 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V

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.

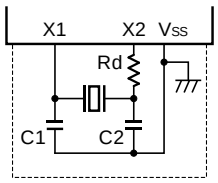
Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

2. V_I: Input voltage level to the port pins.

38.2 Oscillator Characteristics

38.2.1 Main System Clock Oscillator Characteristics

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 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	2.0		20.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V (Grade 3 products)	2.0		16.0	

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.

38.2.2 On-chip Oscillator Characteristics

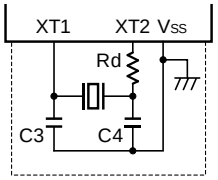
($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Note	f_{IH}	2.7 V ≤ V_{DD} ≤ 5.5 V	2		80	MHz
		1.8 V ≤ V_{DD} < 2.7 V (Grade 3 products)	2		16	
High-speed on-chip oscillator clock frequency accuracy	–		−1.0		+1.0	%
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/040C2H) and bits 0 to 2 of the HOCODIV register.

38.2.3 Subsystem Clock Oscillator Characteristics

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Resonator	Recommended Circuit	Item	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency (f_{XT})	1.8 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.

38.2.4 PLL Circuit Characteristics

($T_A = T_{OPR}$, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Resonator	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
PLL input enable clock frequency Note 1	f_{PLLI}	f_{MAIN}: 4.0 MHz	FMAINDIV[1:0] = 00B	3.92	4.0	4.08	MHz
		f_{MAIN}: 8.0 MHz	FMAINDIV[1:0] = 00B	7.84	8.0	8.16	MHz
		f_{MAIN}: 16.0 MHz	FMAINDIV[1:0] = 10B	7.84	8.0	8.16	MHz
		f_{MAIN}: 20.0 MHz	FMAINDIV[1:0] = 11B	4.90	5.0	5.10	MHz
PLL output frequency (center value)	f_{PLL}	f_{MAIN}: 20 MHz, PLLMULA = 0, PLLMUL = 1	PLLDIV0 = 0, FPLLDIV = 0, PLLDIV1 = 0	f_{PLLI} × 16/2			MHz
			PLLDIV0 = 0, FPLLDIV = 1, PLLDIV1 = 1	f_{PLLI} × 16			MHz
		f_{MAIN}: 4 MHz, PLLMULA = 1, PLLMUL = 1	PLLDIV0 = 0, FPLLDIV = 0, PLLDIV1 = 0	f_{PLLI} × 20/2			MHz
			PLLDIV0 = 0, FPLLDIV = 1, PLLDIV1 = 1	f_{PLLI} × 20			MHz
		f_{MAIN}: 8 MHz or 16 MHz, PLLMULA = 0, PLLMUL = 0	PLLDIV0 = 1, FPLLDIV = 0, PLLDIV1 = 0	f_{PLLI} × 12/4			MHz
			PLLDIV0 = 0, FPLLDIV = 0, PLLDIV1 = 1	f_{PLLI} × 12/2			MHz
		f_{MAIN}: 8 MHz or 16 MHz, PLLMULA = 0, PLLMUL = 1	PLLDIV0 = 1, FPLLDIV = 0, PLLDIV1 = 0	f_{PLLI} × 16/4			MHz
			PLLDIV0 = 0, FPLLDIV = 0, PLLDIV1 = 1	f_{PLLI} × 16/2			MHz
		f_{MAIN}: 8 MHz or 16 MHz, PLLMULA = 1, PLLMUL = 0	PLLDIV0 = 0, FPLLDIV = 0, PLLDIV1 = 0	f_{PLLI} × 10/2			MHz
			PLLDIV0 = 0, FPLLDIV = 1, PLLDIV1 = 1	f_{PLLI} × 10			MHz
Long-term jitter Note 2	t_{LJ}	term = 1 μs		−1		+1	ns
		term = 10 μs		−1		+1	ns
		term = 20 μs		−2		+2	ns

Notes 1. X1 clock or external main system clock can be selected as the PLL input clock.

2. Guaranteed by design, but not tested before shipment.

Caution When using Grade 3 products with 1.8 V ≤ V_{DD} < 2.7 V, the PLL circuit cannot be used.

Remark f_{MAIN}: Main system clock frequency.

38.3 DC Characteristics

38.3.1 Pin Characteristics

For the relationship between the port pins shown in the following tables and the products, refer to **CHAPTER 4 PORT FUNCTIONS**.

($T_A = T_{OPR}$, 1.8 V ≤ $V_{DD} = AV_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS}$ = 0 V) **(1/6)**

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for: P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P130, P140		4.0 V ≤ V_{DD} ≤ 5.5 V			−5.0	mA
				2.7 V ≤ V_{DD} < 4.0 V			−3.0	mA
				1.8 V ≤ V_{DD} < 2.7 V			−0.5	mA
		Per pin for: P10, P12, P14, P30, P120, P140 (special slew rate)		4.0 V ≤ V_{DD} ≤ 5.5 V			−0.6	mA
				2.7 V ≤ V_{DD} < 4.0 V			−0.2	mA
				1.8 V ≤ V_{DD} < 2.7 V			−0.2	mA
		Total of: P40, P41, P120, P125 (for duty factors ≤ 70% Note 2)		4.0 V ≤ V_{DD} ≤ 5.5 V			−20.0	mA
				2.7 V ≤ V_{DD} < 4.0 V			−10.0	mA
				1.8 V ≤ V_{DD} < 2.7 V			−5.0	mA
		Total of: P00, P10 to P17, P30 to P32, P60 to P63, P70 to P73, P130, P140 (for duty factors ≤ 70% Note 2)		4.0 V ≤ V_{DD} ≤ 5.5 V			−30.0	mA
				2.7 V ≤ V_{DD} < 4.0 V			−19.0	mA
				1.8 V ≤ V_{DD} < 2.7 V			−10.0	mA
		Total of all pins (for duty factors ≤ 70% Note 2)	Grade 3 products	4.0 V ≤ V_{DD} ≤ 5.5 V			−50.0	mA
				2.7 V ≤ V_{DD} < 4.0 V			−29.0	mA
				1.8 V ≤ V_{DD} < 2.7 V			−15.0	mA
			Grade 4 products	4.0 V ≤ V_{DD} ≤ 5.5 V			−42.0	mA
				2.7 V ≤ V_{DD} < 4.0 V			−29.0	mA
	I_{OH2}	Per pin for: P80 to P87, P90 to P92		1.8 V ≤ AV_{DD} ≤ 5.5 V			−0.1	mA
		Total of all pins (for duty factors ≤ 70% Note 2)		1.8 V ≤ AV_{DD} ≤ 5.5 V			−2.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} and AV_{DD} pins to an output pin.

2. These output current values are obtained under the condition that the duty factor is 70% or less.
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 $(I_{OH} \times 0.7) / (n \times 0.01)$

<Example> Where I_{OH} = −10.0 mA and n = 80%
Total output current of pins = $(-10.0 \times 0.7) / (80 \times 0.01) \approx -8.7$ 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.

Cautions 1. P10 to P17, P32, P60 to P63, P70 to P72, and P120 do not output high level in N-ch open-drain mode.

2. Operating voltage condition: 1.8 V ≤ $V_{DD} = AV_{DD}$ < 2.7 V is only available for Grade 3 products.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

($T_A = T_{OPR}$, $1.8\ V \le V_{DD} = AV_{DD} \le 5.5\ V$, $V_{SS} = AV_{SS} = 0\ V$) **(2/6)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low Note 1	I_{OL1}	Per pin for: P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P130, P140	$4.0\ V \le V_{DD} \le 5.5\ V$			8.5	mA
			$2.7\ V \le V_{DD} < 4.0\ V$			4.0	mA
			$1.8\ V \le V_{DD} < 2.7\ V$			0.6	mA
		Per pin for: P10, P12, P14, P30, P120, P140 (special slew rate)	$4.0\ V \le V_{DD} \le 5.5\ V$			0.59	mA
			$2.7\ V \le V_{DD} < 4.0\ V$			0.07	mA
			$1.8\ V \le V_{DD} < 2.7\ V$			0.07	mA
		Total of: P40, P41, P120, P125 (for duty factors ≤ 70% Note 2)	$4.0\ V \le V_{DD} \le 5.5\ V$			20.0	mA
			$2.7\ V \le V_{DD} < 4.0\ V$			15.0	mA
			$1.8\ V \le V_{DD} < 2.7\ V$			9.0	mA
		Total of: P00, P10 to P17, P30 to P32, P60 to P63, P70 to P73, P130, P140 (for duty factors ≤ 70% Note 2)	$4.0\ V \le V_{DD} \le 5.5\ V$			45.0	mA
			$2.7\ V \le V_{DD} < 4.0\ V$			35.0	mA
			$1.8\ V \le V_{DD} < 2.7\ V$			20.0	mA
		Total of all pins (for duty factors ≤ 70% Note 2)	$4.0\ V \le V_{DD} \le 5.5\ V$			65.0	mA
			$2.7\ V \le V_{DD} < 4.0\ V$			50.0	mA
			$1.8\ V \le V_{DD} < 2.7\ V$			29.0	mA
	I_{OL2}	Per pin for: P80 to P87, P90 to P92	$1.8\ V \le AV_{DD} \le 5.5\ V$			0.4	mA
		Total of all pins (for duty factors ≤ 70% Note 2)	$1.8\ V \le AV_{DD} \le 5.5\ V$			5.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} and AV_{SS} pins.

2. These output current values are obtained under the condition that the duty factor is 70% or less.
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 $(I_{OL} \times 0.7) / (n \times 0.01)$

<Example> Where I_{OL} = 10.0 mA and n = 80%
Total output current of pins = $(10.0 \times 0.7) / (80 \times 0.01) \approx 8.7$ 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.

Caution Operating voltage condition: $1.8\ V \le V_{DD} = AV_{DD} < 2.7\ V$ is only available for Grade 3 products.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

($T_A = T_{OPR}$, 1.8 V ≤ $V_{DD} = AV_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS}$ = 0 V) **(3/6)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P140 (Schmitt 1 mode)	4.0 V ≤ V_{DD} ≤ 5.5 V	0.65 V_{DD}		V_{DD} Note	V
			2.7 V ≤ V_{DD} < 4.0 V	0.7 V_{DD}		V_{DD} Note	V
			1.8 V ≤ V_{DD} < 2.7 V	0.7 V_{DD}		V_{DD} Note	V
	V_{IH2}	P10, P11, P13, P14, P16, P17, P30, P41, P60 to P63, P70, P71, P73, P120, P125 (Schmitt 3 mode)	4.0 V ≤ V_{DD} ≤ 5.5 V	0.8 V_{DD}		V_{DD} Note	V
			2.7 V ≤ V_{DD} < 4.0 V	0.85 V_{DD}		V_{DD} Note	V
			1.8 V ≤ V_{DD}< 2.7 V	0.85 V_{DD}		V_{DD} Note	V
	V_{IH3}	P10, P11, P13, P14, P16, P17, P30, P62, P63, P70, P71, P73, P125 (TTL mode)	4.0 V ≤ V_{DD} ≤ 5.5 V	2.2		V_{DD} Note	V
			2.7 V ≤ V_{DD} < 4.0 V	2.0		V_{DD} Note	V
			1.8 V ≤ V_{DD}< 2.7 V	1.5		V_{DD} Note	V
	V_{IH4}	P80 to P87, P90 to P92 (fixed to Schmitt 3 mode)	4.0 V ≤ AV_{DD} ≤ 5.5 V	0.8 AV_{DD}		AV_{DD}	V
			2.7 V ≤ AV_{DD} < 4.0 V	0.85 AV_{DD}		AV_{DD}	V
			1.8 V ≤ AV_{DD} < 2.7 V	0.85 AV_{DD}		AV_{DD}	V
	V_{IH5}	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
			1.8 V ≤ V_{DD} < 2.7 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
			1.8 V ≤ V_{DD} < 2.7 V	0.8 V_{DD}		V_{DD}	V
	V_{IH7}	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
			1.8 V ≤ V_{DD} < 2.7 V	0.85 V_{DD}		V_{DD}	V

Note For pins P10 to P17, P32, P60 to P63, P70 to P72, and P120, the maximum V_{IH} value is V_{DD} even when the pins are in N-ch open-drain mode and set to high-level output.

Caution Operating voltage condition: 1.8 V ≤ $V_{DD} = AV_{DD}$ < 2.7 V is only available for Grade 3 products.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

($T_A = T_{OPR}$, 1.8 V ≤ $V_{DD} = AV_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS}$ = 0 V) **(4/6)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, low	V_{IL1}	P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P140 (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
			1.8 V ≤ V_{DD} < 2.7 V	0		0.3 V_{DD}	V
	V_{IL2}	P10, P11, P13, P14, P16, P17, P30, P41, P60 to P63, P70, P71, P73, P120, P125 (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
			1.8 V ≤ V_{DD} < 2.7 V	0		0.4 V_{DD}	V
	V_{IL3}	P10, P11, P13, P14, P16, P17, P30, P62, P63, P70, P71, P73, P125 (TTL mode)	4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			2.7 V ≤ V_{DD} < 4.0 V	0		0.5	V
			1.8 V ≤ V_{DD} < 2.7 V	0		0.32	V
	V_{IL4}	P80 to P87, P90 to P92 (fixed to Schmitt 3 mode)	4.0 V ≤ AV_{DD} ≤ 5.5 V	0		0.5 AV_{DD}	V
			2.7 V ≤ AV_{DD} < 4.0 V	0		0.4 AV_{DD}	V
			1.8 V ≤ AV_{DD} < 2.7 V	0		0.4 AV_{DD}	V
	V_{IL5}	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
			1.8 V ≤ V_{DD} < 2.7 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
			1.8 V ≤ V_{DD} < 2.7 V	0		0.2 V_{DD}	V
	V_{IL7}	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
			1.8 V ≤ V_{DD} < 2.7 V	0		0.4 V_{DD}	V

Caution Operating voltage condition: 1.8 V ≤ $V_{DD} = AV_{DD}$ < 2.7 V is only available for Grade 3 products.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

($T_A = T_{OPR}$, 1.8 V ≤ $V_{DD} = AV_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS}$ = 0 V) **(5/6)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P130, P140 (normal slew rate)	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = −5.0 mA	V_{DD} − 0.9			V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = −3.0 mA	V_{DD} − 0.7			V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = −1.0 mA	V_{DD} − 0.5			V
			1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = −0.5 mA	V_{DD} − 0.5			V
	V_{OH2}	P80 to P87, P90 to P92	1.8 V ≤ AV_{DD} ≤ 5.5 V I_{OH2} = −100 μA	AV_{DD} − 0.5			V
	V_{OH3}	P10, P12, P14, P30, P120, P140 (special slew rate)	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH3} = −0.6 mA	V_{DD} − 0.8			V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH3} = −0.2 mA	V_{DD} − 0.5			V
			1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OH3} = −0.2 mA	V_{DD} − 0.5			V
Output voltage, low	V_{OL1}	P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P130, P140 (normal slew rate)	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 4.0 mA			0.4	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 4.0 mA			0.7	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA			0.4	V
			1.8 V ≤ V_{DD} ≤ 5.5 V I_{OL1} = 0.6 mA			0.4	V
	V_{OL2}	P80 to P87, P90 to P92	1.8 V ≤ AV_{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 ≤ V_{DD} ≤ 5.5 V, I_{OL3} = 0.6 mA			0.8	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL3} = 0.07 mA			0.5	V
			1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OL3} = 0.07 mA			0.5	V

Cautions 1. P10 to P17, P32, P60 to P63, P70 to P72, and P120 do not output high level in N-ch open-drain mode.

2. Operating voltage condition: 1.8 V ≤ $V_{DD} = AV_{DD}$ < 2.7 V is only available for Grade 3 products.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

($T_A = T_{OPR}$, 1.8 V ≤ $V_{DD} = AV_{DD}$ ≤ 5.5 V, $V_{SS} = AV_{SS}$ = 0 V) **(6/6)**

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P140	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P80 to P87, P90 to P92	$V_I = AV_{DD}$				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
	I_{LIH4}	P137, RESET	$V_I = V_{DD}$				1	μA
Input leakage current, low	I_{LIL1}	P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P140	$V_I = V_{SS}$				−1	μA
	I_{LIL2}	P80 to P87, P90 to P92	$V_I = AV_{SS}$				−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
	I_{LIL4}	P137, RESET	$V_I = V_{SS}$				−1	μA
Positive injected current Note	$I_{INJPRMS}$	P00, P10, P11, P13 to P17, P30 to P32, P41, P60 to P63, P140	Per pin, $V_I > V_{DD}$				0.4	mA
			Total of all pins, $V_I > V_{DD}$				4	mA
		P70 to P73, P80 to P87, P90 to P92, P120, P125	Per pin, $V_I > V_{DD} = AV_{DD}$				0.15	mA
			Total of all pins, $V_I > V_{DD} = AV_{DD}$				1	mA
On-chip pull-up resistance	R_U	P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P140	$V_I = V_{SS}$, in input port		10	20	100	kΩ

Note Guaranteed by design, but not tested before shipment.

Caution Operating voltage condition: 1.8 V ≤ $V_{DD} = AV_{DD}$ < 2.7 V is only available for Grade 3 products.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

2. V_I : Input voltage level to the port pins.

38.3.2 Supply Current Characteristics

(1) RL78/F22 (Grade 3 Products)

(T_A = −40 to +105°C, 1.8 V ≤ V_{DD} = AV_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V) **(1/2)**

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} = 80 MHz	f_{CLK} = 40 MHz Notes 3, 4		5.5	11.0	mA
					f_{IH} = 40 MHz	f_{CLK} = f_{IH} Notes 3, 4		5.3	10.0	mA
					f_{IH} = 16 MHz	f_{CLK} = f_{IH} Notes 3, 4		2.5	4.9	mA
					f_{IH} = 2 MHz	f_{CLK} = f_{IH} Notes 3, 4		0.85	1.9	mA
				Resonator operation	f_{MX} = 20 MHz	f_{CLK} = f_{MX} Notes 3, 5		2.9	5.6	mA
					f_{MX} = 16 MHz	f_{CLK} = f_{MX} Notes 3, 5		2.4	4.8	mA
					f_{MX} = 2 MHz	f_{CLK} = f_{MX} Notes 3, 5		0.75	1.8	mA
				Resonator operation (PLL operation) (PLL input clock = f_{MX})	f_{PLL} = 80 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Notes 3, 6		5.6	11.0	mA
					f_{PLL} = 40 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Notes 3, 6		5.4	10.0	mA
					f_{PLL} = 40 MHz, f_{MX} = 4 MHz	f_{CLK} = 40 MHz Notes 3, 6		5.2	9.5	mA
				Subsystem clock operation	f_{SUB} = 32.768 kHz	f_{CLK} = f_{SUB} Note 7		10.5	220	μA
				Low-speed on-chip oscillator clock operation	f_{IL} = 15 kHz	f_{CLK} = f_{IL} Note 8		8.6	220	μA

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. 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 and A/D converter 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, and with setting of ADCEN = 0 or ADSLP = 1.

8. When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped, and with setting of ADCEN = 0 or ADSLP = 1.

Caution When 1.8 V ≤ V_{DD} < 2.7 V, the maximum operating clock (f_{CLK}) is 16 MHz.

Remark f_{MX}: High-speed system clock frequency
f_{SUB}: Subsystem clock frequency
f_{PLL}: PLL clock frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency

(T_A = −40 to +105°C, 1.8 V ≤ V_{DD} = AV_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V) (2/2)

Items	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Notes 1, 3	I_{DD2}	HALT mode Note 2	High-speed on-chip oscillator clock operation	f_{IH} = 80 MHz	f_{CLK} = 40 MHz Note 5		1.8	6.5	mA
				f_{IH} = 40 MHz	f_{CLK} = f_{IH} Note 5		1.5	5.5	mA
				f_{IH} = 16 MHz	f_{CLK} = f_{IH} Note 5		0.8	3.0	mA
				f_{IH} = 2 MHz	f_{CLK} = f_{IH} Note 5		0.4	1.6	mA
			Resonator operation	f_{MX} = 20 MHz	f_{CLK} = f_{MX} Note 6		0.9	3.4	mA
				f_{MX} = 16 MHz	f_{CLK} = f_{MX} Note 6		0.75	3.0	mA
				f_{MX} = 2 MHz	f_{CLK} = f_{MX} Note 6		0.3	1.6	mA
			Resonator operation (PLL operation) (PLL input clock = f_{MX})	f_{PLL} = 80 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Note 7		1.8	6.5	mA
				f_{PLL} = 40 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Note 7		1.7	5.5	mA
				f_{PLL} = 40 MHz, f_{MX} = 4 MHz	f_{CLK} = 40 MHz Note 7		1.5	5.0	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz	f_{CLK} = f_{SUB} Note 8		6.5	120	μA
			Low-speed on-chip oscillator clock operation	f_{IL} = 15 kHz	f_{CLK} = f_{IL} Note 9		6.5	120	μA
	I_{DD3}	STOP mode Note 4	T_A = +25°C				0.5		μA
			T_A = +50°C					8.0	
			T_A = +70°C					18.0	
			T_A = +105°C					75.0	
	I_{SNOZ}	SNOOZE mode	DTC operation				4.0		mA

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. 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, and A/D converter are stopped.

4. When high-speed system clock, subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped. When the RAMSDMD bit and RAMSDS bit of the PSMCR register are set to 11B (shutdown mode).

5. When high-speed system clock, subsystem clock, PLL clock, and low-speed on-chip oscillator clock are stopped.

6. When subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

7. When subsystem clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

8. When high-speed system clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped, and with setting of ADCEN = 0 or ADSLP = 1.

9. When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped, and with setting of ADCEN = 0 or ADSLP = 1.

Caution When 1.8 V ≤ V_{DD} < 2.7 V, the maximum operating clock (f_{CLK}) is 16 MHz.

Remark f_{MX}: High-speed system clock frequency
f_{SUB}: Subsystem clock frequency
f_{PLL}: PLL clock frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency

(T_A = −40 to +105°C, 1.8 V ≤ V_{DD} = AV_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Window watchdog timer operating current	I_{WDT} Notes 1, 2	f_{WDT} = 15 kHz			0.3		µA
A/D converter operating current	I_{ADC} Note 3	When conversion at maximum speed	AV_{DD} = V_{DD} = 5.0 V		1.3	1.6	mA
		When low current conversion	AV_{DD} = V_{DD} = 2.7 V		0.8	1.0	mA
		When internal reference voltage is selected Note 4			128		µA
Sample-and-hold circuit operating current	I_{ADSH} Note 5				0.8	1.2	mA
LVD operating current	I_{LVD} Note 6				0.045		µA
BGO operating current	I_{BGO} Note 7				2.5	12.2	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 15 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. Operating current that increases when the internal reference voltage is selected. This current flows even when conversion is stopped.

5. Operating current that increases when the sample-and-hold circuit is used. This current flows for each analog input channel.

6. 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.

7. Current increased by the BGO operation. The current value is the sum of I_{DD1} and I_{BGO} when the BGO operates in operation mode.

(2) RL78/F22 (Grade 4 Products)

(T_A = −40 to +125°C, 2.7 V ≤ V_{DD} = AV_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V) **(1/2)**

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} = 80 MHz	f_{CLK} = 40 MHz Notes 3, 4		5.5	11.0	mA
					f_{IH} = 40 MHz	f_{CLK} = f_{IH} Notes 3, 4		5.3	10.0	mA
					f_{IH} = 2 MHz	f_{CLK} = f_{IH} Notes 3, 4		0.85	2.0	mA
				Resonator operation	f_{MX} = 20 MHz	f_{CLK} = f_{MX} Notes 3, 5		2.9	5.6	mA
					f_{MX} = 2 MHz	f_{CLK} = f_{MX} Notes 3, 5		0.75	1.9	mA
				Resonator operation (PLL operation) (PLL input clock = f_{MX})	f_{PLL} = 80 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Notes 3, 6		5.6	11.0	mA
					f_{PLL} = 40 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Notes 3, 6		5.4	10.0	mA
					f_{PLL} = 40 MHz, f_{MX} = 4 MHz	f_{CLK} = 40 MHz Notes 3, 6		5.2	9.5	mA
				Subsystem clock operation	f_{SUB} = 32.768 kHz	f_{CLK} = f_{SUB} Note 7		10.5	400	µA
				Low-speed on-chip oscillator clock operation	f_{IL} = 15 kHz	f_{CLK} = f_{IL} Note 8		8.6	400	µA

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. 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 and A/D converter 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, and with setting of ADCEN = 0 or ADSLP = 1.

8. When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped, and with setting of ADCEN = 0 or ADSLP = 1.

Remark f_{MX}: High-speed system clock frequency
f_{SUB}: Subsystem clock frequency
f_{PLL}: PLL clock frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency

(T_A = −40 to +125°C, 2.7 V ≤ V_{DD} = AV_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V) (2/2)

Items	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Notes 1, 3	I_{DD2}	HALT mode Note 2	High-speed on-chip oscillator clock operation	f_{IH} = 80 MHz	f_{CLK} = 40 MHz Note 5		1.8	6.5	mA
				f_{IH} = 40 MHz	f_{CLK} = f_{IH} Note 5		1.5	5.5	mA
				f_{IH} = 2 MHz	f_{CLK} = f_{IH} Note 5		0.4	1.7	mA
			Resonator operation	f_{MX} = 20 MHz	f_{CLK} = f_{MX} Note 6		0.9	3.4	mA
				f_{MX} = 2 MHz	f_{CLK} = f_{MX} Note 6		0.3	1.7	mA
			Resonator operation (PLL operation) (PLL input clock = f_{MX})	f_{PLL} = 80 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Note 7		1.8	6.5	mA
				f_{PLL} = 40 MHz, f_{MX} = 20 MHz	f_{CLK} = 40 MHz Note 7		1.7	5.5	mA
				f_{PLL} = 40 MHz, f_{MX} = 4 MHz	f_{CLK} = 40 MHz Note 7		1.5	5.0	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz	f_{CLK} = f_{SUB} Note 8		6.5	210	μA
			Low-speed on-chip oscillator clock operation	f_{IL} = 15 kHz	f_{CLK} = f_{IL} Note 9		6.5	210	μA
	I_{DD3}	STOP mode Note 4	T_A = +25°C				0.5		μA
			T_A = +50°C					8.0	
			T_A = +70°C					18.0	
			T_A = +105°C					75.0	
			T_A = +125°C					175.0	
	I_{SNOZ}	SNOOZE mode	DTC operation				4.0		mA

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. 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, and A/D converter are stopped.

4. When high-speed system clock, subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped. When the RAMSDMD bit and RAMSDS bit of the PSMCR register are set to 11B (shutdown mode).

5. When high-speed system clock, subsystem clock, PLL clock, and low-speed on-chip oscillator clock are stopped.

6. When subsystem clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

7. When subsystem clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped.

8. When high-speed system clock, PLL clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock are stopped, and with setting of ADCEN = 0 or ADSLP = 1.

9. When high-speed system clock, subsystem clock, PLL clock, and high-speed on-chip oscillator clock are stopped, and with setting of ADCEN = 0 or ADSLP = 1.

Remark f_{MX}: High-speed system clock frequency
f_{SUB}: Subsystem clock frequency
f_{PLL}: PLL clock frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency

(T_A = −40 to +125°C, 2.7 V ≤ V_{DD} = AV_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Window watchdog timer operating current	I_{WDT} Notes 1, 2	f_{WDT} = 15 kHz			0.3		µA
A/D converter operating current	I_{ADC} Note 3	When conversion at maximum speed	AV_{DD} = V_{DD} = 5.0 V		1.3	1.6	mA
		When internal reference voltage is selected Note 4			128		µA
Sample-and-hold circuit operating current	I_{ADSH} Note 5				0.8	1.2	mA
LVD operating current	I_{LVD} Note 6				0.045		µA
BGO operating current	I_{BGO} Note 7				2.5	12.2	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 15 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. Operating current that increases when the internal reference voltage is selected. This current flows even when conversion is stopped.

5. Operating current that increases when the sample-and-hold circuit is used. This current flows for each analog input channel.

6. 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.

7. Current increased by the BGO operation. The current value is the sum of I_{DD1} and I_{BGO} when the BGO operates in operation mode.

38.4 AC Characteristics

38.4.1 Basic Operation

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 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		0.025		0.5	μs
		High-speed system clock operation		0.05		0.5	μs
		PLL clock operation		0.025		0.5	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		0.025		0.5	μs
CPU/peripheral hardware clock frequency	f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V		0.025		66.6	μs
		1.8 V ≤ V_{DD} < 2.7 V		0.0625		66.6	μs
External system clock frequency	f_{EX}			2.0		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 TI13 input high-level width, low-level width	t_{TIH}, t_{TIL}			1/f_{MCK} + 10			ns
TO00 to TO07, TO10 to TO13, TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1, TRJIO0, TRJO0 output frequency	f_{TO}	Normal slew rate, C = 30 pF	4.0 V ≤ V_{DD} ≤ 5.5 V			16	MHz
			2.7 V ≤ V_{DD} < 4.0 V			8	MHz
			1.8 V ≤ V_{DD} < 2.7 V			4	MHz
		TO01, TO06, TO07, TO11, TO13, TRDIOC0, TRDIOD0, TRDIOD1, TRJO0 only, Special slew rate, C = 30 pF				2	MHz
PCLBUZ0 output frequency	f_{PCL}	Normal slew rate, C = 30 pF	4.0 V ≤ V_{DD} ≤ 5.5 V			16	MHz
			2.7 V ≤ V_{DD} < 4.0 V			8	MHz
			1.8 V ≤ V_{DD} < 2.7 V			4	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_{TJIH}, t_{TJIL}	TRJIO0		40			ns
Timer RDe input high-level, low-level width	t_{TDIH}, t_{TDIL}	TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1, TRDCLK0, TRD0RES, TRD1RES		3/f_{TRD}			ns
Timer RDe pulse output forced cutoff signal low-level width	t_{TDSIL}	P137/INTP0	2 MHz < f_{CLK} ≤ 40 MHz	1			μs
			f_{CLK} ≤ 2 MHz	1/f_{CLK} + 1			μs

Cautions 1. Excluding the error in oscillation frequency accuracy.

2. Operating voltage condition: 1.8 V ≤ V_{DD} = AV_{DD} < 2.7 V is only available for Grade 3 products.

Remark f_{MCK}: Timer array unit operation clock frequency

f_{TRD}: Timer RDe operation clock frequency

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

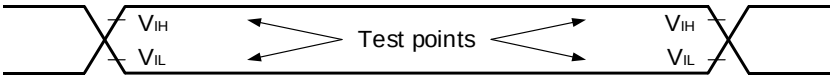
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP9 Note 1		1			μs
KR0 to KR7 key interrupt input low-level width	t_{KR}			250			ns
RESET low-level width	t_{RSL}	Note 1		10			μs
Port output rise time, port output fall time	t_{RO}, t_{FO}	P00, P10 to P17, P30 to P32, P40, P41, P60 to P63, P70 to P73, P120, P125, P130, P140 (normal slew rate) C = 30 pF	4.0 V ≤ V_{DD} ≤ 5.5 V			25	ns
			2.7 V ≤ V_{DD} < 4.0 V			55	ns
			1.8 V ≤ V_{DD} < 2.7 V			65	ns
		P10, P12, P14, P30, P120, P140 (special slew rate) C = 30 pF	4.0 V ≤ V_{DD} ≤ 5.5 V		25 Note 2	60	ns
			2.7 V ≤ V_{DD} < 4.0 V			100	ns
			1.8 V ≤ V_{DD} < 2.7 V			120	ns

Notes 1. Pins RESET, INTP0 to INTP3 have noise filters for transient levels lasting less than 100 ns.

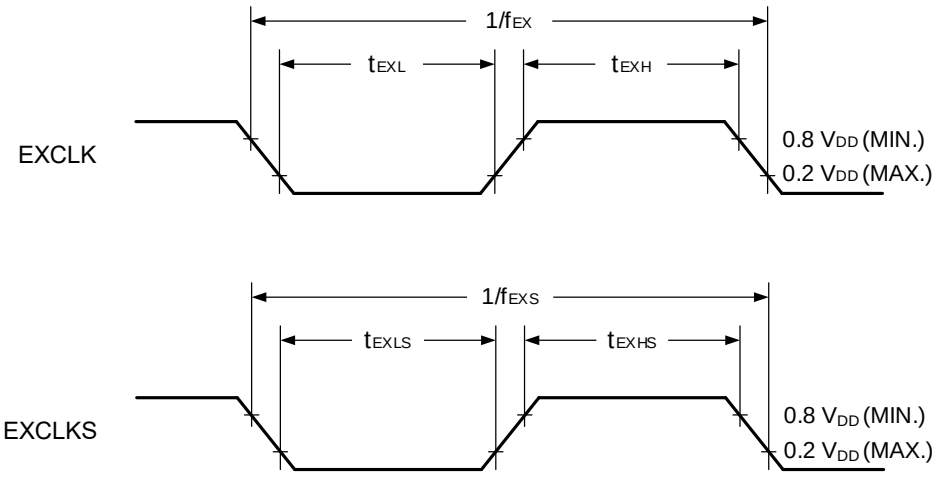
2. T_A = +25°C, V_{DD} = 5.0 V

Caution Operating voltage condition: 1.8 V ≤ V_{DD} = AV_{DD} < 2.7 V is only available for Grade 3 products.

AC Timing Test Points

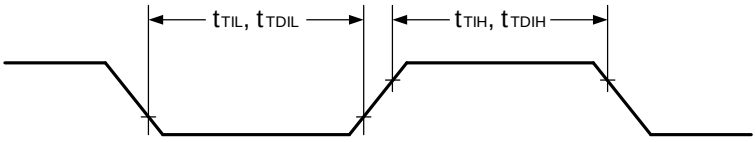


External System Clock Timing

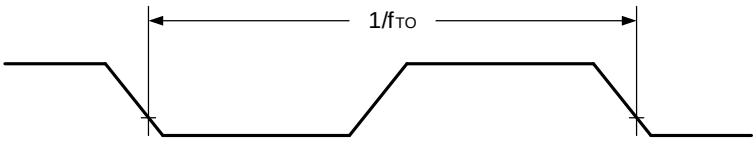


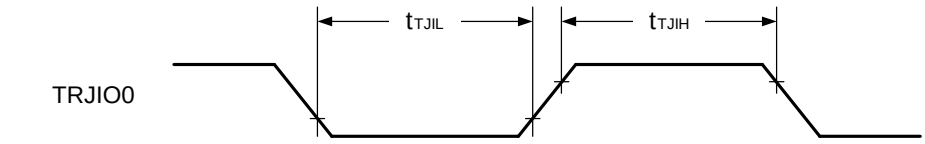
TI/TO Timing

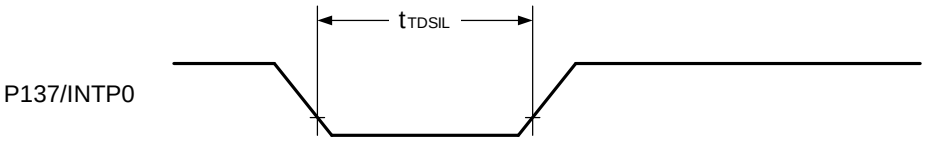
TI00 to TI07, TI10 to TI13, TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1, TRDCLK0, TRD0RES, TRD1RES



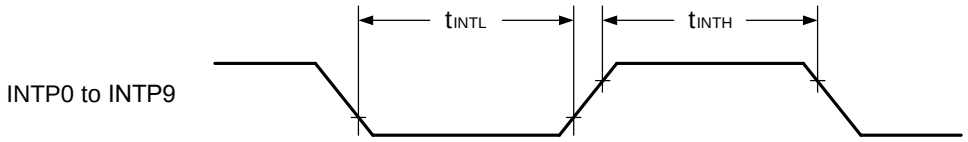
TO00 to TO07, TO10 to TO13, TRDIOA0, TRDIOA1, TRDIOB0, TRDIOB1, TRDIOC0, TRDIOC1, TRDIOD0, TRDIOD1, TRJIO0, TRJO0



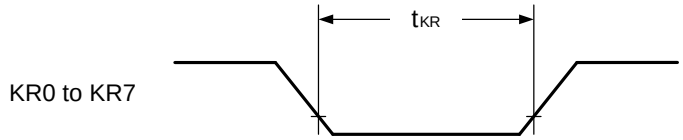




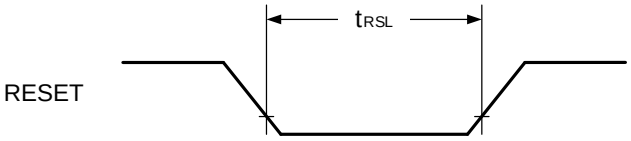
Interrupt Request Input Timing



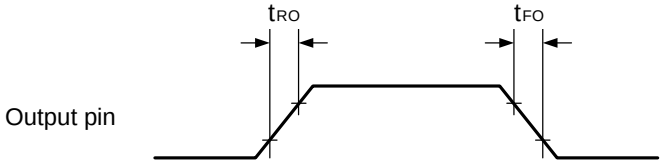
Key Interrupt Input Timing



RESET Input Timing



Output Rising and Falling Timing



38.5 Peripheral Functions Characteristics

38.5.1 Serial Array Unit

(1) During Communication at Same Potential (UART mode) (Dedicated Baud Rate Generator Output)

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

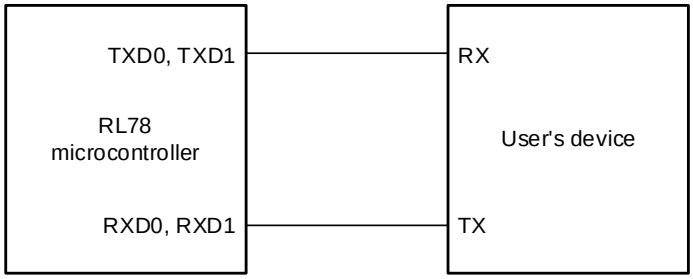
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Transfer rate	–					f_{MCK}/6	bps
		f_{CLK} = 40 MHz, $f_{MCK} = f_{CLK}$ Note	Normal slew rate			6.6	Mbps
			Special slew rate			2	Mbps

Note The operating frequency and maximum transfer rate (normal slew rate) of the CPU/peripheral hardware clock (f_{CLK}) for each product are shown below.

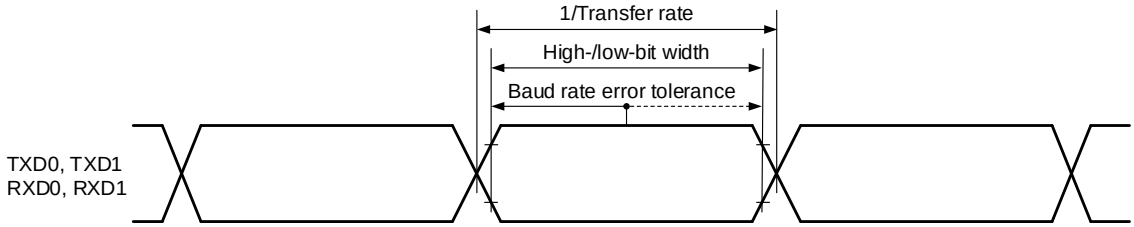
Grade 3 products: 40 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V), transfer rate: MAX. 6.6 Mbps
16 MHz (1.8 V ≤ V_{DD} < 2.7 V), transfer rate: MAX. 2.6 Mbps

Grade 4 products: 40 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V), transfer rate: MAX. 6.6 Mbps

UART Mode Connection Diagram (During Communication at Same Potential)



UART Mode Bit Width (During Communication at Same Potential) (Reference)



Caution Select normal input buffer for the RXD0 and RXD1 pins, and select normal output mode for the TXD0 and TXD1 pins.

Remark f_{MCK}: Serial array unit operation clock frequency

(2) During Communication at Same Potential (CSI mode) (Master Mode, SCKp … Internal Clock Output, Normal Slew Rate)

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY1}	Grade 3 products		100 Note 5			ns
		Grade 4 products		150 Note 5			ns
SCKp high-level width, low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 − 12			ns
		2.7 V ≤ V_{DD} < 4.0 V		t_{KCY1}/2 − 18			ns
		1.8 V ≤ V_{DD} < 2.7 V		t_{KCY1}/2 − 50			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	Grade 3 products	4.0 V ≤ V_{DD} ≤ 5.5 V	33			ns
			2.7 V ≤ V_{DD} < 4.0 V	44			ns
			1.8 V ≤ V_{DD} < 2.7 V	110			ns
		Grade 4 products	4.0 V ≤ V_{DD} ≤ 5.5 V	44			ns
			2.7 V ≤ V_{DD} < 4.0 V	55			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}			30			ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 30 pF Note 4				30	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.
The SIp setup time becomes "to 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 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 SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

5. $t_{KCY1} \geq 4/f_{MCK}$ must also be satisfied.
Grade 3 products: 40 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V), transfer rate: MAX. 10.0 Mbps (f_{MCK} = 40 MHz, t_{KCY1} = 100 ns)
16 MHz (1.8 V ≤ V_{DD} < 2.7 V), transfer rate: MAX. 4.0 Mbps (f_{MCK} = 16 MHz, t_{KCY1} = 250 ns)
Grade 4 products: 40 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V), transfer rate: MAX. 6.6 Mbps (f_{MCK} = 40 MHz, t_{KCY1} = 150 ns)

Cautions 1. Select normal input buffer for the SIp pin, and select normal output mode for the SOp and SCKp pins.

2. Operating voltage condition: 1.8 V ≤ V_{DD} < 2.7 V is only available for Grade 3 products.

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, SCKp … Internal Clock Output, Special Slew Rate)

($T_A = T_{OPR}$, 4.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY1}		500 Note 5			ns
SCKp high-level width, low-level width	t_{KH1}, t_{KL1}		t_{KCY1}/2 − 60			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}		120			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}		80			ns
Delay time from 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 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 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 SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

5. $t_{KCY1} \geq 4/f_{MCK}$ must also be satisfied.

Caution Select normal input buffer for the SIp pin, and select normal output mode and special slew rate for the SOp and SCKp pins.

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, SCKp … External Clock Input, Normal Slew Rate)

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY2}	2.7 V ≤ V_{DD} < 5.5 V	f_{MCK} > 32 MHz	10/f_{MCK} Note 5			ns
			f_{MCK} ≤ 32 MHz	8/f_{MCK} Note 5			ns
		1.8 V ≤ V_{DD} < 2.7 V	f_{MCK} ≤ 16 MHz	6/f_{MCK} and 750 Note 5			ns
SCKp high-level width, low-level width	t_{KH2}, t_{KL2}			t_{KCY2}/2			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}			1/f_{MCK} + 20			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}			1/f_{MCK} + 31			ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	4.0 V ≤ V_{DD} ≤ 5.5 V			2/f_{MCK} + 44	ns
			2.7 V ≤ V_{DD} < 4.0 V			2/f_{MCK} + 57	ns
			1.8 V ≤ V_{DD} < 2.7 V			2/f_{MCK} + 110	ns
SSIp setup time	t_{SSIK}	DAP = 0		120			ns
		DAP = 1		1/f_{MCK} + 120			ns
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 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 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 SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

5. The operating frequency and maximum transfer rate for each product are shown below.
Grade 3 products: 40 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V), transfer rate: MAX. 4.0 Mbps (f_{MCK} = 40 MHz, t_{KCY2} = 250 ns)
16 MHz (1.8 V ≤ V_{DD} < 2.7 V), transfer rate: MAX. 1.3 Mbps (f_{MCK} = 16 MHz, t_{KCY1} = 750 ns)
Grade 4 products: 40 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V), transfer rate: MAX. 4.0 Mbps (f_{MCK} = 40 MHz, t_{KCY1} = 250 ns)

Cautions 1. Select normal input buffer for the SIp, SCKp and SSIp pins, and select normal output mode for the SOp pin.

2. Operating voltage condition: 1.8 V ≤ V_{DD} < 2.7 V is only available for Grade 3 products.

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, SCKp … External Clock Input, Special Slew Rate)

($T_A = T_{OPR}$, 4.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
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
SCKp high-level width, low-level width	t_{KH2}, t_{KL2}		t_{KCY2}/2			ns
SIp setup time (to SCKp↑) Note1	t_{SIK2}		1/f_{MCK} + 50			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}		1/f_{MCK} + 50			ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4			2/f_{MCK} + 80	ns
SSIp setup time	t_{SSIK}	DAP = 0	120			ns
		DAP = 1	1/f_{MCK} + 120			ns
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 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 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 SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

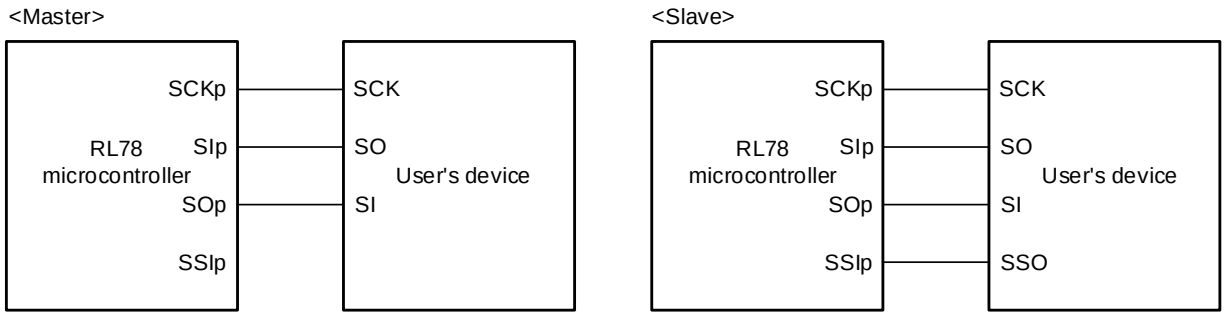
4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select normal input buffer for the SIp, SCKp and SSIp pins, and select normal output mode and special slew rate for the SOp pin.

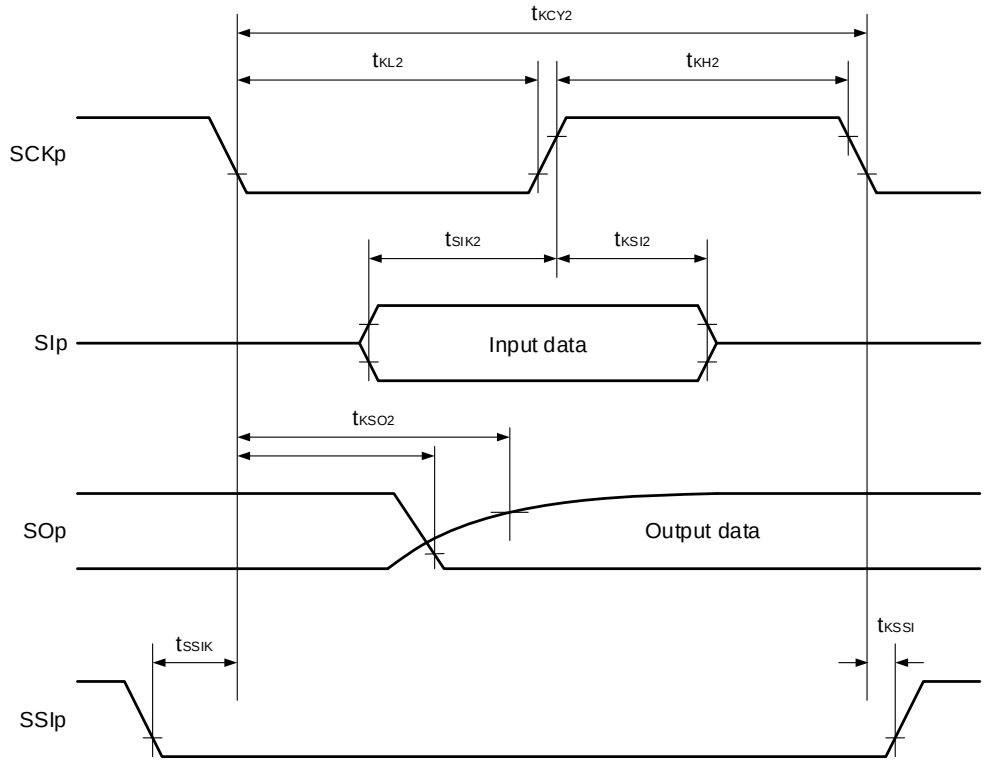
Remarks 1. p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

2. f_{MCK}: Serial array unit operation clock frequency

CSI Mode Connection Diagram (During Communication at Same Potential)

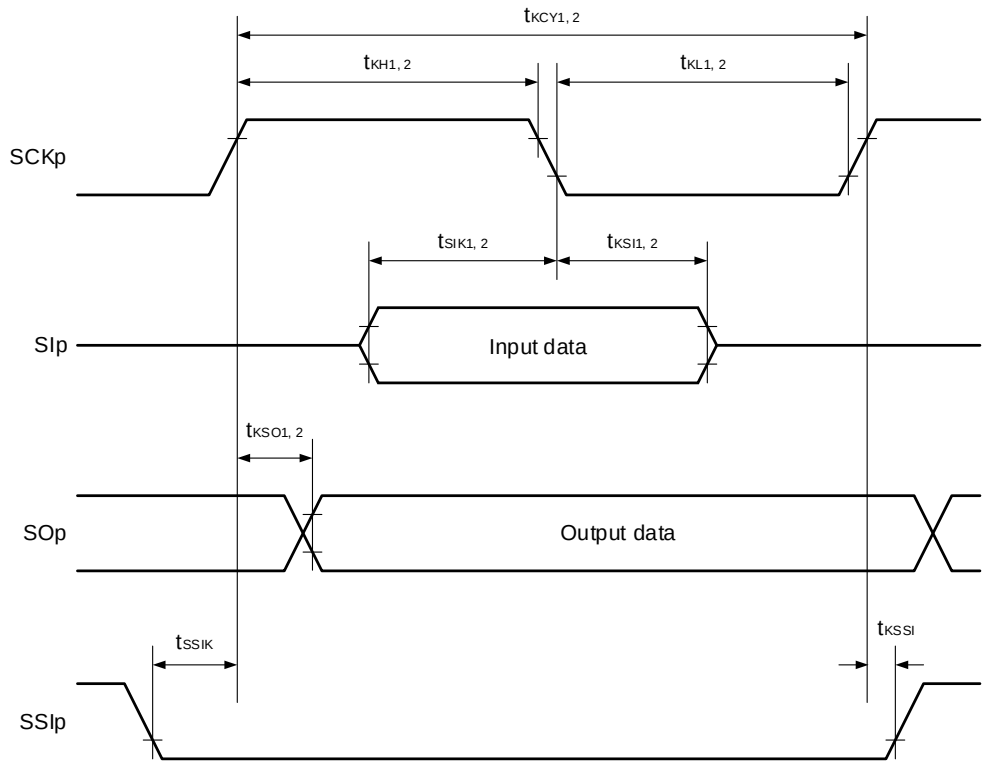


CSI Mode Serial Transfer Timing (During Communication at Same Potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



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)

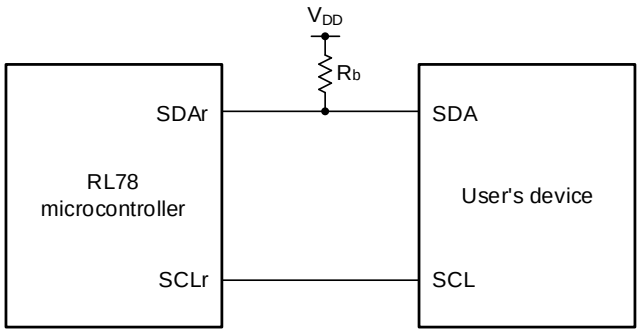
(6) During Communication at Same Potential (Simplified I²C Mode) (SDAr: N-ch Open-drain Output (EV_{DD} Tolerance) Mode, SCLr: Normal Output Mode)

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

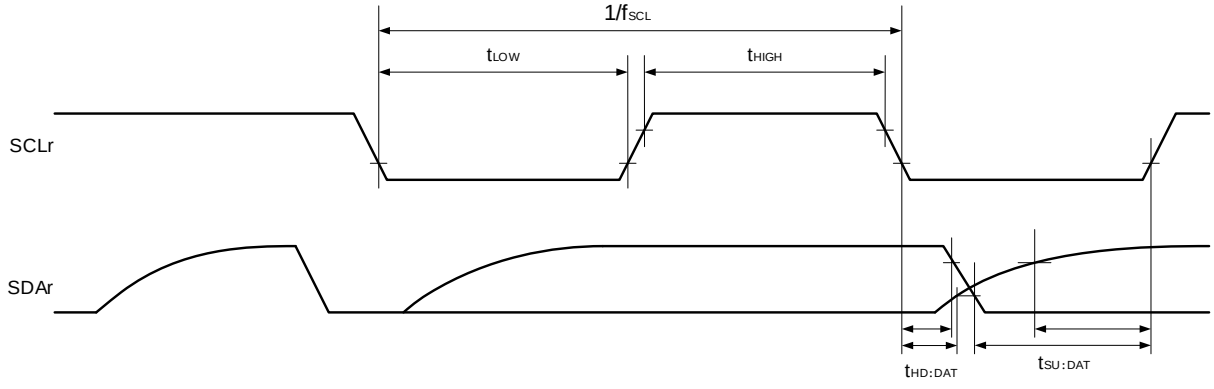
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 5.5 V	C_b = 50 pF, R_b = 2.7 kΩ			1000 Note	kHz
		1.8 V ≤ V_{DD} < 2.7 V	C_b = 100 pF, R_b = 3.0 kΩ			400 Note	kHz
Hold time when SCLr = "L"	t_{LOW}	2.7 V ≤ V_{DD} ≤ 5.5 V	C_b = 50 pF, R_b = 2.7 kΩ	475			ns
		1.8 V ≤ V_{DD} < 2.7 V	C_b = 100 pF, R_b = 3.0 kΩ	1150			ns
Hold time when SCLr = "H"	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 5.5 V	C_b = 50 pF, R_b = 2.7 kΩ	475			ns
		1.8 V ≤ V_{DD} < 2.7 V	C_b = 100 pF, R_b = 3.0 kΩ	1150			ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V	C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 85			ns
		1.8 V ≤ V_{DD} < 2.7 V	C_b = 100 pF, R_b = 3.0 kΩ	1/f_{MCK} + 145			ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V	C_b = 50 pF, R_b = 2.7 kΩ	0		305	ns
		1.8 V ≤ V_{DD} < 2.7 V	C_b = 100 pF, R_b = 3.0 kΩ	0		355	ns

Note f_{SCL} ≤ 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)



Cautions 1. Select normal input buffer and N-ch open-drain output mode for the SDAr pin, and select normal output mode for the SCLr pin.

2. Operating voltage condition: 1.8 V ≤ V_{DD} < 2.7 V is only available for Grade 3 products.

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

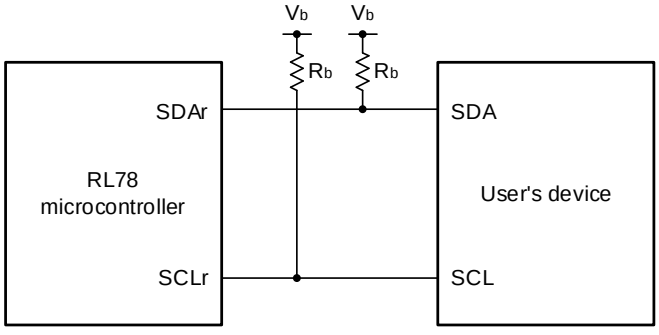
(7) During Communication at Same Potential (Simplified I^2C Mode) (SDAr and SCLr: N-ch Open-drain Output (EV_{DD} Tolerance) Mode)

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 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
		2.7 V ≤ V_{DD} < 4.0 V, C_b = 100 pF, R_b = 2.7 kΩ			
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 3.0 kΩ	1300		ns
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
		2.7 V ≤ V_{DD} < 4.0 V, C_b = 100 pF, R_b = 2.7 kΩ			
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 3.0 kΩ	400		ns
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
		2.7 V ≤ V_{DD} < 4.0 V, C_b = 100 pF, R_b = 2.7 kΩ	1/f_{MCK} + 270		ns
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 3.0 kΩ	1/f_{MCK} + 290		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
		2.7 V ≤ V_{DD} < 4.0 V, C_b = 100 pF, R_b = 2.7 kΩ			
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 3.0 kΩ	0	300	ns

Note f_{SCL} ≤ f_{MCK}/4 must also be satisfied.

Simplified I^2C Mode Connection Diagram (During Communication at Same Potential)



Cautions 1. Select normal input buffer and N-ch open-drain output mode for the SDAr and SCLr pins.

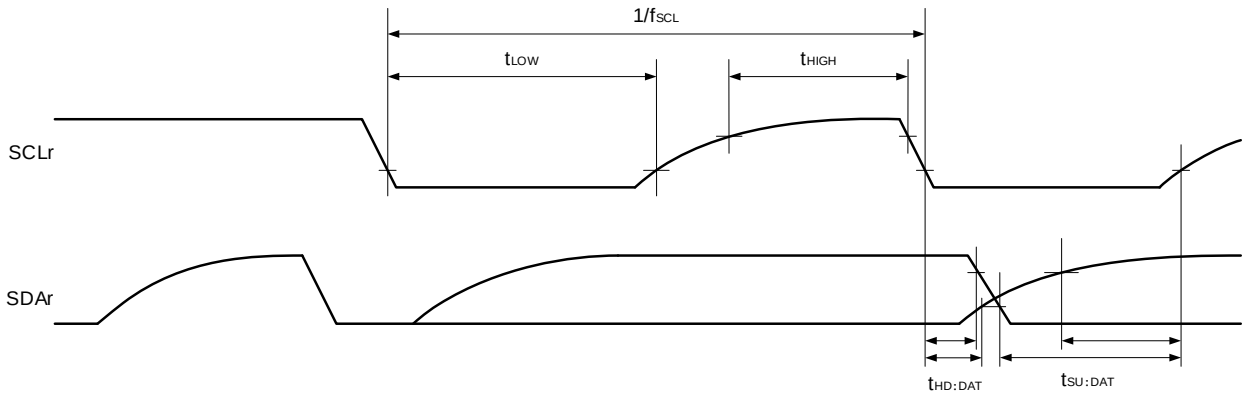
2. Operating voltage condition: 1.8 V ≤ V_{DD} < 2.7 V is only available for Grade 3 products.

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 I²C Mode Serial Transfer Timing (During Communication at Same Potential)



Remark r: IICr (r = 00, 01, 10, 11)

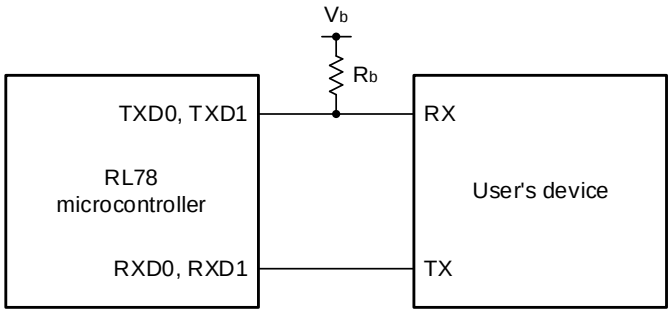
(8) Communication at Different Potential (UART mode) (TXD Output Buffer: N-ch Open-drain, RXD Input Buffer: TTL)

($T_A = T_{OPR}$, 4.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

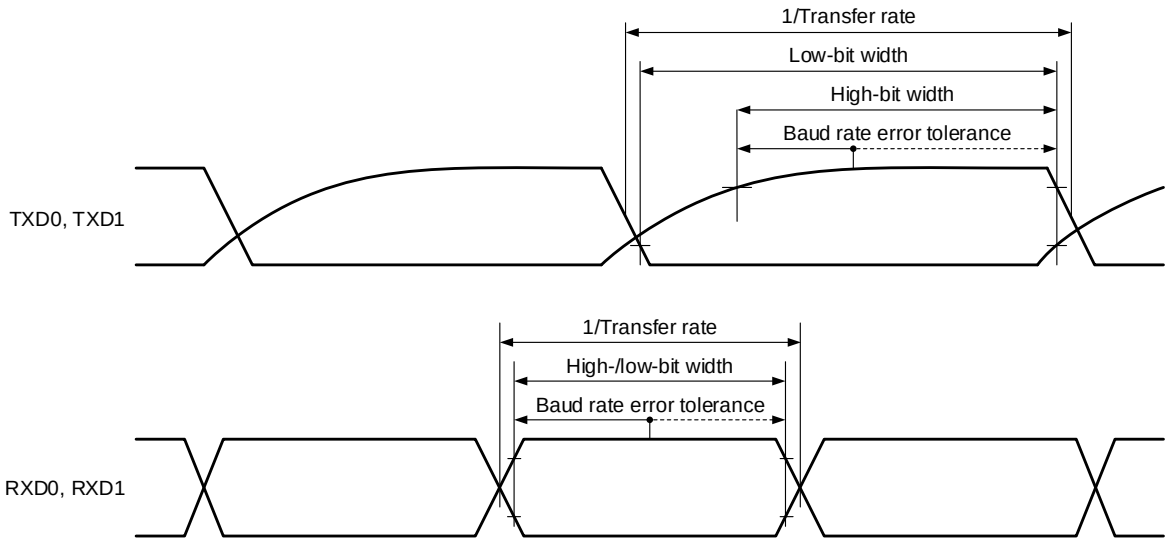
Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Transfer rate	–	Reception	2.7 V ≤ V_b ≤ V_{DD}, V_{IH} = 2.2 V, V_{IL} = 0.8 V					f_{MCK}/6	bps
				Theoretical value of the maximum transfer rate Note (C_b = 30 pF)	Grade 3 products			5.3	Mbps
					Grade 4 products			4.0	
		Transmission	2.7 V ≤ V_b ≤ V_{DD}, 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	Grade 3 products			5.3	Mbps
					Grade 4 products			4.0	

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 TTL input buffer for the RXD0 and RXD1 pins, and select N-ch open-drain output mode for the TXD0 and TXD1 pins.

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, SCKp … Internal Clock Output, Normal Slew Rate)

($T_A = T_{OPR}$, 4.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

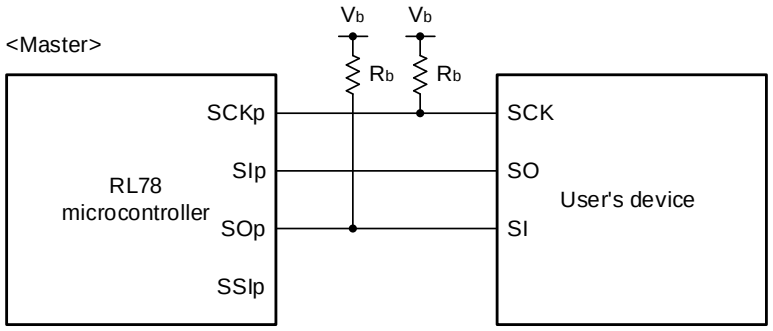
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	400 Note3			ns
SCKp high-level width	t_{KH1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}/2 − 75			ns
SCKp low-level width	t_{KL1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}/2 − 20			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	150			ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	70			ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	30			ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ	30			ns
Delay time from SCKp↓ to SOp output Note1	t_{KSO1}	2.7 V ≤ V_b ≤ V_{DD}, C_b = 30 pF, R_b = 1.4 kΩ			120	ns
Delay time from SCKp↑ to SOp output Note2	t_{KSO1}	2.7 V ≤ V_b ≤ V_{DD}, 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. $t_{KCY1} \geq 4/f_{MCK}$ must also be satisfied.

CSI Mode Connection Diagram (During Communication at Different Potential)



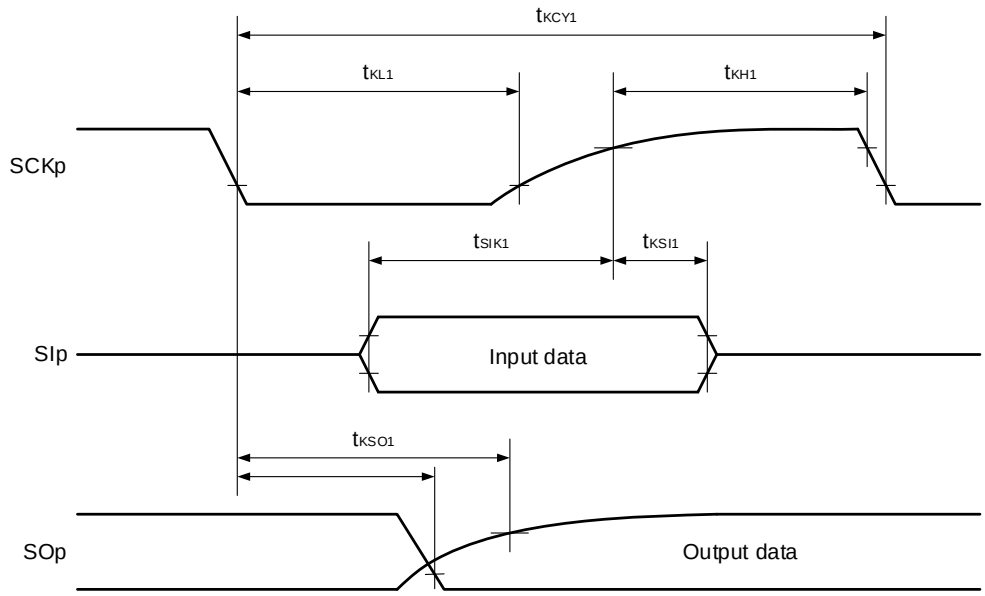
Caution Select TTL input buffer for the SIp pin, and select N-ch open-drain output mode for the SOp and SCKp pins.

Remarks 1. R_b [Ω]: Communication line (SCKp, SOp) pull-up resistance,
C_b [F]: Communication line (SOp, SCKp) load capacitance,
V_b [V]: Communication line voltage

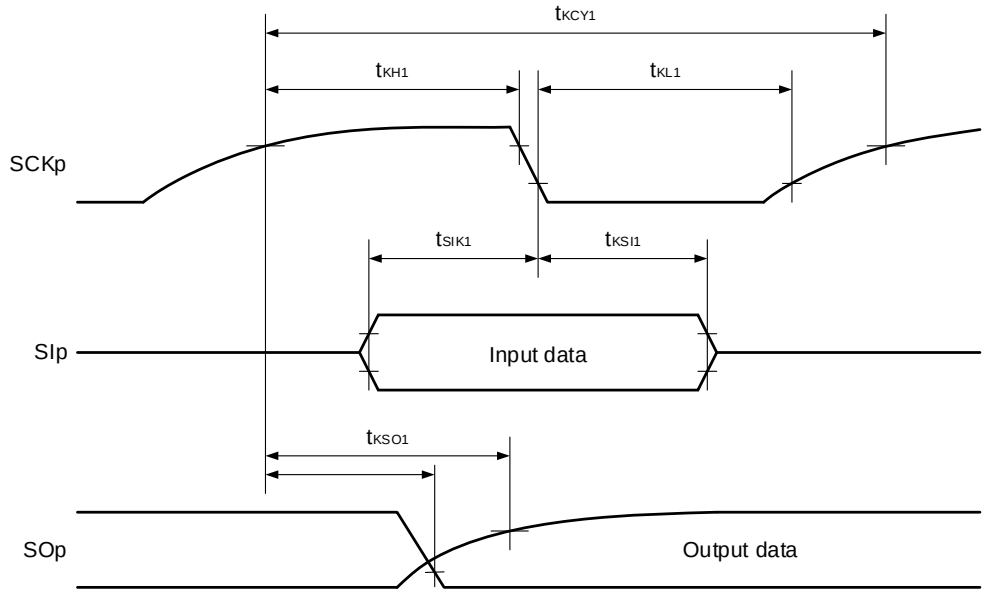
2. p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

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 ≤ V_{DD} ≤ 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)



Remark p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

(10) During Communication at Different Potential (3 V Supply System) (CSI Mode) (Slave Mode, SCKp … External Clock Input, Normal Slew Rate)

($T_A = T_{OPR}$, 4.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

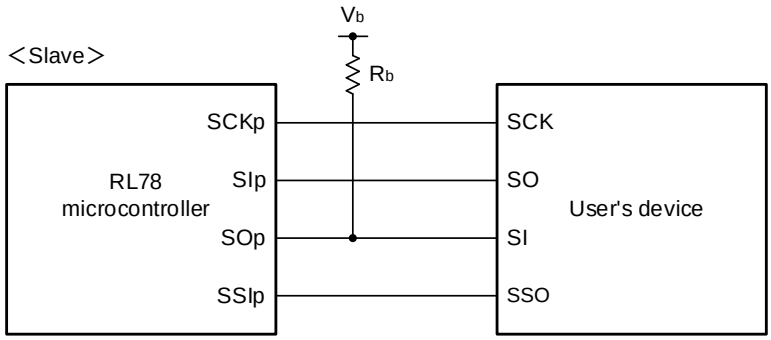
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY2}	2.7 V ≤ V_b ≤ V_{DD}	32 MHz < f_{MCK}				
			Grade 3 products	18/f_{MCK}			ns
			Grade 4 products	20/f_{MCK}			ns
			24 MHz < f_{MCK} ≤ 32 MHz				
			Grade 3 products	14/f_{MCK}			ns
			Grade 4 products	16/f_{MCK}			ns
			20 MHz < f_{MCK} ≤ 24 MHz				
			Grade 3 products	12/f_{MCK}			ns
			Grade 4 products	12/f_{MCK}			ns
			8 MHz < f_{MCK} ≤ 20 MHz				
			Grade 3 products	10/f_{MCK}			ns
			Grade 4 products	10/f_{MCK}			ns
			4 MHz < f_{MCK} ≤ 8 MHz				
			Grade 3 products	8/f_{MCK}			ns
			Grade 4 products	8/f_{MCK}			ns
			f_{MCK} ≤ 4 MHz				
			Grade 3 products	6/f_{MCK}			ns
			Grade 4 products	6/f_{MCK}			ns
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 SCKp↑) Note 1	t_{SIK2}			90			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}			1/f_{MCK} + 50			ns
Delay time from 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
SSIp setup time	t_{SSIK}	DAP = 0		120			ns
		DAP = 1		1/f_{MCK} + 120			ns
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 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 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 SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

CSI Mode Connection Diagram (During Communication at Different Potential)



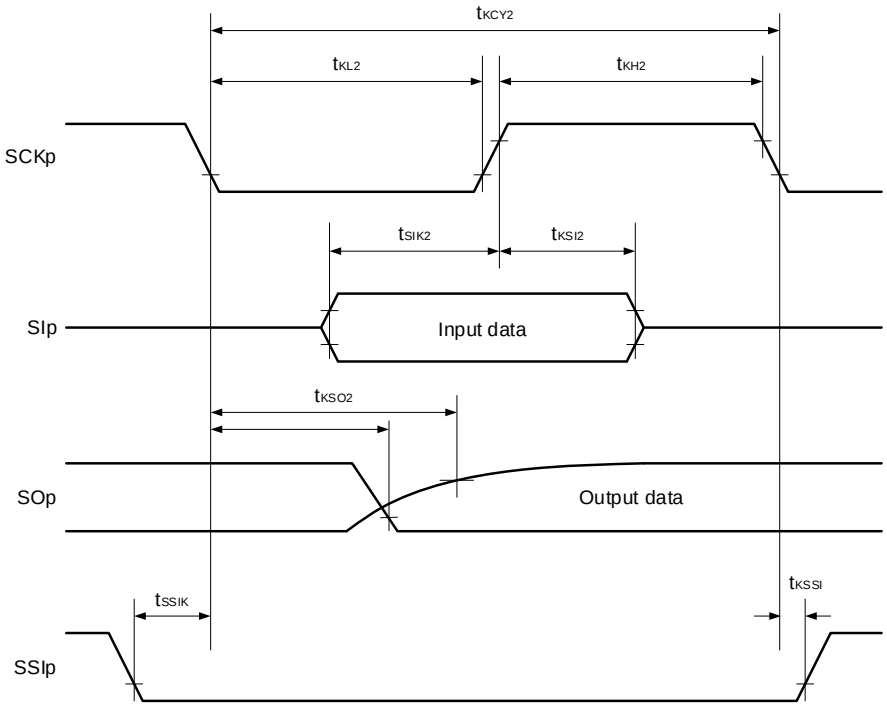
Caution Select TTL input buffer for the SIp, SCKp and SSIp pins, and select N-ch open-drain output mode for the SOp pin.

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

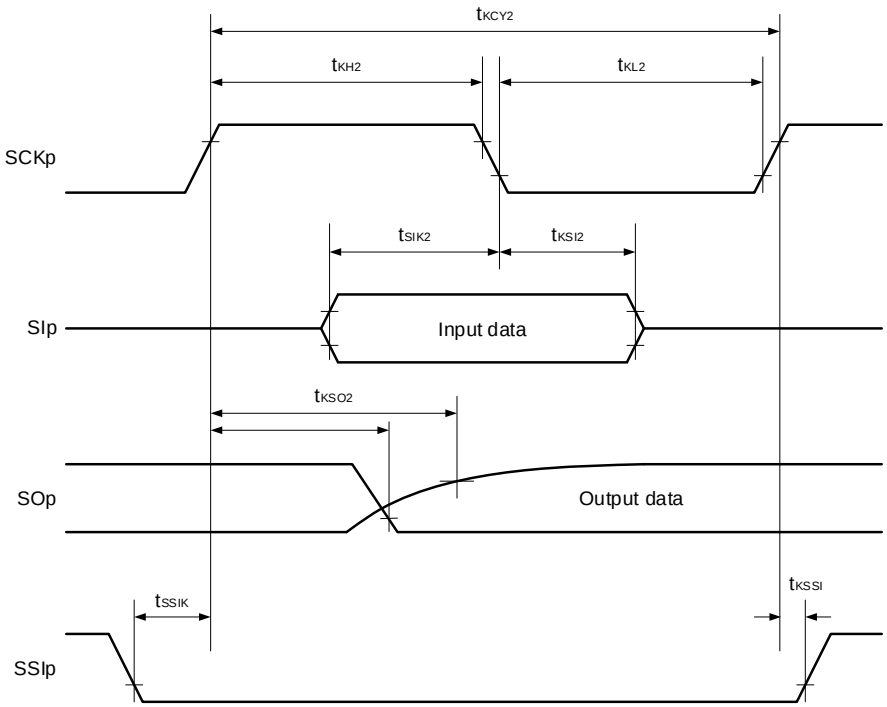
2. p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

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 ≤ V_{DD} ≤ 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)



Remark p: CSIp (p = 00, 01, 10, 11), m: Unit m (m = 0, 1), n: Channel n (n = 0, 1)

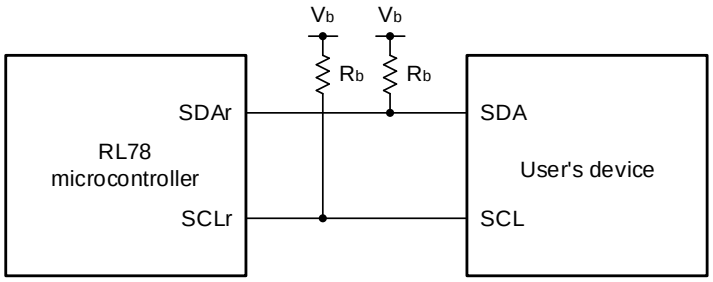
(11) During Communication at Different Potential (3 V Supply System) (Simplified I^2C Mode) (SDAr: TTL Input Buffer Mode and N-ch Open-drain Output (EV_{DD} tolerance) Mode, SCLr: N-ch Open-drain Output (EV_{DD} Tolerance) Mode)

($T_A = T_{OPR}$, 4.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

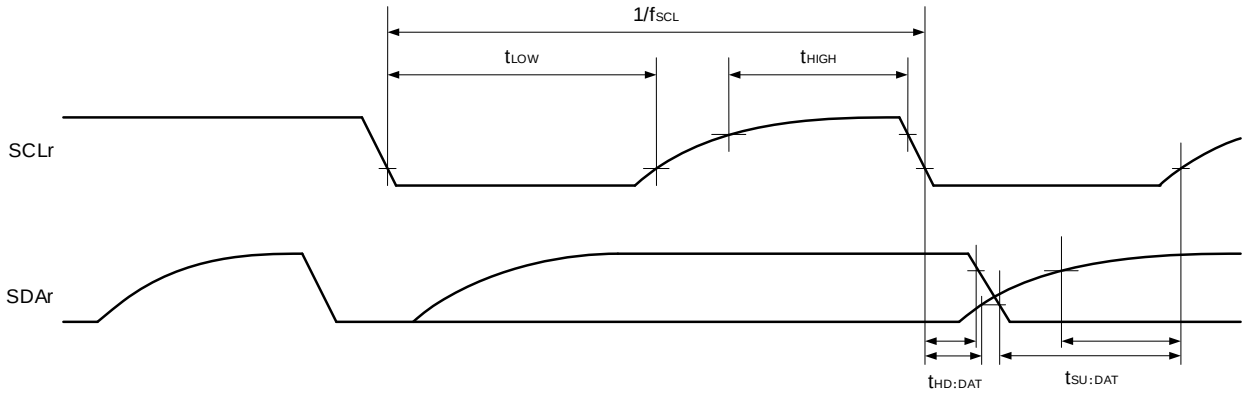
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ		400 Note	kHz
Hold time when SCLr = "L"	t_{LOW}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	1200		ns
Hold time when SCLr = "H"	t_{HIGH}	2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 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 $f_{SCL} \le f_{MCK}/4$ must also be satisfied.

Simplified I^2C Mode Connection Diagram (During Communication at Different Potential)



Simplified I^2C Mode Serial Transfer Timing (During Communication at Different Potential)



Caution Select TTL input buffer and N-ch open-drain output mode for the SDAr pin, and select 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

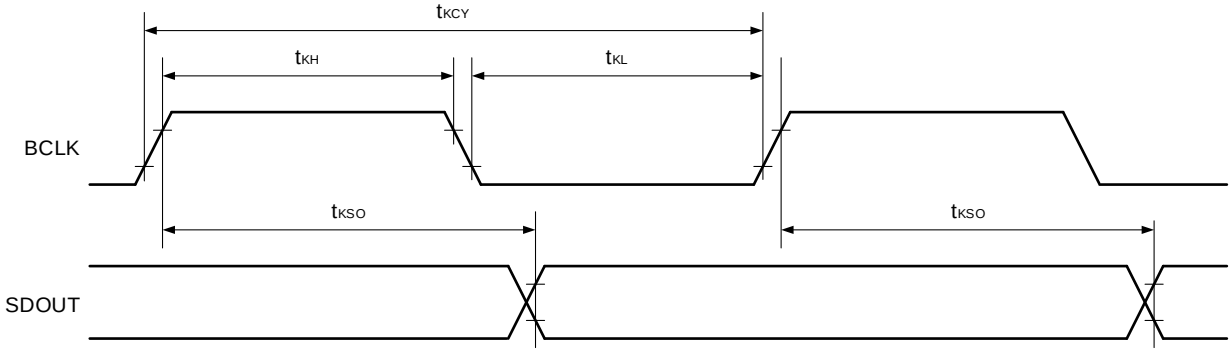
2. f_{MCK}: Serial array unit operation clock frequency

(12) During Communication at Same Potential (Simplified I²S Master Transmission) (CSI Mode, Slave Transmission Mode)

($T_A = T_{OPR}$, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions Note	MIN.	TYP.	MAX.	Unit
BCLK cycle time	t_{KCY}	Data rate = 2.5 MHz		400		ns
BCLK high-level width, low-level width	t_{KH}, t_{KL}		Value greater than 0.35 t_{KCY}			ns
SDOUT output delay time	t_{KSO}				Value less than 0.80 t_{KCY}	ns

Note The CPU/peripheral hardware clock frequency (f_{CLK}) must satisfy the condition of 8 MHz to 40 MHz.



Caution Set the I2SEN bit in the I2SER register to 1 (enable simple I2S communication function) and the TO121 and TO120 bits in the PWMDLY2 register to 01B (TO12 output is delayed by one cycle of f_{CLK}).

Remarks 1. BCLK: TO12 output of timer array unit 1, channel 2
SDOUT: SO10 output of serial array unit 1, channel 0
LRCLK: TO13 output of timer array unit 1, channel 3

2. Serial array unit 1, channel 0: CSI mode, slave transmission mode
Timer array unit 1, channel 2: Interval timer mode
Timer array unit 1, channel 3: External event counter mode

38.5.2 Serial Interface IICA

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 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 Note 3: 10 MHz ≤ f_{MCK}					0	1000	kHz
		Fast mode: 3.5 MHz ≤ f_{MCK}			0	400			kHz
		Normal mode: 1 MHz ≤ f_{MCK}	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 ACK (acknowledge) timing.

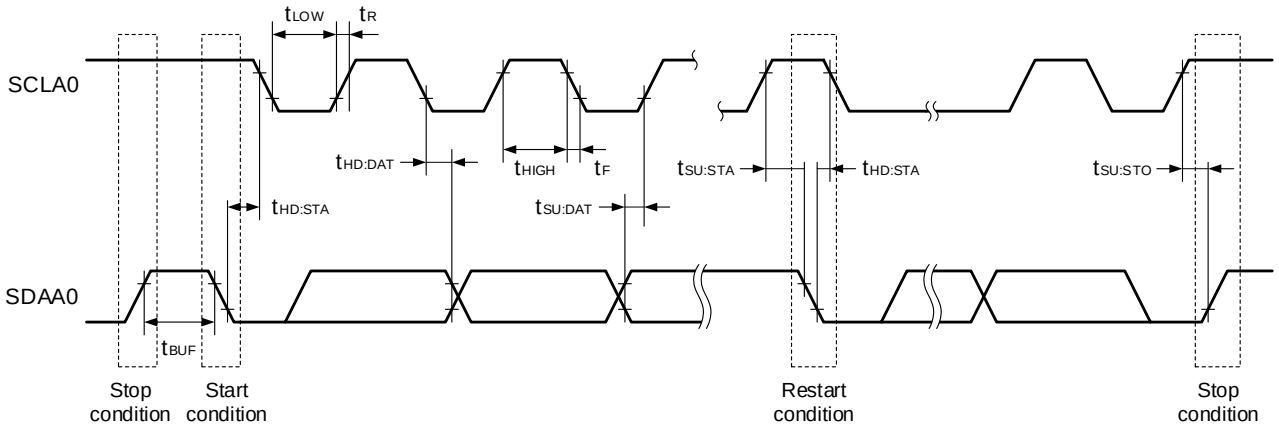
3. Fast mode plus is only available when 2.7 V ≤ V_{DD} ≤ 5.5 V.

Caution Operating voltage condition: 1.8 V ≤ V_{DD} < 2.7 V is only available for Grade 3 products.

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



38.5.3 On-chip Debug (UART)

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate	–		115.2 k		1 M	bps

Caution Operating voltage condition: 1.8 V ≤ V_{DD} < 2.7 V is only available for Grade 3 products.

38.5.4 LIN/UART Module (RLIN3) UART Mode

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Transfer rate	–	2.7 V ≤ V_{DD} ≤ 5.5 V	Operation mode, HALT mode	LIN communication clock source (f_{CLK} or f_{MX}): 4 MHz to 40 MHz			4000	kbps
			SNOOZE mode	LIN communication clock source (f_{CLK}): 2 MHz to 40 MHz			9.6	
		1.8 V ≤ V_{DD} < 2.7 V	Operation mode, HALT mode	LIN communication clock source (f_{CLK} or f_{MX}): 4 MHz to 16 MHz			2666	
			SNOOZE mode	LIN communication clock source (f_{CLK}): 2 MHz to 16 MHz			9.6	

Caution Operating voltage condition: 1.8 V ≤ V_{DD} < 2.7 V is only available for Grade 3 products.

38.6 Analog Characteristics

38.6.1 A/D Converter Characteristics

- When Reference voltage (+) = AV_{REFP}, Reference voltage (−) = AV_{REFM} = 0 V.
- Target ANI pin: ANI0 to ANI10, ANI22 to ANI27, Internal reference voltage (+).
- Conversion mode: High-speed conversion mode, Low-current conversion mode

(1) High-speed Conversion Mode

($T_A = T_{OPR}$, 2.7 V ≤ $AV_{DD} = V_{DD}$ ≤ 5.5 V, $AV_{SS} = V_{SS}$ = 0 V, Reference Voltage (+) = AV_{REFP}, Reference Voltage (−) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES				12	bit
Overall error Note 1	ABS	ANI0 to ANI10 Note 2, [4.5 V ≤ AV_{REFP}/AV_{DD} ≤ 5.5 V]			±5.0	LSB
		ANI0 to ANI10 Note 2, [2.7 V ≤ AV_{REFP}/AV_{DD} < 4.5 V]			±5.0	LSB
		ANI1, ANI2 Note 3, [4.5 V ≤ AV_{REFP}/AV_{DD} ≤ 5.5 V] [0.25 V ≤ V_{AIN} ≤ AV_{DD} − 0.25 V]			±6.0	LSB
		ANI1, ANI2 Note 3, [2.7 V ≤ AV_{REFP}/AV_{DD} < 4.5 V] [0.25 V ≤ V_{AIN} ≤ AV_{DD} − 0.25 V]			±8.0	LSB
		ANI22 to ANI27, [4.5 V ≤ AV_{REFP}/AV_{DD} ≤ 5.5 V]			±11	LSB
		ANI22 to ANI27, [2.7 V ≤ AV_{REFP}/AV_{DD} < 4.5 V]			±13	LSB
Integral linearity error Note 1	INL	ANI0 to ANI10			±3.0	LSB
		ANI22 to ANI27			±7.0	LSB
Differential linearity error Note 1	DNL	ANI0 to ANI10			±1.5	LSB
		ANI22 to ANI27			±3.5	LSB
Zero-scale error Note 1	ZSE	ANI0 to ANI10 Note 2			±4.5	LSB
		ANI22 to ANI27			±8.5	LSB
Full-scale error Note 1	FSE	ANI0 to ANI10 Note 2			±4.5	LSB
		ANI22 to ANI27			±8.5	LSB
Reference voltage (+)	AV_{REFP}		2.7		V_{DD}	V
Analog input voltage	V_{AIN}	ANI0 to ANI10, ANI22 to ANI27	0		AV_{REFP}	V
Internal reference voltage (+)	V_{BGR}	2.7 V ≤ V_{DD} ≤ 5.5 V	1.42	1.48	1.54	V
Analog input slew rate	SR				0.4	V/µs
Operation clock	f_{AD}		2		40	MHz
Conversion time Note 4 (per 1 channel)	t_{CONV}	ADCLK = 40 MHz, input impedance ≤ 0.5 kΩ				
		ANI0 to ANI10 Note 2	1.125			µs
		ANI22 to ANI27	1.8			µs
		ANI1, ANI2 Note 3	1.85			µs

Notes 1. Excludes quantization error (±1/2 LSB).

2. In case that dedicated sample & hold circuit is not used.

3. In case that dedicated sample & hold circuit is used.

4. The A/D conversion processing time (t_{CONV}) consists of sampling time and time for conversion by successive approximation.

(2) Low-current Conversion Mode

($T_A = T_{OPR}$, 1.8 V ≤ $AV_{DD} = V_{DD}$ < 2.7 V, $AV_{SS} = V_{SS}$ = 0 V, Reference Voltage (+) = AV_{REFP}, Reference Voltage (−) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES				12	bit
Overall error Note 1	ABS	ANI0 to ANI10, [1.8 V ≤ $AV_{REFP} = AV_{DD}$ < 2.7 V]			9.0	LSB
		ANI22 to ANI27, [1.8 V ≤ $AV_{REFP} = AV_{DD}$ < 2.7 V]			13	LSB
Integral linearity error Note 1	INL	ANI0 to ANI10, [$AV_{REFP} = V_{DD}$], [1.8 V ≤ $AV_{REFP} = AV_{DD}$ < 2.7 V]			4.0	LSB
		ANI22 to ANI27, [$AV_{REFP} = V_{DD}$], [1.8 V ≤ $AV_{REFP} = AV_{DD}$ < 2.7 V]			7.0	LSB
Differential linearity error Note 1	DNL	ANI0 to ANI10, [$AV_{REFP} = V_{DD}$], [1.8 V ≤ $AV_{REFP} = AV_{DD}$ < 2.7 V]			2.5	LSB
		ANI22 to ANI27, [$AV_{REFP} = V_{DD}$], [1.8 V ≤ $AV_{REFP} = AV_{DD}$ < 2.7 V]			3.5	LSB
Zero-scale error Note 1	ZSE	ANI0 to ANI10 Note 2, [$AV_{REFP} = V_{DD}$], [1.8 V ≤ $AV_{REFP} = AV_{DD}$ < 2.7 V]			8.0	LSB
		ANI22 to ANI27, [$AV_{REFP} = V_{DD}$], [1.8 V ≤ $AV_{REFP} = AV_{DD}$ < 2.7 V]			10.5	LSB
Full-scale error Note 1	FSE	ANI0 to ANI10 Note 2, [$AV_{REFP} = V_{DD}$], [1.8 V ≤ $AV_{REFP} = AV_{DD}$ < 2.7 V]			8.0	LSB
		ANI22 to ANI27, [$AV_{REFP} = V_{DD}$], [1.8 V ≤ $AV_{REFP} = AV_{DD}$ < 2.7 V]			10.5	LSB
Reference voltage (+)	AV_{REFP}		1.8		V_{DD}	V
Analog input voltage	V_{AIN}	ANI0 to ANI10, ANI22 to ANI27	0		AV_{REFP}	V
Internal reference voltage (+)	V_{BGR}	1.8 V ≤ V_{DD} < 2.7 V	1.42	1.48	1.54	V
Operation clock	f_{AD}		2		8	MHz
Conversion time Note 2 (per 1 channel)	t_{CONV}	ADCLK = 8 MHz, input impedance ≤ 0.5 kΩ				
		ANI0 to ANI10	6.75			µs
		ANI22 to ANI27	7.63			µs

Notes 1. Excludes quantization error (±1/2 LSB).

2. The A/D conversion processing time (t_{CONV}) consists of sampling time and time for conversion by successive approximation.

Cautions 1. Operating voltage condition: 1.8 V ≤ $AV_{DD} = V_{DD}$ < 2.7 V is only available for Grade 3 products.

2. The sample and hold circuit cannot be used when 1.8 V ≤ V_{DD} < 2.7 V.

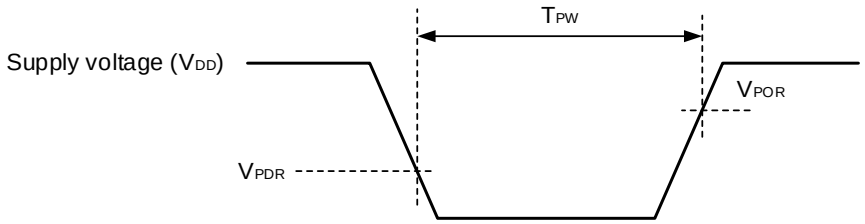
38.6.2 POR Circuit Characteristics

($T_A = T_{OPR}$, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage Note 1	V_{POR}, V_{PDR}		1.43	1.50	1.57	V
Minimum pulse width Note 2	T_{PW}		300			μs
Detection delay time	T_{PD}				350	μs

Notes 1. This indicates the POR circuit characteristics, and normal operation is not guaranteed under the condition of less than lower limit operation voltage (1.8 V).

2. Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}.



38.6.3 LVD Circuit Characteristics

(1) LVD0 Detection Voltage of Reset Mode

($T_A = T_{OPR}$, $V_{PDR} \le V_{DD} \le 5.5$ V, $V_{SS} = 0$ V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD00}	Power supply rise time	3.84	3.96	4.08	V
			Power supply fall time	3.76	3.88	4.00	V
		V_{LVD01}	Power supply rise time	2.88	2.97	3.06	V
			Power supply fall time	2.82	2.91	3.00	V
		V_{LVD02} Note	Power supply rise time	1.84	1.90	1.95	V
			Power supply fall time	1.80	1.86	1.91	V
Minimum pulse width		t_{LW}		500			μs
Detection delay time		t_{LD}				500	μs

Note This setting is available only for Grade 3 products (T_A = −40 to +105°C) and when used with $f_{CLK} \le 16$ MHz.

(2) LVD1 Detection Voltage of Interrupt Mode

($T_A = T_{OPR}$, $V_{PDR} \le V_{DD} \le 5.5$ V, $V_{SS} = 0$ V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD10}	Power supply rise time	4.08	4.16	4.24	V
			Power supply fall time	4.00	4.08	4.16	V
		V_{LVD11}	Power supply rise time	3.88	3.96	4.04	V
			Power supply fall time	3.80	3.88	3.96	V
		V_{LVD12}	Power supply rise time	3.68	3.75	3.82	V
			Power supply fall time	3.60	3.67	3.74	V
		V_{LVD13}	Power supply rise time	3.48	3.55	3.62	V
			Power supply fall time	3.40	3.47	3.54	V
		V_{LVD14}	Power supply rise time	3.28	3.35	3.42	V
			Power supply fall time	3.20	3.27	3.34	V
		V_{LVD15}	Power supply rise time	3.07	3.13	3.19	V
			Power supply fall time	3.00	3.06	3.12	V
		V_{LVD16}	Power supply rise time	2.91	2.97	3.03	V
			Power supply fall time	2.85	2.91	2.97	V
		V_{LVD17}	Power supply rise time	2.76	2.82	2.87	V
			Power supply fall time	2.70	2.76	2.81	V
		V_{LVD113} Note	Power supply rise time	2.05	2.09	2.13	V
			Power supply fall time	2.00	2.04	2.08	V
		V_{LVD114} Note	Power supply rise time	1.94	1.98	2.02	V
			Power supply fall time	1.90	1.94	1.98	V
		V_{LVD115} Note	Power supply rise time	1.84	1.88	1.91	V
			Power supply fall time	1.80	1.84	1.87	V
Minimum pulse width		t_{LW}		500			μs
Detection delay time		t_{LD}				500	μs

Note This setting is available only for Grade 3 products (T_A = −40 to +105°C) and when used with $f_{CLK} \le 16$ MHz.

38.7 Power Supply Voltage Rising Time

($T_A = T_{OPR}$, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum power supply voltage rising slope Note 1	S_{VRMAX}	0 V → V_{DD} (LVD0ENB = 0 or 1 Note 2)			50 Note 3	V/ms

Notes 1. Maintain the reset state by the voltage detection circuit (LVD0) or external reset up to the operating voltage range.

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.

38.8 Regulator Output Voltage Characteristics

($T_A = T_{OPR}$, V_{SS} = 0 V)

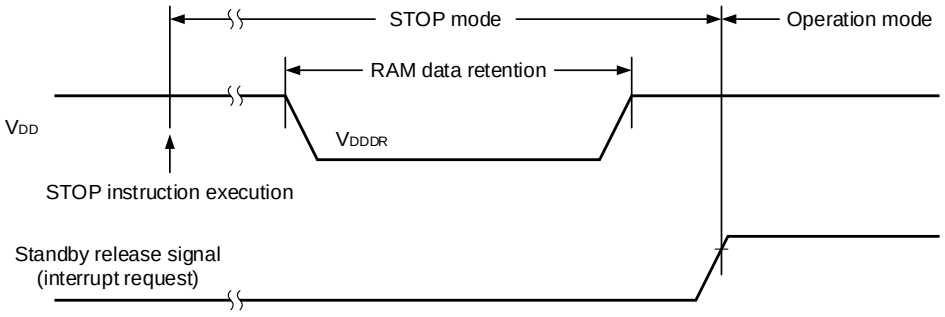
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
REGC output voltage	V_{OREGC}	C = 0.47 to 1 µF	1.45	1.50	1.56	V

38.9 RAM Data Retention Characteristics

($T_A = T_{OPR}$, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.47 Note		5.5	V

Note This depends on the POR detection voltage. For a falling voltage, data in RAM are retained until the voltage reaches the level that triggers a POR reset but not once it reaches the level at which a POR reset is generated.



38.10 Flash Memory Programming Characteristics

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency	f_{CLK}		2		40	MHz
Number of code flash rewrites Notes 1, 2, 3	Cerwr	Retained for 20 years, T_A = +85°C Note 4	1,000			Times
Number of data flash rewrites Notes 1, 2, 3		Retained for 20 years, T_A = +85°C Note 4	10,000			
		Retained for 5 years, T_A = +85°C Note 4	100,000			
Erase time	Terasa	Block erase	5			ms
Write time	Twrwa	1 word write	10			μs

Notes 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The starting point of the retaining years are after the erase.

2. When using flash memory programmer and Renesas Electronics self programming code.

3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

4. The average temperature for data retention.

Caution Operating voltage condition: 1.8 V ≤ V_{DD} < 2.7 V is only available for Grade 3 products.

(1) Code Flash Memory Processing Time

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item		f_{CLK} = 2 MHz		f_{CLK} = 4 MHz		f_{CLK} = 8 MHz		f_{CLK} = 16 MHz		Unit
	Conditions	TYP.	MAX.	TYP.	MAX.	TYP.	MAX.	TYP.	MAX.	
Programming time	4 bytes	52.0	470.0	43.0	400.0	38.0	360.0	36.0	340.0	μs
Erasure time	2 KB	7.8	265.0	6.5	240.0	5.9	225.0	5.8	220.0	ms
Blank checking time	4 bytes	–	19.2	–	13.2	–	10.3	–	9.1	μs
	2 KB	–	1310.0	–	660.0	–	335.0	–	270.0	μs

Item		f_{CLK} = 20 MHz		f_{CLK} = 32 MHz		f_{CLK} = 40 MHz		Unit
	Conditions	TYP.	MAX.	TYP.	MAX.	TYP.	MAX.	
Programming time	4 bytes	36.0	340.0	35.0	330.0	35.0	330.0	μs
Erasure time	2 KB	5.8	220.0	5.7	220.0	5.7	220.0	ms
Blank checking time	4 bytes	–	8.9	–	8.5	–	8.5	μs
	2 KB	–	270.0	–	245.0	–	245.0	μs

Cautions 1. The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

2. Operating voltage condition: 1.8 V ≤ V_{DD} < 2.7 V is only available for Grade 3 products.

(2) Data Flash Memory Processing Time

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item		f_{CLK} = 2 MHz		f_{CLK} = 4 MHz		f_{CLK} = 8 MHz		f_{CLK} = 16 MHz		Unit
	Conditions	TYP.	MAX.	TYP.	MAX.	TYP.	MAX.	TYP.	MAX.	
Programming time	1 byte	52.0	470.0	43.0	400.0	38.0	360.0	36.0	340.0	µs
Erasure time	1 KB	10.4	320.0	7.8	265.0	6.5	240.0	6.3	235.0	ms
Blank checking time	1 byte	–	19.2	–	13.2	–	10.3	–	9.1	µs
	1 KB	–	2605.0	–	1305.0	–	660.0	–	530.0	µs

Item		f_{CLK} = 20 MHz		f_{CLK} = 32 MHz		f_{CLK} = 40 MHz		Unit
	Conditions	TYP.	MAX.	TYP.	MAX.	TYP.	MAX.	
Programming time	1 byte	36.0	340.0	35.0	330.0	35.0	330.0	µs
Erasure time	1 KB	6.3	235.0	6.2	230.0	6.2	230.0	ms
Blank checking time	1 byte	–	8.9	–	8.5	–	8.5	µs
	1 KB	–	530.0	–	475.0	–	475.0	µs

Cautions 1. The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

2. Operating voltage condition: 1.8 V ≤ V_{DD} < 2.7 V is only available for Grade 3 products.

38.11 Dedicated Flash Memory Programmer Communication (UART)

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate	–	During serial programming	115.2 k		1 M	bps

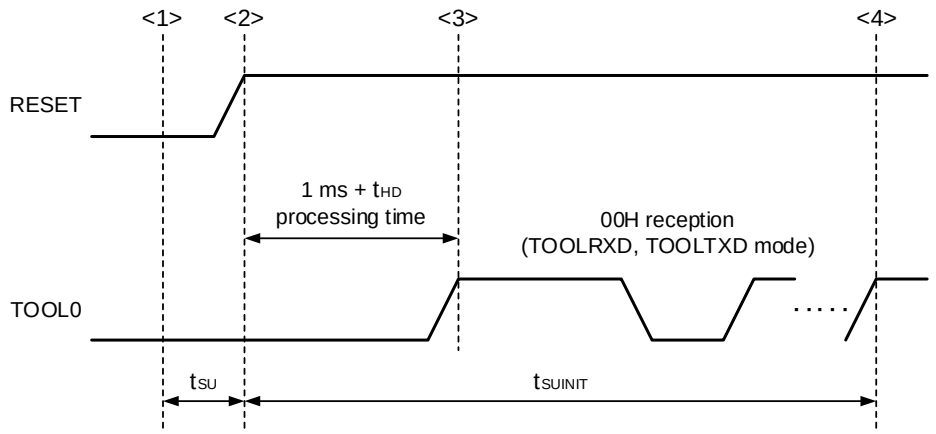
Caution Operating voltage condition: 1.8 V ≤ V_{DD} < 2.7 V is only available for Grade 3 products.

38.12 Timing of Entry to Flash Memory Programming Modes

($T_A = T_{OPR}$, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Time to complete the communication for the initial setting after the external reset is released	t_{SUINIT}	POR and LVD reset must be released before the external reset is released.			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	t_{SU}	POR and LVD reset must be released before the external reset is released.	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset must be released before the external reset is released.	1			ms

Caution Operating voltage condition: 1.8 V ≤ V_{DD} < 2.7 V is only available for Grade 3 products.



<1> The low level is input to the TOOL0 pin.

<2> The external reset is released (POR and LVD reset must be released before the external reset is released.).

<3> The TOOL0 pin is set to the high level.

<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: Communication for the initial setting must be completed within 100 ms after the external reset is released during this period

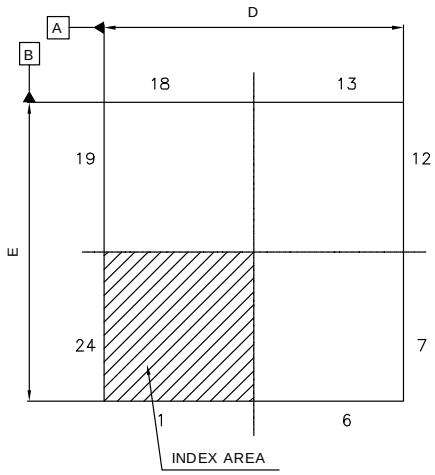
t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level

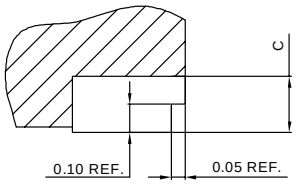
t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)

CHAPTER 39 PACKAGE DRAWINGS

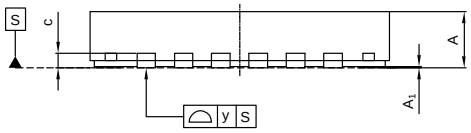
39.1 24-pin Products

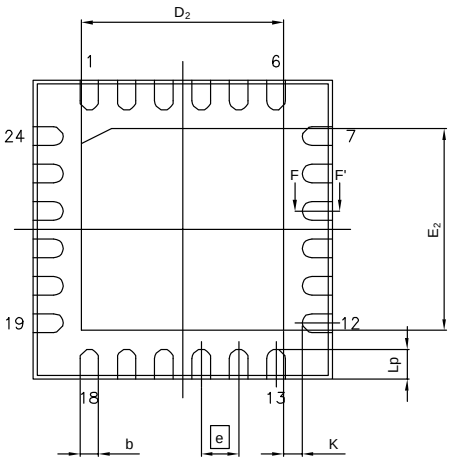
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN24-4x4-0.50	PWQN0024KJ-A	0.04





TERMINAL CROSS SECTION F-F'

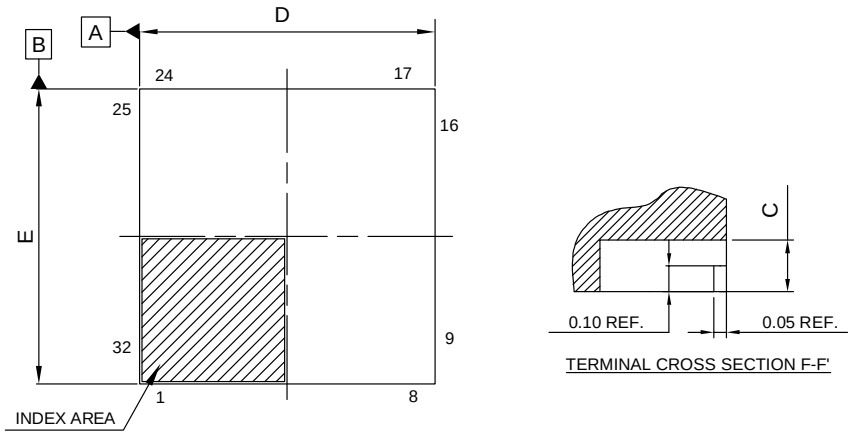


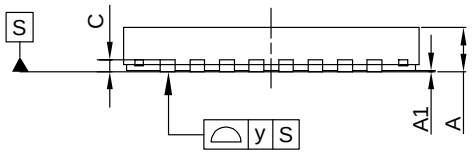


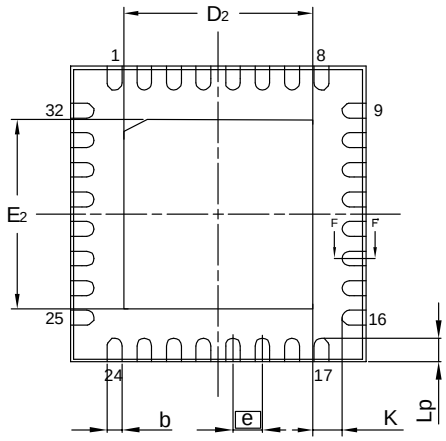
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	3.85	4.00	4.15
E	3.85	4.00	4.15
A	—	—	0.80
A_1	0.00	—	0.05
b	0.18	0.25	0.30
e	0.50 BSC		
Lp	0.35	0.40	0.45
y	—	—	0.08
c	—	0.20	—
K	0.20	—	—
D_2	—	2.70	—
E_2	—	2.70	—

39.2 32-pin Products

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN32-5x5-0.50	PWQN0032KH-A	0.06



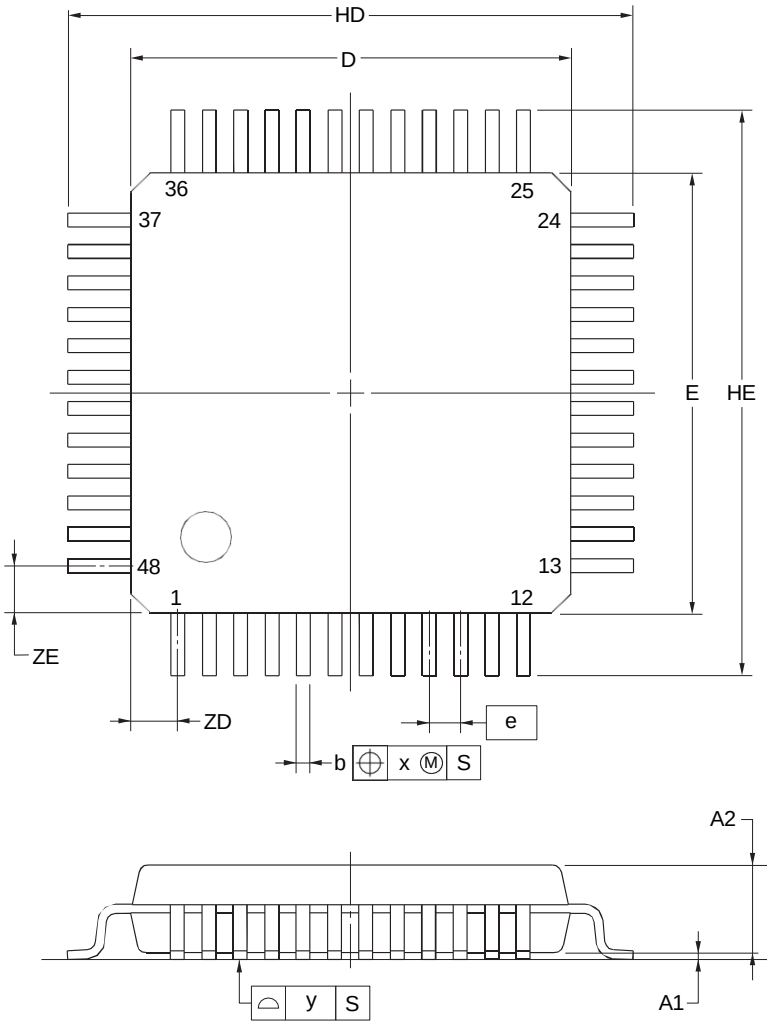


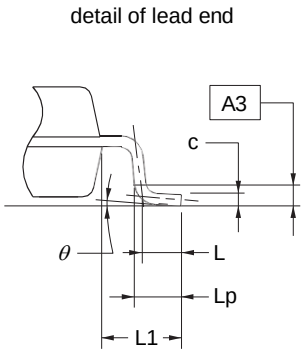


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	4.85	5.00	5.15
E	4.85	5.00	5.15
A	—	—	0.80
A_1	0.00	—	0.05
b	0.18	0.25	0.30
e	0.50 BSC		
Lp	0.35	0.40	0.45
y	—	—	0.08
c	—	0.20	—
K	0.20	—	—
D_2	—	3.20	—
E_2	—	3.20	—

39.3 48-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LFQFP48-7x7-0.50	PLQP0048KF-A	P48GA-50-8EU-1	0.16





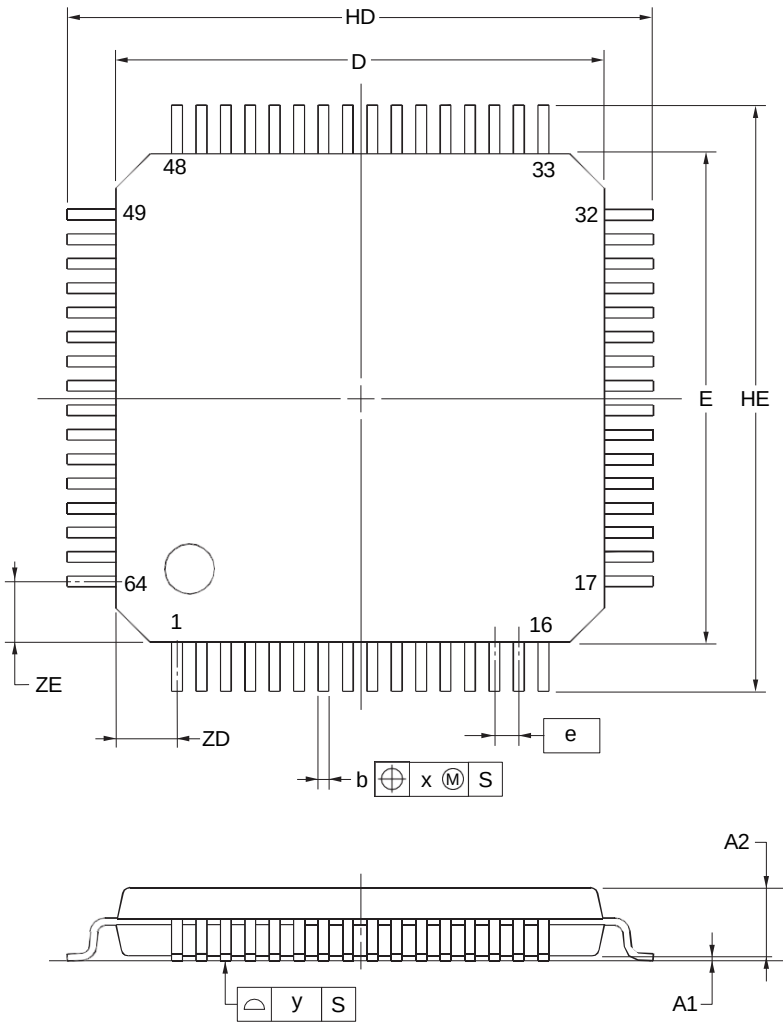
(UNIT:mm)

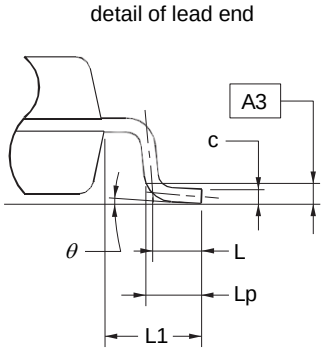
ITEM	DIMENSIONS
D	7.00±0.20
E	7.00±0.20
HD	9.00±0.20
HE	9.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.22±0.05
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	0.75
ZE	0.75

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

39.4 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





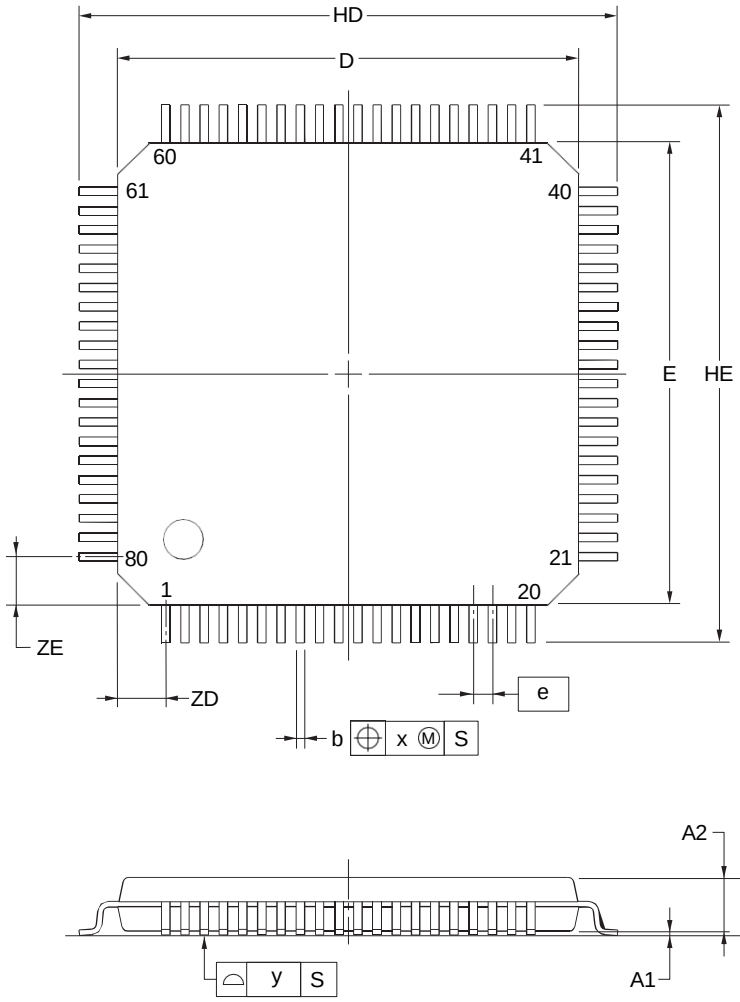
(UNIT:mm)

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

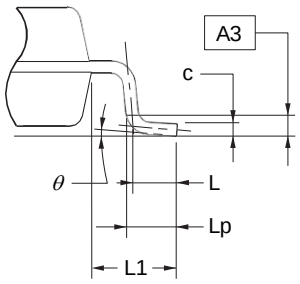
NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

39.5 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



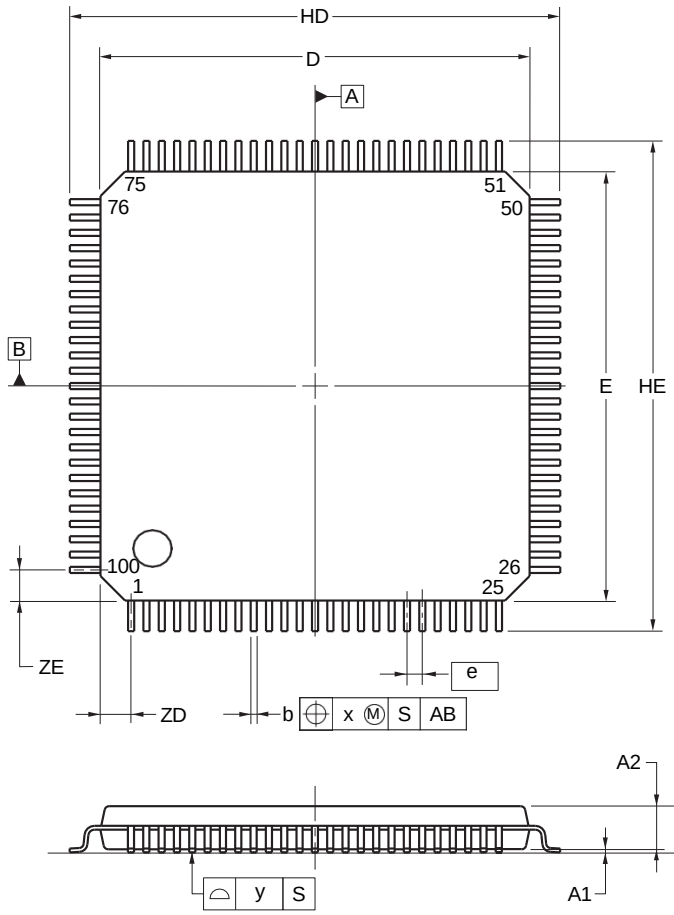
(UNIT:mm)

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

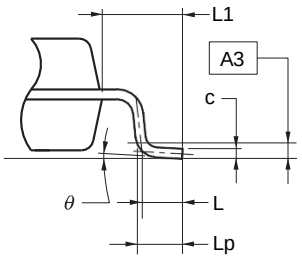
NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

39.6 100-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

APPENDIX A REVISION HISTORY

A.1 Major Revision History from Rev.1.10 to Rev.1.20

(1/2)

Page	Description	Classification
CHAPTER 4 PORT FUNCTIONS		
p.257	4.3.17 Figure 4-91, deleted caution 2 at the bottom of figure.	(c)
CHAPTER 8 TIMER RDe		
p.556	8.2.32 Figure 8-35, added Caution 7.	(c)
p.657	8.6.3.8, added Cautions (8) and (9).	(c)
CHAPTER 13 D/A CONVERTER		
p.778	13.4.1, added explanation on using it as a reference voltage for a comparator.	(c)
CHAPTER 14 COMPARATOR		
p.785, 786	14.2.2 Figure 14-3, added note 7 to CDFS1, 0 bits.	(c)
p.789	14.2.4 Figure 14-5, added explanation of IVCMP0n pin to caution.	(c)
p.794	14.3, added explanation for using comparator.	(c)
CHAPTER 15 CAPACITIVE SENSING UNIT (CTSU2SLa)		
p.810	15.2.2 Figure 15-7, corrected the description of the MD1 bit.	(c)
p.811	15.2.2 Figure 15-7, corrected the description of the PUMPON bit.	(c)
p.822	15.2.7 Figure 15-12, corrected the description of the ICOMP1 and MFC[1:0] bits.	(c)
p.857	15.4.11, added explanation of the impact of "changes in port input/output current".	(c)
p.858	15.4.11 Table 15-4, corrected the referenced document in the note.	(c)
CHAPTER 17 SERIAL INTERFACE IICA		
p.1079	17.3.7, added a note to the description in the text.	(c)
p.1079	17.3.7, added a new Remark.	(c)
p.1084	17.4.2, added Caution 3.	(c)
CHAPTER 20 DTC		
p.1576	20.3.4, added explanation for using multiple control data.	(c)
CHAPTER 24 STANDBY FUNCTION		
p.1667	24.3.3 (2) Figure 24-13, corrected flowchart.	(c)
CHAPTER 29 FUNCTIONAL SAFETY		
p.1756	29.3.3 (1) Figure 29-8, added example description to ERAD bit.	(c)
p.1773	29.3.4 (7) Figure 29-22, added Remarks below the figure.	(c)
p.1789	29.3.7 (4) Figure 29-36 (a), corrected a typo (PLL through mode → Clock through mode).	(a)
CHAPTER 32 OPTION BYTE		
p.1799	32.2 Figure 32-1, corrected the description of “Caution”.	(c)
CHAPTER 33 FLASH MEMORY		
p.1806	33, added a note on code flash memory erasure in the text.	(c)
p.1806	33, corrected description of data flash memory rewriting.	(c)
p.1819	33.7, added explanation for self-programming to Caution 5.	(c)
p.1820	33.7.1 Figure 33-8, corrected description in flowchart.	(c)
p.1829	33.7.2.8 Figure 33-16, corrected description of SQEND and ESQEND bits.	(c)
p.1832	33.7.2.11, corrected description of PFS register.	(c)
p.1847	33.7.6.9, added description of processing time to the command completion confirmation procedure for the flash memory area sequencer.	(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

(2/2)

Page	Description	Classification
p.1847	33.7.6.9, added description of processing time to the command completion confirmation procedure for the extra area sequencer.	(c)
p.1847	33.7.6.10, added description of processing time to the forced termination procedure of the flash memory area sequencer.	(c)
p.1848	33.7.7, corrected description of boot swap procedure.	(c)
p.1854	33.7.10 (2), added text to note.	(c)
p.1855	33.8, corrected description of security settings.	(c)
CHAPTER 37 ELECTRICAL SPECIFICATIONS (RL78/F25)		
p.1902	37.3.1 (6/6), corrected description of positive injected current.	(b)
p.1935	37.6.3, corrected the description in the comparator characteristics table.	(b)
CHAPTER 38 ELECTRICAL SPECIFICATIONS (RL78/F22)		
p.1954	38.3.1 (6/6), corrected description of positive injected current.	(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

A.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Section indicates the section of each edition.

Edition	Description	Section
Rev.1.10	First edition issued.	(All chapters)

RL78/F22, F25 User’s Manual: Hardware

Publication Date: Rev.1.20 Jun 30, 2025

Published by: Renesas Electronics Corporation

RL78/F22, F25

Renesas Electronics Corporation

R01UH1061EJ0120